TOWARD LEVERAGING ARTIFICIAL INTELLIGENCE TO SUPPORT THE IDENTIFICATION OF ACCESSIBILITY CHALLENGES

Wajdi Mohammed R Aljedaani Sr.

Dissertation Prepared for the Degree of

DOCTOR OF PHILOSOPHY

UNIVERSITY OF NORTH TEXAS

May 2023

APPROVED:

Stephanie Ludi, Major Professor
Mohamed Wiem Mkaouer, Committee Member
Hyunsook Do, Committee Member
Paul Tarau, Committee Member
Gergely Záruba, Chair of the Department of Computer Science and Engineering
Shengli Fu, Interim Dean of the College of Engineering
Victor Prybutok, Dean of the Toulouse Graduate School

Aljedaani Sr., Wajdi Mohammed R. *Toward Leveraging Artificial Intelligence to Support the Identification of Accessibility Challenges*. Doctor of Philosophy (Computer Science and Engineering), May 2023, 214 pp., 34 tables, 329 numbered references.

The goal of this thesis is to support the automated identification of accessibility in user reviews or bug reports, to help technology professionals prioritize their handling, and, thus, to create more inclusive apps. Particularly, we propose a model that takes as input accessibility user reviews or bug reports and learns their keyword-based features to make a classification decision, for a given review, on whether it is about accessibility or not. Our empirically driven study follows a mixture of qualitative and quantitative methods. We introduced models that can accurately identify accessibility reviews and bug reports and automate detecting them. Our models can automatically classify app reviews and bug reports as accessibility-related or not so developers can easily detect accessibility issues with their products and improve them to more accessible and inclusive apps utilizing the users' input. Our goal is to create a sustainable change by including a model in the developer's software maintenance pipeline and raising awareness of existing errors that hinder the accessibility of mobile apps, which is a pressing need. In light of our findings from the Blackboard case study, Blackboard and the course material are not easily accessible to deaf students and hard of hearing. Thus, deaf students find that learning is extremely stressful during the pandemic.

ACKNOWLEDGMENTS

Obtaining a Ph.D. is like seeing my greatest dreams become a reality. This journey was filled with moments that were both exciting and frustrating, as well as unpredictable. Despite this, many people offered me encouragement and inspiration, which helped me keep going. At this point, I have nothing except endless gratitude for the people who have supported me along the journey.

First and foremost, I am incredibly grateful to the Almighty God for successfully enabling me to complete this doctoral research.

I would like to express my gratitude to my parents and my siblings for the insightful advice and understanding they have provided. I am at a loss for words to adequately express my gratitude to my parents for the selfless love and direction they have shown me throughout my life, as well as for the unwavering moral, spiritual, emotional, and financial support they have offered me, and for instilling in me an optimistic and determined outlook. I will never be able to adequately express my gratitude to you both for having faith in me and allowing me to have freedom. You are rock-solid support that I can always rely on in any situation.

I indebted a tremendous amount of gratitude to my advisor Dr. Stephanie Ludi and co-advisor, Dr. Mohamed Wiem Mkaouer, for their consistent and invaluable guidance and support, for several insightful discussions and comments, for providing necessary information and direction, and for assisting me with all of the necessary protocols on the way to completing my Ph.D. I am deeply appreciative of all of these things. Their perceptive remarks drove me to hone my thinking, which elevated the quality of my work to a higher level. This research was only feasible with the helpful assistance of participants like you.

I would like to take this opportunity to thank everyone who has supported and encouraged me throughout my doctoral studies, including Hossam Kalifa, Moath Almesbahi, Sami Saeed, Talal Almutari, Nasser Almudar, Ahmed Almusaad, Hossam Bakeet, abdulrazaq Alsehli, Kariem Sabir, and Mary Sabir. In particular, I would like to thank Hossam Kalifa.

TABLE OF CONTENTS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

In the modern digital age, there is a need to ensure that devices, content, and applications can be utilized effectively by all people [278]. In the modern education and software engineering fields, the need for accessibility has been popularized because of the recognition that different people have varying abilities, and there is a need to design software that enables every user to access the full range of the functions provided by such programs [80]. However, although the subject of accessibility has gained prominence in the past few years, it is considered that a lot still needs to be done to deal with accessibility problems [278, 67, 111]. In particular, research has shown that students with disabilities are the most affected by accessibility issues [80, 67].

Given the complexity of modern software systems, there has been a need to simplify them and make them usable in everyday life. For that reason, the subject of accessibility has become ubiquitous, which aims at ensuring that software systems are both acceptable and usable [142, 129]. It would be plausible to describe an accessible platform as one which has a universal design that enables even users with various challenges to benefit from such systems [81]. A universal design is defined as a range of approaches, designs, and ideas that can be infused into a certain software system that makes it easier for end-users to utilize the program [160]. In most cases, developers often make the assumption that users are able to read texts on the screen, type using their keyboard, utilize the mouse to point or select items, and hear sounds from the system [78]. Unfortunately, some people have physical and mental difficulties that make them unable to effectively utilize the functions in a given technology platform [185].

There is a growing need to ensure the accessibility of software applications. In some countries, laws have been made to compel software designers to create accessible programs. Furthermore, there is increasing recognition that a software application should follow the functional requirements needed. It should also be integrated with all steps of the software development process, including design, system specification, and testing [327]. However, there is no clarity regarding the utilization of accessibility aspects in the software design process and the generation of software platforms [307]. When accessibility is not considered during the beginning of software development, it often results in poor user feedback, lack of usability, and possible re-engineering to rectify problems. It would also be important to mention that accessibility specifications have moved a notch higher, and modern-day developers are supposed to ensure that many categories of end-users can effectively utilize the platform [293].

There is a plethora of research that addresses the importance of accessibility for disabled persons. Some of the literature on the topic has focused on accessibility for Android applications [307, 48]. It is prudent to note that although accessibility issues have been extensively addressed, there is a dearth of literature on the issues that students and teachers are facing with regard to accessibility and the challenges of practitioners in ensuring accessibility for various types of users.

1.1. Research Challenges

It is crucial that mobile applications be accessible to allow all individuals with different abilities to have fair access and equal opportunities [144]. Prior studies investigated the accessibility issues raised in Android applications [48, 307], and others evaluated the accessibility of various websites [9, 107, 158, 315]. Accessibility tools are still underused as there is a lack of trust in these tools, and so developers prefer to perform manual analysis. Additionally, recent studies questioned the challenges that educators encounter related to accessibility and to overcome these challenges. These challenges reveal a lack of accessibility culture. In this proposal, we focus on the following challenges:

- The COVID-19 pandemic has necessitated the introduction of various public health measures to control its spread, including social distance measures. Such policies have affected nearly every sector of the economy, including education. Unfortunately, Gleason et al. [128] indicated that People With Disabilities (PWD) are often disproportionately affected in times of drastic and unintended changes. In the case of COVID-19, PWD faces challenges in education because social distance measures have forced education institutions to shift from face-to-face learning to e-learning. As noted by Hanjarwati and Suprihatiningrum [134], some of the challenges faced include a lack of support, expensive internet access, and the inability to work with the e-learning system, among others. It is important to raise awareness of how inclusivity in education can be achieved during the COVID-19 era, such as by promoting the use of blended learning, providing sign language options, and improving support for disabled persons [301]. It is also important to resolve barriers to education for disabled students, which include technical problems, time, and absence of simultaneous translation, among others [46]. However, studies performed in Saudi Arabia on disability, specifically with deaf students, are limited. There are only two studies that investigated deaf education during the pandemic. Madhesh [193] investigated deaf students' situations through 20 ministries of education channels that were utilized during the locked-down period. The second study was conducted by Alsadoon and Turkestani [46], where they investigated the instructors' obstacles while teaching online classes. Both studies were conducted on teachers of deaf students, but they did not examine the deaf students' challenges and concerns during the sudden shift to online learning. This focuses on the challenges deaf students have faced when transitioning to online learning during the pandemic. More specifically, this investigation is unique since, compared to other countries, online learning is not very established in Saudi Arabia, and its implementation has mostly been heightened by COVID-19. Saudi society is also traditional and conservative, wherein the deaf culture is still new and not well-established [39]. Thus, such students may have low self-esteem in communication [10], which may affect how they learn using the e-learning platforms. Furthermore, the context of this research is unique in terms of its gender focus, given that the Technical and Vocational Training Corporation (TVTC) only admits male students, unlike the participants in other studies that were both male and female. Therefore, the context of this investigation is unique, and its findings will greatly contribute to the body of research on the subject.

- Analyzing app reviews was used by technology professionals to identify issues with their mobile apps [192, 91, 181]. However, accessibility in user reviews is rarely studied especially for mobile applications [112]. Identifying accessibility-related reviews is currently done using two main methods: manual identification and automatic detection [112]. The manual identification approach is time-consuming, especially with the vast number of reviews that users upload to the app stores, and so it becomes impractical. The automated detection method employs a string-matching technique as a predefined set of keywords are searched for in the app reviews [112]. These keywords were extracted from the British Broadcasting Corporation (BBC) recommendations for mobile accessibility [62]. While this method sounds more practical than the manual one, it has its own drawbacks: the string-matching technique ignores that keywords derived from guidelines do not necessarily match the words expressed in reviews posted by users. This mismatch includes but is not limited to situations when the keywords are incorrectly spelled by users. More importantly, the presence of certain keywords in a review does not necessarily mean that the review is about accessibility. For example, consider the following reviews from Eler et al. dataset [112]:

  > This is the closest game to my old 2001 Kyocera 2235's inbuilt game. Everything is so simple and easy to comprehend, but that doesn't mean that it is easy to complete right off the bat. Going into the sewers almost literally blind (sight and knowledge of goods in inventory) is a great touch too. Keep at it. I'll support you at least in donations.

  This review contains a set of keywords that could indicate accessibility (e.g., old, blind, and sight), but it is not an accessibility review. In this review, the word old refers to a device rather than a person. The words blind and sight refer to knowledge of goods in the game rather than describing a player's vision. Therefore, the discovery of accessibility reviews heavily relies on the *context*, so simply searching for their existence in the review text is inefficient. Due to the overhead of manual identification and the high false-positiveness of automated detection, these two methods remain impractical for developers to use, and so, accessibility reviews remain hard to identify and prioritize for correction. To address this challenge, it is critical to design a solution with *learning capabilities*, which can take a set of examples that are known to be accessibility reviews and another set of examples that are not about accessibility but do contain accessibility-related keywords, and learn how to distinguish between them. Thus, there is a need to for automated detection of accessibility user reviews that may support various applications and provide actionable insights to software practitioners and researchers, including empirical studies around user reviews.

- The Internet has become an effective tool through which people communicate their feelings, emotions, and ideas [124]. Business analysts use this data to monitor people's perceptions and opinions about their products. Natural Language Processing (NLP) based methods have been widely used to automatically detect data contents from the text [90]. Artificial Intelligence (AI) based approaches have gained prominence for the development of sentiments or emotion-based systems [198]. In state-of-the-art Sentiment Analysis techniques, the issue is that they access the response in the context of positive or negative aspects but not the specific feelings of the customer and the intensity of their response. As the expression of users' thoughts regarding the apps, reviews are used as a tool. If the accessibility features address the users' needs, the user reviews are written with positive sentiments. On the other hand, if the accessibility features are not meeting user requirements, then attention is needed by the developers. These reviews reflect negative sentiments. Therefore, a review serves as a way to measure user satisfaction or dissatisfaction with accessibility, and negative reviews help identify accessibility topics that need to be fixed. For many persons with disabilities (such as those who are deaf or blind), expressing their reviews of various apps can be challenging. However, they can express their emotions (positive, neutral, or negative) towards an app, which may help developers understand whether it is accessible or not. Thus, there is a need to automated detection sentiment analysis of accessibility user reviews could be the solution for determining the emotions of people with disabilities towards the accessibility of mobile devices.

- People with disabilities or special needs rely heavily on accessibility software applications in their everyday life (find accessible locations, customized UIs, voice translation, communication, driving, shopping, etc.). Having accessibility-related bugs can severely impact their lives, from preventing them from participating in new activities to threatening their lives in critical situations due to the sensitive nature of disabled people. Therefore, identifying and prioritizing these bugs are of crucial importance. Yet, the manual identification of these bug reports is time-consuming, human-intensive, and error-prone. The textual nature of bug reports adds another layer of challenges related to the meaning ambiguity of these natural language descriptions. To illustrate this problem, let us consider the following two examples:

  Example 1: Missing labels on the buttons in the "Select how you want to use Weave [1]

  Example 2: Performance issue: TextArea very slow when accessibility API turned on[2]

  While the first bug report describes a missing textual label in a graphical component, making it not accessible for blind users, the second bug report is related to a performance issue. Despite containing the keyword accessibility, this bug is not related to the accessibility of the software but to a performance regression detected when integrating the accessibility library, through its API, to the system.

[1]https://bugzilla.mozilla.org/show_bug.cgi?id=533573

[2]https://bugs.chromium.org/p/chromium/issues/detail?id=868830

These examples show that we cannot rely on the keyword accessibility to identify accessibility-related bug reports, as the first example (accessibility bug report) did not contain the keyword *accessibility*, while the second example (non-accessibility bug report) did.

1.2. Research Goals

The goal of this thesis is to support and encourage accessibility adoption in both educational frameworks and industrial projects. We do this by revealing the challenges that educators, students, and practitioners face when it comes to adopting accessibility tools in their software systems. Therefore, we performed a mixed methods study to address our goal, which involves both quantitative and qualitative analysis. We plan to survey students and teachers about their challenges, particularly when they were forced to learn online during the COVID-19 pandemic. Furthermore, research shows that developers create inaccessible systems because of insufficient technical skills, ignorance of accessibility guidelines, and a lack of awareness of the essence of creating accessible systems [111]. As a result, we also plan to shed light on all the different errors and defects that developers experience when they integrate accessibility guidelines into their modern software systems. To cope with the above-mentioned challenges, throughout this research project, we aim to achieve the following research goals:

- **Goal 1:** *To identify the accessibility problems and challenges faced by students.*

  We aim to investigate the accessibility challenges faced by the education sector when using the software. To achieve this goal, we will survey students to get their views and input regarding their challenges. Furthermore, we conduct sentiment analysis and user reviews to understand what the users of educational software think about the accessibility of the systems. Therefore, we will use multiple experiments to indicate the accessibility of education systems from the users' perspectives.

- **Goal 2:** *To provide developers with insights on ensuring software accessibility.*

  We aim to provide the software development industry with recommendations on how accessibility can be achieved. To do that, we will highlight the different errors and defects that developers experience with creating software and their adherence to recommended accessibility guidelines. We will use machine learning techniques to develop tools to support the automated identification of accessibility in user reviews, to help technology professionals prioritize their handling and, thus, create more inclusive apps. Particularly, we design a model that takes as input accessibility user reviews and learns their keyword-based features to make a binary decision, for a given review, on whether it is about accessibility or not.

A schematics summary of the most important research activities conducted during the Ph.D. trajectory is presented in Figure 1.1. It provides the timeline of the Ph.D. project containing an overview of the time frame of different articles' submission and acceptance.

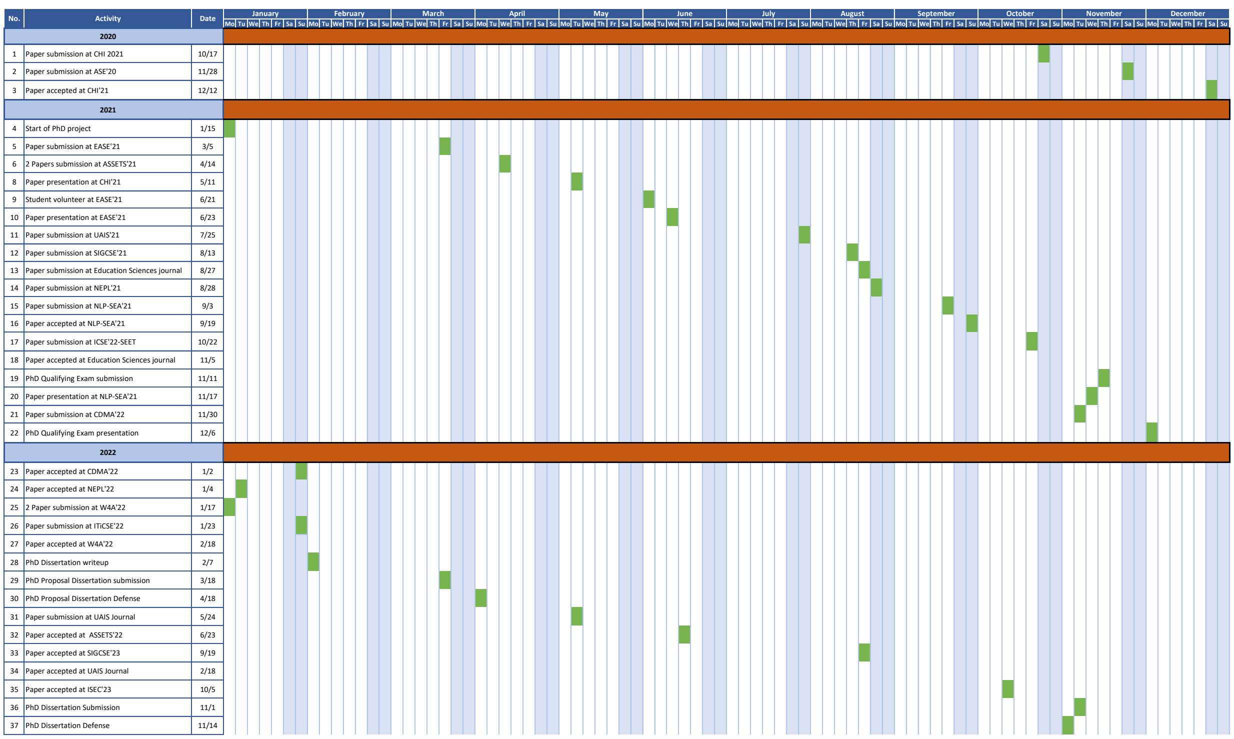

FIGURE 1.1. Research schedule.

1.3. Ph.D. Publications

This section outlines our achieved contributions as part of the Ph.D. work.

(1) **Aljedaani, Wajdi**, Rrezarta Krasniqi, Sanaa Aljedaani, Mohamed Wiem Mkaouer, Stephanie Ludi, and Khaled Al-Raddah. *If online learning works for you, what about deaf students? Emerging challenges of online learning for deaf and hearing-impaired students during COVID-19: a literature review.* Universal access in the information society (2022): 1-20 [UAIS] [27].

(2) **Aljedaani, Wajdi**, Mohamed Wiem Mkaouer, Stephanie Ludi, and Yasir Javed. "Automatic Classification of Accessibility User Reviews in Android Apps." In 2022 7th International Conference on Data Science and Machine Learning Applications (CDMA), pp. 133-138. IEEE, 2022 [CDMA] [28].

(3) **Aljedaani, Wajdi**, Mohamed Wiem Mkaouer, Stephanie Ludi, Ali Ouni, and Ilyes Jenhani. "On the identification of accessibility bug reports in open source systems." In Proceedings of the 19th International Web for All Conference, pp. 1-11. 2022 [W4A] [29].

(4) Amaar, Aashir, **Wajdi Aljedaani**, Furqan Rustam, Saleem Ullah, Vaibhav Rupapara, and Stephanie Ludi. *Detection of Fake Job Postings by Utilizing Machine Learning and Natural Language Processing Approaches.* Neural Processing Letters, 30 page (2022) [NPL] [52].

(5) **Aljedaani, Wajdi**, Furqan Rustam, Mohamed Wiem Mkaouer, Abdullatif Ghallab, Vaibhav Rupapara, Patrick Bernard Washington, Ernesto Lee, and Imran Ashraf. "Sentiment analysis on Twitter data integrating TextBlob and deep learning models: The case of US airline industry." Knowledge-Based Systems 255 (2022): 109780 [KBS] [34].

(6) Rupapara, Vaibhav, Furqan Rustam, **Wajdi Aljedaani**, Hina Fatima Shahzad, Ernesto Lee, and Imran Ashraf. *Blood cancer prediction using leukemia microarray gene data and hybrid logistic vector trees model.* Scientific Reports, 15 pages, 2022 [Scientific Reports] [260].

(7) Rustam, Furqan, Aijaz Ahmad Reshi, **Wajdi Aljedaani**, Abdulaziz Alhossan, Abid Ishaq, Shabana Shafi, Ernesto Lee et al. *Vector mosquito image classification using*

novel RIFS feature selection and machine learning models for disease epidemiology. Saudi journal of biological sciences, 12 pages, 2022 [SJBS] [263].

(8) **Aljedaani, Wajdi**, Mona Aljedaani, Eman Abdullah AlOmar, Mohamed Wiem Mkaouer, Stephanie Ludi, and Yousef Bani Khalaf. *I Cannot See You—The Perspectives of Deaf Students to Online Learning during COVID-19 Pandemic: Saudi Arabia Case Study.* Education Sciences, 24 pages, 2021 [Education Sciences] [21].

(9) **Aljedaani, Wajdi**, Furqan Rustam, Stephanie Ludi, Ali Ouni, and Mohamed Wiem Mkaouer. *Learning Sentiment Analysis for Accessibility User Reviews.* In 2021 36th IEEE/ACM International Conference on Automated Software Engineering Workshops (ASEW), 2021 [NLP-SEA] [33].

(10) **Aljedaani, Wajdi**, Anthony Peruma, Ahmed Aljohani, Mazen Alotaibi, Mohamed Wiem Mkaouer, Ali Ouni, Christian D. Newman, Abdullatif Ghallab, and Stephanie Ludi. *Test Smell Detection Tools: A Systematic Mapping Study.* Evaluation and Assessment in Software Engineering, 2021 (acceptance rate: 23%) [EASE] [32].

(11) AlOmar, Eman Abdullah, **Wajdi Aljedaani**, Murtaza Tamjeed, Mohamed Wiem Mkaouer, and Yasmine N. El-Glaly. *Finding the Needle in a Haystack: On the Automatic Identification of Accessibility User Reviews.* In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, 2021 (acceptance rate: 26.3%) [CHI] [40].

(12) Ye, Xin, Yongjie Zheng,**Wajdi Aljedaani**, and Mohamed Wiem Mkaouer. *Recommending pull request reviewers based on code changes.* Soft Computing, 15 pages, 2021 [Soft Computing] [322].

(13) Fang, Fan, John Wu, Yanyan Li, Xin Ye, **Wajdi Aljedaani**, and Mohamed Wiem Mkaouer. *On the classification of bug reports to improve bug localization.* Soft Computing, 18 pages, 2021 [Soft Computing] [117].

(14) Alkhazi, Bader, Andrew DiStasi, **Wajdi Aljedaani**, Hussein Alrubaye, Xin Ye, and Mohamed Wiem Mkaouer. *Learning to rank developers for bug report assignment.* Applied Soft Computing, 15 pages, 2020 [Applied Soft Computing] [37].

(1) **Aljedaani, Wajdi**, Mona Aljedaani, Eman Abdullah AlOmar, Mohamed Wiem Mkaouer, Stephanie Ludi, and Yousef Bani Khalaf. *I Cannot See You—The Perspectives of Deaf Students to Online Learning during COVID-19 Pandemic: Saudi Arabia Case Study.* Education Sciences, 24 pages, 2021 [Education Sciences] [21].

(2) AlOmar, Eman Abdullah, **Wajdi Aljedaani**, Murtaza Tamjeed, Mohamed Wiem Mkaouer, and Yasmine N. El-Glaly. *Finding the Needle in a Haystack: On the Automatic Identification of Accessibility User Reviews.* In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, 2021 (acceptance rate: 26.3%) [CHI] [40].

(3) **Aljedaani, Wajdi**, Furqan Rustam, Stephanie Ludi, Ali Ouni, and Mohamed Wiem Mkaouer. *Learning Sentiment Analysis for Accessibility User Reviews.* In 2021 36th IEEE/ACM International Conference on Automated Software Engineering Workshops (ASEW), 2021 [NLP-SEA] [33].

(4) **Aljedaani, Wajdi**, Rrezarta Krasniqi, Sanaa Aljedaani, Mohamed Wiem Mkaouer, Stephanie Ludi, and Khaled Al-Raddah. *If online learning works for you, what about deaf students? Emerging challenges of online learning for deaf and hearing-impaired students during COVID-19: a literature review.* Universal access in the information society (2022): 1-20 [UAIS] [27].

(5) **Aljedaani, Wajdi**, Mohamed Wiem Mkaouer, Stephanie Ludi, and Yasir Javed. "Automatic Classification of Accessibility User Reviews in Android Apps." In 2022 7th International Conference on Data Science and Machine Learning Applications (CDMA), pp. 133-138. IEEE, 2022 [CDMA] [28].

(6) **Aljedaani, Wajdi**, Mohamed Wiem Mkaouer, Stephanie Ludi, Ali Ouni, and Ilyes Jenhani. "On the identification of accessibility bug reports in open source systems." In Proceedings of the 19th International Web for All Conference, pp. 1-11. 2022 [W4A] [29].

1.4. Organization of the Dissertation

The thesis is organized as follows: Chapter 2 provides a background of this thesis, while Chapter 3 introduces the literature review. In Chapter 4, we discuss our contribution of highlighting high-demanding issues that deaf students experienced in higher education during the pandemic. Chapter 5 explores the perspectives of deaf students on online learning during the COVID-19 pandemic. Chapter 6 investigates the status of accessibility in the Blackboard mobile application. Chapter 7 presents our proposed approach to automate the detection of accessibility user reviews. Chapter 8 presents our proposed approach to automate the detection of accessibility based on the WCAG guideline. Chapter 9 describes our approach to learning sentiment analysis for accessibility user reviews. Chapter 10 presents our contribution to automatically identifying accessibility bug reports in open-source systems. The implication of the study is discussed in Chapter 11. In Chapter 12, we discuss the threats to the validity of our work. Finally, a summary and future research directions are presented in Chapter 13.

CHAPTER 2

BACKGROUND

In this chapter, we introduce some terminologies and concepts that have been used in Ph.D. research. We contextualize our work with respect to literature around web accessibility, mobile accessibility standards/guidelines, mobile learning and LMS, and the Case Study of Technical and Vocational Training Corporation (TVTC).

2.1. Web Accessibility

As the internet becomes a more integrated part of the global culture, web accessibility has increased in importance. Expanding access to disabled people who are unable to use a mouse, have sight impairments or color blindness, or otherwise cannot use an internet browser in a traditional way. Steps towards a solution were taken by publishing standards that web developers would follow and check their web pages against to see if they had created a web page that was accessible. The accessibility benchmark became a way to ensure that all users could see and enjoy the content and a sign of quality and excellence on the part of the web developer. Responsive web design, having the web page flex to be used in any size screen or device, used to be separate from accessibility. Still, with the increasing assistive technologies using non-standardized screen sizes and employing the use of touch screens, responsiveness became just another part of being accessible. The standards of accessibility are based on the WCAG, Section 508 & 504, and the requirements of the ADA (Americans with Disabilities Act).

2.1.1. Web Content Accessibility Guidelines (WCAG)

First published in May 1999, the Web Content Accessibility Guidelines (WCAG)[1] is a project started by the Web Accessibility Initiative, a subset of the World Wide Web Consortium (W3C)[2] that is dedicated to making the internet more usable for people with disabilities. In 1997 the White House endorsed the Web Accessibility Initiative, and the

[1]https://www.w3.org/WAI/standards-guidelines/wcag/

[2]https://www.w3.org/

group was started. WCAG 2.0 was not published until December 2008 and did not become an International Organization for Standardization until 2012.

W3C was published in 2018, introducing the priority of users with vision impairments rather than blind users, and emphasizing responsive design for mobile devices. The newest publication of the WCAG is the 2.2 working draft published in August 2020 [7]. The guide intends to improve access to a broader range of disabilities while also using language that is not overly technical so as to be able to access the results more easily. The document suggests the most accessible options with the notation (AAA). The second-best options are shown as (AA), and the third-most acceptable but still accessible options are shown with (A). The technical changes range from aria labels that translate text into a format that can be read by a screen reader to standards that make videos easier for hearing-impaired users to understand.

The WCAG guidelines include a list of identical detailed techniques that individuals might use in order to achieve the requirements. Additionally, it covers common failures. W3C updates the documentation as technologies develop without affecting the document's overall structure. Figure 2.1 presents an overview structure of WCAG principles and guidelines.

- Adding alternative text to images.
- Typing text labels into form inputs.
- Adding titles to frames and iframes.
- Ensuring that all videos have transcripts or captions.
- Using headings such as H1, H2, H3 in a hierarchical order without skipping numbers.

2.1.2. Section 508 and 504

In 1998 Section 508 was added to the Americans with Disabilities Act[3], ensuring access to electronic and information technology. The most significant influence this section has had on the internet is requiring federal websites to make sure their text can be read by a screen reader, accessible software used by sight-impaired and blind individuals to read the

[3] https://www.ada.gov/

text on the screen audibly. Section 508 did allow for one loophole, suggesting that these accommodations did not have to be made if such efforts were an undue burden.

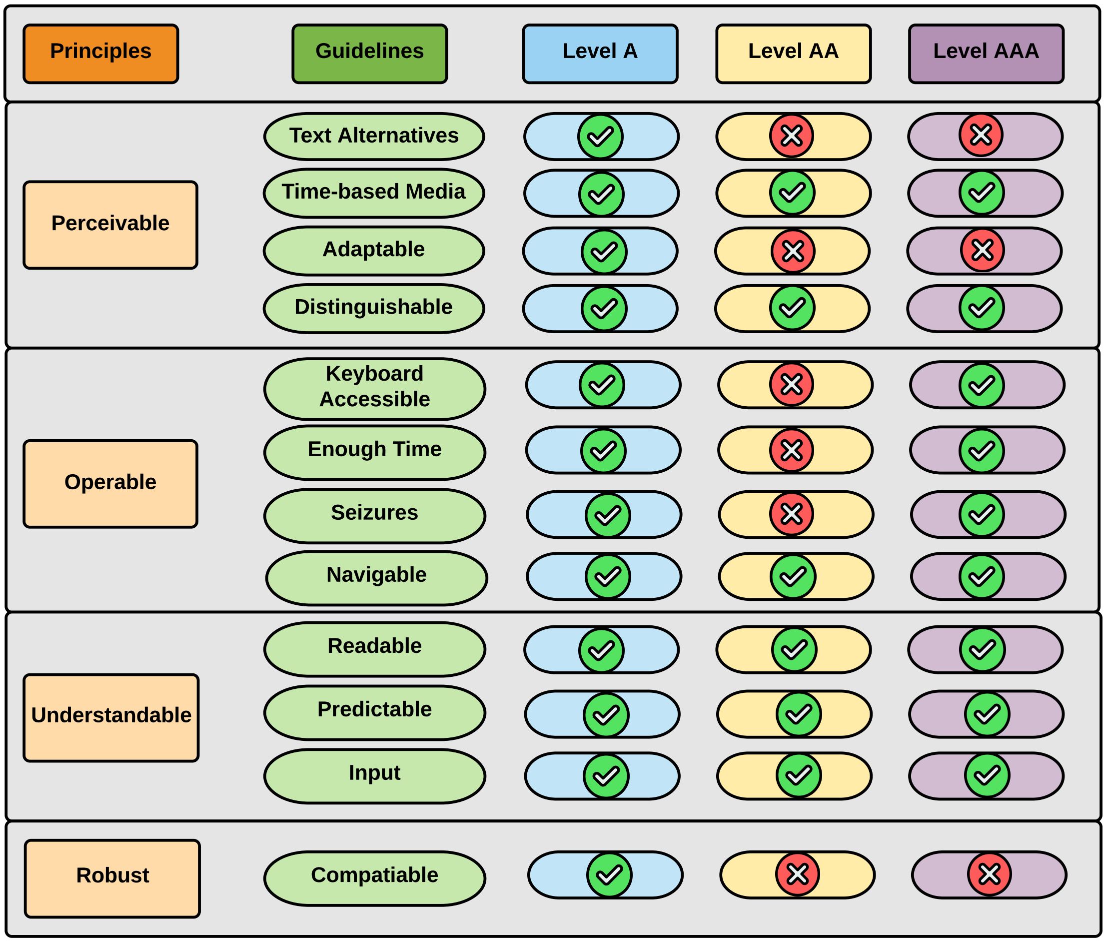

FIGURE 2.1. Overview structure of WCAG principles and guidelines.

The Americans with Disabilities Act of 1990 (ADA) is a civil rights law protecting people with disabilities against discrimination, which also applies to K-12 and higher education students. The two sections of the ADA that apply to web accessibility are Title II " prohibits discrimination against qualified individuals with disabilities in all programs, activities, and services of public entities..." and Title III "prohibits discrimination on the basis of disability in the activities of places of public accommodations" [6]. While the ADA was written without the consideration of web accessibility, it applies to the places and spaces used to communicate with all people, such as government websites. The document has been used in lawsuits to qualify the complaint of a disabled person to use a government website.

Section 504, the Rehabilitation Act of 1973, was the basis for the ADA, preventing prejudice against disabled individuals who were receiving federal funding. The way this factored into schools would be how tests were administered and standardized testing was considered. Learning disabilities, hyperactivity, and anxiety were considered along with disabilities such as audible, visual, and motor function.

Web accessibility is a sign of good web design, a collective effort of many good practice suggestions, and a process that has developed over time from both government law and nonprofit organizations with the intent to make the internet a place for people from all backgrounds and ability levels to access.

2.2. Mobile Accessibility Standards/Guidelines

Apps accessibility in mobiles is controlled by benchmarks and standards stipulated by the World Wide Web Consortium (W3C)[4]. The W3C, through the Web Accessibility Initiative (WAI), provides a range of guidelines and standards that are frequently updated to address any emerging issues in accessibility. The guidelines mainly include Web Content Accessibility Guidelines (WCAG), User Agent Accessibility Guidelines (UAAG), Accessible Rich Internet Applications (WAI-ARIA), and Authoring Tool Accessibility Guidelines (ATAG). Although there are many standards on accessibility, those that relate to mobile accessibility are yet to be developed, meaning that the current guidelines in use are the W3C through the WCAG 2.0 principles. Such rules apply to native apps, mobile web apps, and mobile web content, among others. WCAG defines accessibility in terms of four principles which include ease of operation (operable), understandable app content (understandable), robustness (robust), and coherent app content (perceivable) [273].

Other than the standards provided by W3C, other independent organizations, such as the British Broadcasting Corporation (BBC) have drafted their own accessibility standards. A document called BBC Mobile Accessibility Guidelines by the BBC contains these guidelines [63]. The rules contained in the BBC guidelines are similar to those provided by

[4] https://www.w3.org/

the W3C [311]. These standards mainly guide on principles, audio and video, designs of the apps, focus, forms, images, links, notifications, scripts, and dynamic content, structure, and text equivalence, as described in detail in Table 8.1.

Although BBC accessibility guidelines provide detailed instructions, there are additional standards given by WCAG 2.0 in relation to mobile accessibility. Some of the additional guidelines include rules on how to reduce content in the mobile version, eliminating form fields that are beside their labels, accessibility of interactive controls, noticeability of apps content, text minimization, provision of clear guidelines, and adjustment of the app to the different orientation of the device. The accessibility guidelines provided by BBC and WCAG are crucial. However, other organizations may also come up with accessibility guidelines to complement the existing ones.

2.3. Mobile Learning and LMS

With the increasing need to offer online education by universities and institutions around the world, the adoption of learning management systems (LMS) has also increased [275]. Indeed, mobile learning offers several benefits, including location-based services, cost-effectiveness, and education aid, among others. It has also been considered that LMS systems help in improving students' problem-solving skills, performance, and knowledge and create an individualized learning system [275, 141]. There are five authoring tools that are considered part of a learning management system: content collaboration, content delivery, content development, content distribution, and content management [141]. During the current Coronavirus (COVID-19) pandemic, the need to reduce physical interaction in higher education institutions has increased the adoption of LMS in order to facilitate mobile learning [96]. Hence, LMS systems have helped many institutions deliver instruction exclusively online, where content is managed, distributed, and delivered to students.

2.3.1. Blackboard

The Blackboard LMS provides a personalized intuition that helps learners to engage with their tutors, provides data handling capabilities, and is flexible to any teaching approach

[70]. Blackboard was developed by Matthew Pittinsky and Michael Chasen in 1997. It is considered an excellent LMS system because it is easily available on many devices, provides quick feedback, makes tracking easier, and has better communication [16, 30]. Learning institutions can choose between two Blackboard systems: Networked Transaction Environment (NTE), which helps in supporting commercial transactions, and the Networked Learning Environment (NLE), which offers academic capabilities that support online learning. Blackboard is one of the most popular LMS systems today and is used by very many institutions globally.

TABLE 2.1. List of the keywords used to identify user reviews refer to accessibility. We followed the BBC standards and guidelines for mobile accessibility [63].

Guideline	Description	Relevant Keywords
Principles	These guidelines require a focus on three principles of developing usable and inclusive applications. First, developers should utilize all web standards as required. Secondly, there should be utilization of interact controls. Thirdly, content and functionality in the app should support native features of the app.	Accessibility, disability, screen reader, blind talkback, operable, impaired, impairment
Audio/video	Applications should provide alternative formats such as transcripts, sign language, or subtitles. Autoplay should be disabled, and the user should be provided with play/pause/stop or mute buttons to control audio. There should be no conflict between audio in application media of native assistive technology.	Subtitle, sign language, audio description, transcript, autoplay, mute, volume, can't hear
Design	The color in the app background should have appropriate contrast, and touch targets must be large enough to be touched effectively. Visible state change should be experienced in every item in the app that has been focused on. Unnecessary or frequent flickering of content must be avoided.	Contrast, background color, flicker, visual cue, touch size, overlap, font size, dark/light mode, eyestrain, seizure, can't see
Focus	There should be a logical organization of items, and users should be offered alternative input methods. Interactive and inactive elements should be focusable and non-focusable, respectively. Keyboard traps should be eliminated, and focus should not change suddenly when the app is utilized.	Focusable, control focus, keyboard trap, focus order, navigable, input/type
Forms	Every form of control must have a label. All labels must have a logical grouping, and a default input format must be given. Labels should be close to their form controls.	Unique label, missing label, visible label layout, voice-over
Images	Text images should not be included. Any background images that have content should have another accessible alternative.	Image of text, hidden text, text alternative, background image
Links	Any navigation links must indicate the function of the link. If a link to an alternative format is clicked, the user should be notified of the redirection to the alternative. Several links that redirect to the same source should be put together in one link.	Link description, unique desc., duplicate link, alternative format
Notifications	Error messages should be clear. Any notifications given must be easily seen or heard. There should be standard system notifications where necessary.	Operating inclusive, haptic, vibration, feedback, alert dialog, understandable, unfamiliar
Dyn. content	Applications should be made in a progressive manner that enables every user to benefit from them. Appropriate notifications should be given for automatic page refreshes. Flexible interaction input control must be given.	Animated content, page refresh, automatic refresh, timeout, adaptable, input sign
Structure	Every page on the application should be uniquely identified. Content should be arranged in a hierarchical and logical manner with appropriate headings. One accessible component should be used to group interface objects, controls or elements.	Page title, screen title, heading, header unique descriptive
Text equivalent	Applications should give the objective of a specific image or its editorial aim. In addition, visual formatting must be complemented by other ways to give meaning. There should be no conflict between decorative images with assistive technology. Every element must have well-placed and effective accessibility properties.	Alternative text, non-visual, content description decorative content, no-text-content

2.4. Case Study of Technical and Vocational Training Corporation (TVTC)

The Technical and Vocational Training Corporation (TVTC) is a public tertiary education institution in the Kingdom of Saudi Arabia, which was established in 1980. TVTC provides vocational education and training, making it important in workforce development. It consists of three sectors: vocational training centers, technical colleges, and secondary institutions, whose numbers are 65, 35, and 35, respectively. The TVTC has also provided accreditation to approximately 1000 private institutions. Thus, the TVTC is very instrumental in KSA's tertiary education and greatly contributes in providing labor supply to national and international labor markets [302].

The Technical and Vocational Training Corporation (TVTC) is a public tertiary education institution in the Kingdom of Saudi Arabia, which was established in 1980. TVTC provides vocational education and training, making it important in workforce development. It consists of three sectors: vocational training centers, technical colleges, and secondary institutions, whose numbers are 65, 35, and 35, respectively. The TVTC has also provided accreditation to approximately 1000 private institutions. Thus, the TVTC is very instrumental in KSA's tertiary education and greatly contributes in providing labor supply to national and international labor markets [302, 22].

2.4.1. Deaf Education in TVTC

There are four branches that deaf students can go to in TVTC, where they can access special education programs. The branches are distributed in the KSA, particularly in the middle and central areas. The numbers of such institutions, with their locations, are as follows: (1) Riyadh, (2) Madinah, (3) Buraydah, and (4) Dammam. All these branches teach two majors: business and computer technology. The number of students is almost less than 100 at each campus. However, deaf students in the institutions are not studying with other students without disabilities. According to Özokcu and Yildirim [235], disabled students are afraid to learn in inclusive classes because they are treated differently from others. In terms of teachers, the majority of the teachers are specialized (for specials educations), except for the general courses, such as English, Mathematics, or other classes. Currently, most teachers are working without interpreters because they are conversant with sign language. For those teachers that are not familiar with sign language, they seek the assistance of interpreters. Currently, there are only three interpreters in the department because there are only a few students. If deaf students need assistance in any of the college services, they contact the department to provide them with an interpreter. Mostly, if students are holding seminars with spoken lecturers, they have to request an interpreter before the session.

2.4.2. Learning Process at TVTC

The normal education process in Saudi Arabia involves face-to-face learning, which entails training in the classroom that has been conducted since the 1950s [266]. Face-to-face learning includes in-person lectures and textbook readings, which were mostly preferred in the pre-pandemic period because they emphasize human-human interactions [17]. With the advent of technology, blended learning was introduced in Saudi Arabia, where face-to-face interactions were complemented by technology. For example, as of 2016, TVTC had introduced a Learning System Management (LMS) known as Doroob to make learning more interactive and student-centered [14]. The learning process at TVTC in the pre-pandemic period was as follows:

- Students did not have experience on how to use the full functionality of Blackboard, except to review course materials [172].
- Students used to give their homework and projects as hand-outs or use Dropbox to submit them. For project courses, students used to email teachers about any updates and the final submission of the course delivery.
- Students used Rayat (a portal that enables students/trainees to obtain many services such as tracking training records and attendance and grades, etc.) [298] to access their grades, personal information, and the process of attending their courses.

2.4.3. Learning Management Systems at TVTC

Learning Management Systems (LMSs) have become very popular in modern universities because of their ability to deliver content remotely, enhance interactions, improve feedback, and provide analytics to teachers to assess performance [16, 20, 51]. The first LMS system was known as FirstClass and was developed in 1990 [16]. Some of the most popular LMS applications in Saudi Arabia today are: Blackboard (89% popularity), Moodle (7% popularity), and D2L (4% popularity) [16]. Therefore, Blackboard is the most popular LMS application in institutions of higher learning in Saudi Arabia.

In 2016, TVTC introduced an LMS system known as Doroob to make learning more interactive, and student-centered [14]. Before COVID-19, Saudi institutions, such as TVTC and King Saud University, were gradually adopting Blackboard LMS in order to improve their online learning channels [51]. However, the disruption of learning brought by COVID-19 in Saudi Arabia led to a sudden and quick shift towards Blackboard LMS [38].

2.5. Chapter Summary

: This chapter covered Ph.D. research terminology. Web accessibility, mobile accessibility standards and guidelines, mobile learning and learning management systems, and the Technical and Vocational Training Corporation (TVTC) case study are used to contextualize our work. The chapter did not contain relevant studies pertinent to this dissertation's purpose.

Next chapter, we provide related work. We discuss various research that shaped our technique. The literature review has seven sections: (1) research solely focused on deaf students during COVID-19, (2) e-learning system for deaf students, (3) deaf students in online learning, (4) accessibility with deaf students, (5) user reviews in app evolution, (6) accessibility detection in user reviews, and (7) text document categorization.

CHAPTER 3

RELATED WORK

This section provides the related work. We highlight several previous works that profoundly influenced our approach. We split the related works into seven sections: (1) studies exclusively focused on deaf students during COVID-19, (2) e-learning system for deaf students, (3) deaf students in online learning, (4) accessibility with deaf students, (5) user review, which briefly highlights the role of user reviews in app evolution, (6) accessibility in a user review, focuses particularly on detection of accessibility in user reviews, and (7) classification of text documents, where we focus on current approaches in the classification of text such as user reviews by different taxonomies. Table 3.1 presents a summary of systematic analysis studies investigating on deaf and hard of hearing during the COVID-19 pandemic.

3.1. Deaf Students During COVID-19 Studies

Several studies have addressed the subject of deaf and hard of hearing (DHH) studies. For instance, Kritzer and Smith [169] conducted a survey in the United States, which emphasized the need for parents to seek appropriate learning services and opportunities for their DHH children and communicate with them. Another study by Smith and Colton [280] utilized a literature review method and proposed using YouTube channels in teaching DHH students during the COVID-19 pandemic. The authors demonstrated how YouTube videos were made, and shared and their usefulness in educating DHH students. Further, Sutton [289] utilized a literature review methodology and evaluated best practices to help DHH students during COVID-19, such as providing speech-to-text services and facilitating communication through accommodation. Research by Lazzari and Baroni [171] investigated remote teaching experiences in Italy and found that remote teaching using technology helped to learn during COVID-19, although some challenges were experienced, such as inadequate materials. Another study by Alsadoon and Turkestani [46] sought to identify the obstacles to e-learning and found that technical problems, time, and translation problems were severe challenges to DHH distance learning.

Research by Fernandes et al. [119] in Indonesia investigated how deaf voters were educated and indicated that videos effectively conducted voter education in Indonesia and visual and social media. Furthermore, in their study, Lynn et al. [191] evaluated how students were learning chemistry during the pandemic and found that access services such as interpreters and captioners were vital in DHH education and making sufficient accommodations to ensure inclusion. Another research by Swanwick et al. [290] was conducted in Ghana to determine how the pandemic had affected deaf education and found that exclusion for DHH students was different in various cultural contexts and developmental areas. Research by Paatsch and Toe [237] utilized a literature review methodology and indicated that DHH students in typical classrooms developed pragmatic skills and proposed using the conversation model to deal with the challenges faced by such students. Finally, Tomasuolo et al. [297] conducted exploratory research in Italy and found that initiatives at the political and informal levels promoted sign language and assisted in DHH education during the pandemic.

3.2. E-Learning System for Deaf Students

Previous studies have suggested the importance of introducing e-learning systems for deaf students. For example, a study by Alcazar et al. [15] found that introducing a speech-to-visual approach e-learning system had a great advantage when teaching deaf students because it enabled their comprehension of material and addressed their individual needs. Furthermore, a study by Batanero et al. [60] found that adopting an improved Moodle learning platform improved the academic performance of deaf and deaf-blind students by 46.25% and deaf-blind students by 87.5%. In addition, Batanero-Ochaita et al. [61] found that deaf students had a positive attitude towards the Moodle Learning Platform, although their perceptions differed on the ease-of-use and difficulty when using the platform.

3.3. Deaf Students in Online Learning

Several studies have looked at the subject of online learning for deaf students. For example, a study by Long et al. [188] focused on blended learning for deaf and hard-of-hearing students and found that the inclusion of online learning aspects improved their interactions with their teachers and other students. Another study by Slike et al. [279] found that, although there are many successes in teaching deaf students using online means, there are also challenges related to system ‘glitches’, lack of captions, teachers who are not used to handling deaf students in virtual classrooms, among others. In addition, a study by Yoon and Kim [325] suggested the need to improve learning materials in the classroom because it established that captions have a significant effect on the content comprehension of deaf students taking online courses.

A study by Burgstahler [82] was conducted to identify the online learning practices that are most suitable for students with disabilities and found the “Universal Design” (UD) strategy to be very effective in inclusive educational practices. Research by McKeown [206] found three types of challenges that faced deaf students when accessing online learning: course content and material challenges, learning management system (LMS) challenges, and course content and material challenges. A study by Carpenter et al. [99] established that online technology had improved deaf education and made communication easier and proposes the use of best practices that can boost the utilization of online learning. Another research by Mohammed [212] found that deaf students participating in online education faced problems related to institutional support, social inequalities, and inappropriate sociolinguistic history. Another study by Musyoka and Smith [223] found that, since online deaf learning involves the use of English and American Sign Language (ASL), language barriers were considered as a challenge. A study by Long et al. [187][12] reported that online learning for deaf students provides special benefits that were realized through academic achievement and the quality of interaction in online learning platforms greatly determined the success of the students. Additionally, a study by Caupayan and Pogoy [85] established that, although deaf students faced challenges in online learning, the support they received from various stakeholders helped them to overcome them. From the related work, there is no study addressing online learning for deaf students in Saudi Arabia, which is the focus of the current study.

Table 3.1. Summary of systematic analysis studies in the related work for deaf and hard of hearing students sorted by the year.

Study	Year	Purpose	Method	Source of Info	Participants	Sample Size	Study Location
Long et al. [188]	2007	Understanding student perceptions of communication in blended learning courses	Survey	NTID	Students	908	United States
Slike et al. [279]	2008	Investigating successes and challenges in offering online courses in a "virtual classroom" format to deaf hard of hearing	Online Synchronous Tool	University of Pennsylvania	Students	26	United States
Yoon and Kim [325]	2011	Capturing the effects of captions on deaf students' content comprehension, cognitive load and motivation in online learning	Comprehension test & survey	Korean Nazarene University	Students	62	South Korea
Burgstahler [82]	2015	Identifying online learning practices make social inclusion possible for individuals with disabilities	Literature review	N/A	N/A	N/A	N/A
Alcazar et al. [15]	2016	Creating a supplementary English elearning system made for the Deaf	Survey	Philippine Institute of the Deaf	Students, Teachers	N/A	Philippines
McKeown [206]	2019	Proposing a model which describes the three barriers deaf students might encounter in an online learning situation	Literature review	N/A	N/A	N/A	N/A
Batanero et al. [60]	2019	Testing a redesign of the Moodle platform on deaf and deaf-blind students	Empirical study	Moodle platform	Students	23	N/A
Counselman et al. [99]	2020	Exploring current trends in online higher education, data on the experience of Deaf/deaf/Hard of Hearing students and current options for improving inclusively in the online classroom	Literature review	N/A	N/A	N/A	N/A
Fernandes et al. [119]	2020	Examining how the education of voters for deaf people	Interview	GERKATIN	Students	33	Indonesia
Lynn et al. [191]	2020	Identifying Successes and challenges in teaching chemistry	Author's insights	NTID	N/A	N/A	United States
Tomasuolo et al. [297]	2020	Exploring the impacts of the recent pandemic crisis	Exploratory research	Literature review	N/A	N/A	Italy
Smith and Colton [280]	2020	Developing a YouTube channel that focuses on providing	Literature review	Author's experience	N/A	N/A	United States
Alsadoon and Turkestani [46]	2020	Identifying the lecturers' obstacles during virtual classroom	Interview	King Saud University	lecturers	11	Saudi Arabia
Kritzer and Smith [169]	2020	Recommending parents about educating DHH children	Survey	United States	Parents	133	United States
Swanwick et al. [290]	2020	Investigating the impact on deaf adults, children and their families, focusing on issues of inclusion	Interview	Ghana	Teachers, leaders, Students	6	Ghana
Lazzari and Baroni [171]	2020	Presenting the remote teaching experience	Survey	Scuola Audiofonetica	Students	233	Italy
Paatsch and Toe [237]	2020	Investigating pragmatic skills among deaf children	Literature review	Existing evidence	N/A	N/A	Australia
Krishnan et al. [168]	2020	Identify the Challenges Faced by hearing impairment	Survey	MySkill Foundation	Students	10	Malaysia
Halley Sutton [289]	2020	Suggesting best practices if courses shift to online learning	Literature review	Existing evidence	N/A	N/A	United States
Kylie Sommer [284]	2020	Identify the Effect on deaf and hard of hearing	Survey	Lee University	Students	36	United States
Mantzikos and Lappa [196]	2020	Analyzing the difficulties and barriers individuals deaf	Literature review	Existing evidence	N/A	N/A	Greece
Mohammed [212]	2021	Investigating how an emergent system of e-learning that started during crisis conditions affects the linguistic access of deaf students	Interview	Primary school & secondary school	Students, teachers, interpreters, parents	N/A	Trinidad and Tobago
Batanero-Ocha et al. [61]	2021	Analyzing the difficulties and barriers individuals deaf	Empirical study	Moodle platform	Students	23	N/A
Musyoka and Smith [223]	2021	Identifying language barriers and academic performance when discussing mainstreamed D/HH students' online teaching	Literature review	Existing evidence	N/A	N/A	United States
Long et al. [187]	2021	Understanding the factors contributing to the academic achievement and the interaction of students in online learning	Survey	RIT	Students	88	United States
Caupayan and Pogoy [85]	2021	examining and interpreting the lived experiences of 14 purposively selected deaf students who chose online modality for their education	Interview	La Salle University	Students	14	Philippines
This work	**2021**	**Investigating the challenging and concerns of deaf students**	**Interview & Survey**	**TVTC**	**Students**	**65**	**Saudi Arabia**

3.4. Accessibility with Deaf Students Studies

Several studies have addressed the topic of accessibility with deaf students. For example, a study by Sommer [284] in the US utilized a survey method to demonstrate how access to information by DHH students has been hampered by the COVID-19 pandemic, which has had emotional effects on them. Another study by Mantzikos and Lappa [196] reviewed existing evidence on overcoming difficulties and barriers to deaf education. It established the use of media, such as educational TV programs, helped improve access to information, although it was essential to improve new principles and approaches that helped DHH students. Research by Krishnan et al. [168] in Malaysia used a survey method to investigate students' challenges during Covid-19 and found that accessibility by DHH students was hindered by lack of familiarity to online devices, hearing devices, emotional effects of the pandemic, and disruptions to their education.

The current study deviates from those reviewed because it focuses on accessibility by deaf students in Saudi Arabia. Specifically, it focuses on TVTC in Saudi Arabia which any other study has not addressed. TVTC has adequate and trained faculty with a lot of experience who offer more support for the deaf. This research also differs from the rest in terms of the methodology and because this study is a larger scale research in terms of the sample compared to the previous ones.

3.5. User Reviews

Many researchers concluded that reviews and ratings posted by users on app store platforms could play an essential role in apps' evolution since most developers consider users' reviews when working on a new release [91, 181, 238, 245, 23, 264, 31, 229]. Maalej et al. [192] proposed to consider user input as the first means of requirements elicitation in software development. Similarly, Vu et al. [309] emphasized the role of users in the software lifecycle by developing an approach to identify useful information from users' reviews. Moreover, Seyff et al. [271] suggested continuous requirements elicitation from end-users feedback using mobile devices.

Considering the fact that user reviews can be a powerful driver of mobile app evolution, we are looking into whether we can effectively detect accessibility reviews from users' feedback. This is important because in a highly competitive market, identifying accessibility issues from users' reviews can help developers improve their apps in order to attract more customers and provide better services to users with different abilities.

It is crucial that mobile applications be accessible to allow all individuals with different abilities to have fair access and equal opportunities [144]. Prior studies investigated the accessibility issues raised in Android applications [48, 307], and others evaluated the accessibility of various websites [9, 107, 158, 315, 126]. To the best of our knowledge, there is no study that classifies user reviews in Android applications using machine learning.

3.6. Accessibility in User Reviews

Even though user reviews can be a robust tool for mobile apps evolution, and even mature apps have many trivial accessibility issues [114, 319], only 1.24% of mobile app users report accessibility issues to app stores [112]. In other words, 98.76% of mobile app users do not post accessibility issues in the form of reviews on app stores. In an effort to find whether mobile app users post accessibility-related issues to app stores, Eler et al. [112] investigated 214,053 mobile app reviews using a string-matching approach. They depend on a set of 213 keywords derived from 54 BBC recommendations [62] proposed for mobile accessibility. In their work, they inspected 214,053 user reviews to identify reviews pertaining to accessibility. Their approach classified a total of 5,076 reviews as accessibility reviews. However, through a manual inspection later, the researchers found that only 2,663 of the reviews were really about accessibility. We used these 2,663 identified accessibility reviews as one of the two groups in our training set required for supervised machine learning. We created the second group (i.e., non-accessibility reviews) from their total dataset (i.e., 214,053). So far, this is one of the preliminary studies related to accessibility in mobile app user reviews.

3.7. Accessibility in Open-Source Applications

Many studies have conducted qualitative mobile-bug reports platform analysis [31] [69] [194] and Android-related bug reporting tool [215]. Markus et al. [197] propose a Braille interface platform named MOST with a wide range of applications. Al-Subaihin et al. [12] presented an assessment of mobile web application accessibility. McIlroy et al. [204] introduced an automatic labeling approach based on the types of user review issues. Liu et al. [186] conducted a study on Android applications to detect performance bugs to identify common patterns. Alshayban et al. [48] have analyzed 1,000 Android applications based on three perspectives developers, users, and applications for accessibility issues. Panichella et al. [240] proposed an approach using machine learning, which incorporated three NLP, sentiment, and text analysis techniques to introduce a taxonomy for classifying user reviews.

Vendome et al. [307] examined the Stack Overflow developer discussions of the Android app's accessibility. They have identified posts based on a list of keywords that have been chosen from the accessibility guide for mobile applications. They analyzed all the questions asked in the Stack Overflow and answers labeled Android and found 810 out of 1,442. In a study similar to ours, Eler et al. [113] performed an investigation on user reviews related to mobile accessibility. The study applied to user reviews of 701 applications from the Google Play Store. Their approach was to manually analyze the user reviews using a list of more than 200 keywords that refer to mobile accessibility.

3.8. Classification of Text Documents

Many studies classify app reviews using different taxonomies [91, 105, 148, 203, 241, 245, 24, 26, 37, 25, 173], for various purposes: detection of potential feature requests, bug reports, complaints, and praises, etc. Even though many of them identify reviews related to app usability, there is no explicit mention of accessibility-related issues [112].

Recently, many researchers have been focusing on feature extraction from text documents such as user reviews, product descriptions, and user stories, as well as classifying the features in one way or the other. For instance, Iacob and Harrison [148] mine app reviews to extract feature requests. They also use linguistic rules and Latent Dirichlet Allocation (LDA) to identify and group all common topics available in in-app reviews. Similarly, Galvis and Winbladh [83] analyzed automated topic modeling techniques on user comments for mobile applications. While Dumitru et al. [109] and Hariri et al. [136] extract features from *app descriptions* in order to make recommendations for feature implementation, they group the features using clustering algorithms. Similarly, Harman et al. [137] employ data mining techniques to extract app features from official app descriptions. They used collocation and greedy algorithms to extract app features and group them features.

Unlike automatic approaches, the classification of text documents using a set of *predefined keywords* has been vastly performed across different domains in software engineering. For instance, Eler et al. [112] relied on 213 keywords to identify accessibility-related reviews. Strogylos and Spinelles [287] identified refactoring-related commits using one keyword refactor. Similarly, Ratzinger et al. [253] used 13 keywords to detect refactoring in commit messages. Later, Murphy-Hill et al. [221] replicated Ratzinger's work in two open-source software using the 13 keywords Ratzinger used. However, they disproved the previous assumption that commits messages in the version history of programs are indicators of refactoring activities. The reasoning behind their findings is that developers do not always report refactoring activities as they might associate refactoring activities with other activities, such as adding a feature. AlOmar et al. [41] have also explored how developers document their refactoring activities in commit messages using a variety of 87 textual patterns (i.e., keywords and phrases). Similarly, we believe users can express accessibility concerns without explicitly using any accessibility keywords from the BBC guidelines, as assumed by Eler et al. [112].

In contrast to the keyword-based approaches, we used an automated machine learning approach since learning approaches outperform the accuracy of the keyword-based approach by at least 1.45 times [42, 195]. On the other hand, a keyword-based identification approach (i.e., relying on an existing set of predefined keywords) could generally miss certain reviews, not only because reviews left by users might not always use those keywords to express an accessibility concern but also because a single word might not be enough to convey an accessibility message. For example, the review I hope someday we change the size of the fonts; here the context provides an accessibility concern even though the user is not explicitly using keywords such as disabled, blind or low vision.

3.9. Chapter Summary

: This chapter examined several studies that shaped our methodology. Even though the chapter on the literature review has covered seven sections. The problems of online learning for deaf and hard-of-hearing students remain understudied.

In the next chapter, we examine the difficulties experienced by deaf and hearing-impaired pupils during the COVID-19 pandemic.

CHAPTER 4

IF ONLINE LEARNING WORKS FOR YOU, WHAT ABOUT DEAF STUDENTS? EMERGING CHALLENGES OF ONLINE LEARNING FOR DEAF AND HEARING-IMPAIRED STUDENTS DURING COVID-19: A LITERATURE REVIEW

4.1. Introduction

The COVID-19 pandemic has largely affected the education sector, particularly deaf education. According to Krishnan et al. [168], various measures to reduce the spread of the disease have led academic institutions to unprecedented changes to their academic activities. For instance, to comply with the social distancing requirement, most schools transitioned to online learning, while some have been forced to temporarily shut down if such technology was unavailable [193]. Although these measures have significantly reduced the spread of the virus, they have also introduced several challenges, severely impacting the educational systems worldwide [27].

Deaf education has been facing a unique set of challenges during COVID-19. To start with, distance learning platforms were quickly adopted mainly for non-disabled students since they represent the mainstream [196]. Despite their absolute right to access information, deaf students were initially left out of distance learning under the justification that them constituting a hard-to-manage population, requiring more specialized educational approaches [183]. In general, the social distancing measures have led to the exclusion and isolation of deaf students from instructors who could not promptly respond to their educational needs [168]. In addition, deaf students have experienced significant difficulties with information sharing. These issues include inadequate access to sign interpreters, loss of visual cues, auditory signal issues arising from the use of face masks, lack of transcripts or captions to lectures, etc. [268]. As noted by Swanwick et al. [290], the United Nations [226] made a

This entire chapter is reproduced from Aljedaani, Wajdi, Rrezarta Krasniqi, Sanaa Aljedaani, Mohamed Wiem Mkaouer, Stephanie Ludi, and Khaled Al-Raddah, "If online learning works for you, what about deaf students? Emerging challenges of online learning for deaf and hearing-impaired students during COVID-19: a literature review," Universal Access in the Information Society (2022): 1-20, https://link.springer.com/article/10.1007/s10209-022-00897-5, with permission from Springer Nature.

declaration titled "Disability-Inclusive Response to COVID-19", which acknowledged that people with disabilities took the hardest *hit* during the pandemic and their education requires immediate assistance.

While existing literature has focused on improving accessibility for disabled students in higher education, the pandemic has exposed critical weaknesses of e-learning systems for students with special needs that may need to be addressed. One way to strengthen virtual education is to identify challenges and barriers that appeared during the COVID-19 pandemic. One of the major concerns that students with disabilities had to cope with was adjusting to a completely new format of remote learning, and instructions [280, 236, **?**, 102]. With the strict regulations that all students had to comply with, students with disabilities, in general, and deaf students, in particular, were the most to suffer from them [193, 295, 208, 305]. The goal of this paper is to review and expose the major challenges that deaf students faced during the pandemic. We start with reviewing all research papers that were written as a response to these challenges. Then we analyze them to extract and categorize all the highlighted problems. Given that several studies have identified those challenges, our research aims to systematically collect and categorize them. This study reviews 34 papers to extract challenges and their corresponding key mitigation plans.

Reviewing the literature on the challenges facing deaf education during the current pandemic can provide solutions to e-learning beyond the pandemic. Previous studies have focused on general e-learning experiences, such as Mseleku [216], while others have looked at accessibility to online education by generally disabled students [261]. To the best of our knowledge, this is the first paper to review the literature related to accessibility challenges in the context of the COVID-19 pandemic.

The **contributions** of this chapter are:

- A literature review of 34, peer-reviewed deaf and hearing-impaired publications related to deaf students' education during the COVID-19 pandemic to provide a catalog for future research in this area;
- An exploration of the challenges faced by deaf and hearing-impaired students during

the COVID-19 pandemic;

- Key takeaways extracted from the reviewed studies for researchers and educators to improve the learning experience of deaf students;
- A replication package of our survey for extension purposes [3].

4.2. Research Questions

This chapter aims to explore the barriers to deaf and hard-of-hearing students in education during the COVID-19 pandemic. The study may help identify and critically expose the wide range of concerns and difficulties faced by deaf students during the pandemic. Furthermore, our literature review findings may serve as a comprehensive source for improving deaf education. Specifically, we investigate the following Research Questions (RQs):

RQ_1: What challenges and concerns are deaf and hard-of-hearing students in higher education facing with an online education during the COVID-19 pandemic?

RQ1 investigates a series of challenges and concerns during remote learning that deaf and hard-of-hearing students had to endure on the rise of the COVID-19 pandemic. We will explore more in-depth the findings related to recently published work in this domain and discuss implications since COVID-19 emerged as a global humanitarian problem.

RQ_2: What are emerging solutions to better handle challenges faced in deaf education during the COVID-19 pandemic?

RQ2 investigates the extent to which emerging in-demand solutions can be proposed to overcome some of the major barriers pinpointed in RQ1. At a larger schema, these solutions can serve as a mediating, non-perfunctory source of information to cope better with remote learning. It will shed light on alternating strategies and guidelines that could facilitate deaf and hearing-impaired remote learning and methods that could be implemented within institutions globally for more efficient remote learning.

4.3. Methodology

This present research is a Literature Review. It explores the existing, most up-to-date scholarly sources relating to the subject of the research to answer the research questions. The objective is to explore the key challenges of the deaf and hearing-impaired in education during the COVID-19 pandemic. This section is divided into the three phases followed when selecting relevant publications: planning, execution, and synthesis. Each of these steps is explained in the following sections.

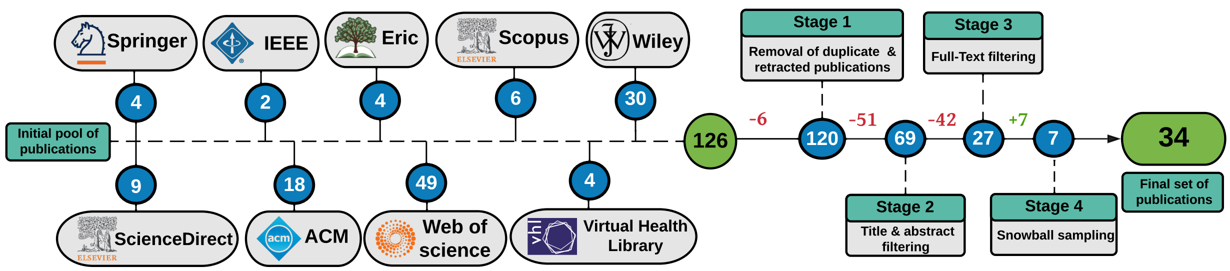

FIGURE 4.1. Overview of publications filtering process.

4.3.1. Planning

This step entailed refining our search strategy for literature. In line with the literature review methodology, we formulated a set of keywords related to our study, which we searched on various digital repositories.

Search Keywords We conducted a pilot search [97] to guide our formulation of search keywords in two repositories: ACM and IEEE. We wanted to identify the synonyms and words that are used when describing the barriers to deaf education during the COVID-19 period. Therefore, our search was restricted to the abstracts and titles only. Such a strategy helped in avoiding false positives. The search string used is as follows:

```
Title:("covid*" AND "deaf*" OR "hard of hear*" OR "hearing-impaired" OR
"hearing loss") AND Abstract:("educat*" OR "covid*" OR "e-learning" OR
"elearning" OR "Distance Learning" OR "online" OR "remote")
```

TABLE 4.1. Overview of targeted digital libraries used to collect published work.

Digital Library	Digital Library URL
ACM Digital Library	https://dl.acm.org/
IEEE Xplore	https://ieeexplore.ieee.org/
Science Direct	https://www.sciencedirect.com/
Scopus	https://www.scopus.com/
Springer Link	https://link.springer.com/
Web of Science	https://webofknowledge.com/
Wiley	https://onlinelibrary.wiley.com/
Virtual Health Library	https://pesquisa.bvsalud.org/
Eric	https://eric.ed.gov/

Digital Libraries A literature search was carried out in the following libraries: Scopus, IEEE Xplore, ACM Digital Library, Web of Science, Springer Link, Virtual Health Library, Wiley, ERIC, and Science Direct. We selected the nine libraries in order to ensure maximum coverage of the topic so that no important study was left out and utilized by similar studies (e.g., [32]). The various libraries queried are provided in Table 4.1. The libraries contained studies related to ours and in the fields of hearing-impaired education.

Inclusion/Exclusion Criteria. These criteria were useful in filtering and pruning our search results so that we were only left with those publications that were aligned with our study. For example, it was essential to ensure that we got studies in the education context and written in English while excluding those in the medical area and not peer-reviewed. We also included papers that were available in digital format and published during the COVID-19 period. The inclusion/ exclusion criteria are given in Table 4.2. Although we aimed at a final pool of relevant papers, the initial search results helped in manual filtering to evaluate the appropriateness of the studies for our research. For example, it was crucial to know the kind of obstacles they identified. Regarding the time frame, we restricted it to 2020, 2021, and 2022, which are the years that have been affected by the COVID-19 pandemic.

TABLE 4.2. Inclusion and exclusion criteria.

Inclusion Factors	**Exclusion Factors**
Papers are in education area	Websites, leaflets, and grey literature
Papers are written in English	Full-text not available online
Papers available in digital format	Published before 2020
Papers related to COVID-19 period	Papers related to medical area

Backward/Forward Snowballing. We undertook to snowball to add valuable articles to the ones we had obtained using automated search. According to Wohlin [317], snowballing involves reviewing papers that have emerged for a literature search and identifying articles that have cited the given paper (forward snowballing) or those that have been cited in the paper (backward snowballing). We conducted the snowballing in a closed recursive manner to make it more effective. As a result, we got a total of 10 articles from snowballing, from where we selected 7 that met our selection criteria. Finally, we included the articles from the snowballing activity to make our final count of 34 articles.

Exclusion During Data Extraction. Researchers can still eliminate some of the selected articles even at the data extraction stage. Such a situation occurs when the researcher discovers that a paper is a duplicate of another or meets the exclusion criteria. For example, we had an article that provided general information about communication obstacles during COVID-19 without focusing on deaf students [84], while another took a medical perspective instead of an educational one [122].

4.3.2. Execution

This section depicts the search results from the various digital libraries. The first search in all nine repositories gave a total of 126 articles. After that, we used four stages to evaluate the most relevant publications to our study. The first stage involved removing duplicate and retracted publications, where 6 articles were eliminated, and 120 publications proceeded to the next phase. The second stage was the title and abstract filtering, where we utilized our inclusion and exclusion criteria. In total, we removed 51 publications and allowed 69 to move to the next phase. For instance, the application of our inclusion and exclusion criteria led to the elimination of grey literature, non-peer-reviewed materials, and articles published before 2020, among others. The third stage was full-text filtering, which led to the removal of 42 articles and allowed 27 to move to the next phase. The final stage involved performing both forward and backward snowball sampling [317] that led to the addition of 7 articles. In total, 34 articles were selected for further analysis. Figure 4.1 shows the search execution process. Finally, we presented the titles of the 34 papers illustrated in the form of a word cloud as depicted in Figure 4.2.

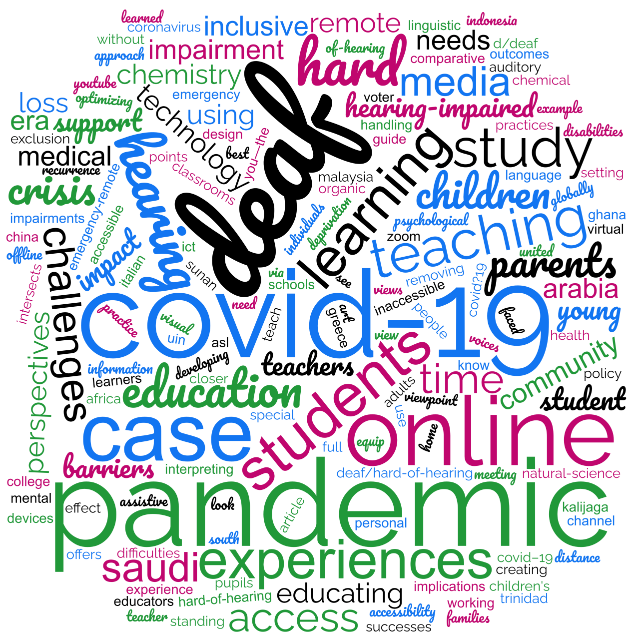

FIGURE 4.2. Word cloud of the titles of the selected papers.

4.3.3. Synthesis

During the synthesis phase, we examined the collected data with regard to how they could meet our research objectives. We classified the articles according to their country of origin and year of publication in order to understand where and when the barriers to deaf education were experienced. We ensured that every study was thoroughly scrutinized for concrete evidence and that all facts were provided. Careful examination was done to collect all data related to deaf education obstacles during COVID-19. For each study, we extracted the disability type such as deaf, hard-of-hearing, or hearing impairment, the year of publication, study methodology, source of information, approach of collecting data, participant type (i.e., teachers, students, leaders), study sample size, and study location.

To reduce bias in our data, we utilized a peer-review strategy, where all researchers reviewed the data, and any points of contention were discussed. The data was transferred to a Google Spreadsheet to ensure the collaboration of all the authors was in sync during the research. It is important to state that three of the authors were familiar with the scope of studies and have made similar publications and contributions in the past [29, 270, 28, 33, 40].

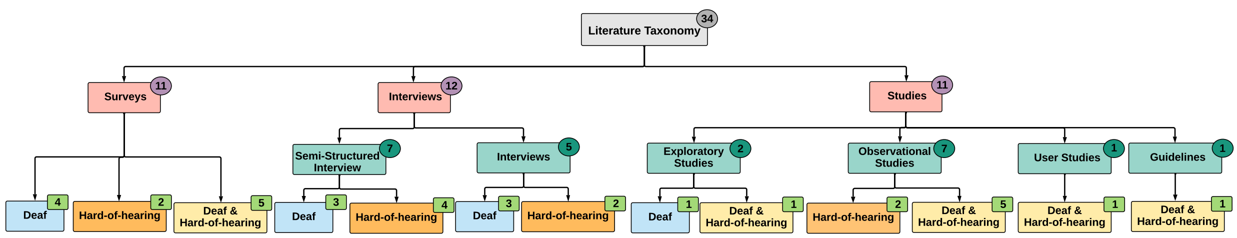

FIGURE 4.3. Overview of literature taxonomy of the selected research papers in our dataset. It highlights the methodology used and the targeted user group.

4.4. Results

This section reports the findings that we obtained by synthesizing various articles according to the scope of two research questions asked in this study. We analyzed a total of 34 articles. We report the characteristics of this set of studies extensively in Table 4.3 and 4.4. The data collected from this set of studies ranged from 2020 to 2022. The types of methods that these studies carried out are represented via a taxonomy as depicted in Figure 4.3. The figure provides a grouping of all the studies according to the methodology and methods used in the studies and the focus of the studies. From the figure, the most notable and common studies were those conducted in the form of surveys, interviews, and observational studies. The most common artifacts used to carry out those types of studies included social media, questionnaires, phone interviews, and other related documents such as guidelines. Regarding the focus of the studies, the most popular target groups in the surveys, interviews, and other studies were hard-of-hearing and deaf. Using the literature taxonomy, we were able to overview the studies we selected.

In Figure 4.4, we wanted to establish the country of origin from which the studies were done. According to the collected data, we notice that most of the studies were conducted in the USA. The second-highest number of studies originated from Indonesia, Saudi Arabia, the United Kingdom, Greece, Italy, and Malaysia, with all other countries having one study each. Such findings can help motivate scholars from countries with few or no studies to research deaf challenges in their locations. Figure 4.5 shows an overview of the types of the dataset used across all 20 studies. It is evident from the figure that social media was the most common and diverse data collection method that was used in the studies. It is possible to speculate that most researchers used social media platforms because of their popularity and because their use has not been affected by the social distancing measures implemented during the COVID-19 pandemic. The extensive use of technology during the pandemic, especially in education, also means that students with learning disabilities faced all kinds of challenges of different ranges from the unsuitability of technologies to health matters. That does not rule out the fact that similar challenges were not observed in other countries such as Indonesia or Italy and other countries. In fact, we noticed that the types of challenges that deaf students experienced were almost uniform across countries. It is important to place our findings within the context of the two research questions that we developed in this study.

RQ$_1$: What are the challenges and concerns that deaf and hard-of-hearing students in higher education are having with an online education during COVID-19 pandemic?

In this research question, we wanted to identify the issues that deaf students were facing during the pandemic. From our findings, we categorized the challenges into four categories: technological; educational; accessibility; and usage issues, and health-related.

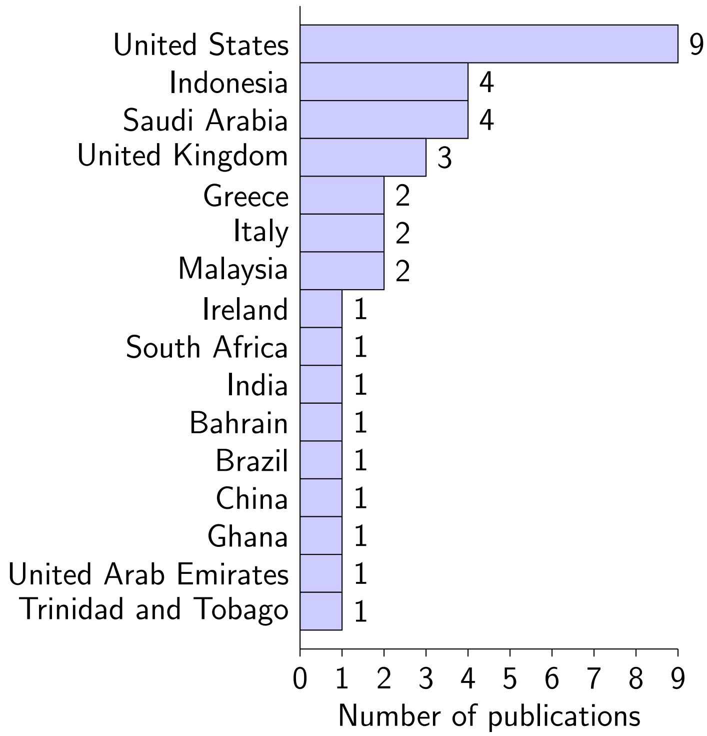

FIGURE 4.4. Distribution of publications across countries.

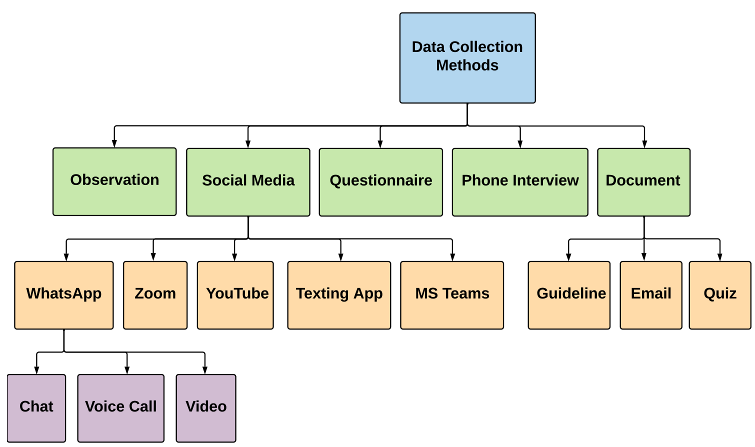

FIGURE 4.5. Overview of types of dataset used across 34 studies.

- **Technology related challenges.** Our main focus was to explore how technical issues affected deaf education during the COVID-19 pandemic. Deaf education faced four challenges: unavailability of hearing devices, disruptions during online lessons, and lack of familiarity with the online devices [168, 295, 54, 232, 134]. It is noted that the challenges in deaf education during COVID-19 can be grouped into three groups:

technological, organizational, and methodical [171, 46]. Technological challenges are those related to accessibility; organization depends on the collaboration between teachers, while methodical indicates how the instructions were taught. Mohammed [212] pointed out that video quality, internet stability, and language modality posed major technological barriers to distance learning for deaf students. Aljedaani et al. [21] presented the challenges faced by deaf students in distance learning and found that among 8 of the participants, 96.9% faced issues with internet connectivity at home, and 72% of the responses showed inaccessibility of the content.

Our interpretation of the aforementioned challenges is as follows. While students in general picked up quickly using new technology [98], this was not regarded as a doable option for students with disabilities [193]. On the contrary, they were faced with a series of challenges with the setup of the technology. First, the students with disabilities found it challenging to use the recommended technology. That was primarily because the interfaces of the software and applications were not designed to accommodate students with hearing disabilities [86, 233]. Second, students experienced enormous challenges adapting the use of video conferencing for synchronous lectures [46, 157]. Third, it was overwhelmingly difficult for students with hearing needs to follow conversations with multiple signers communicating simultaneously [295, 296, 288]. The lack of simultaneous translation was also one of the major obstacles to address [46]. Finally, delays in mainstream and remote classroom setup while interacting with deaf students or asking questions were a significant problem (since deaf students use sign language to ask questions), which the translator then interprets to the instructor [237, 291]. The findings of Alqraini and Alasim [44] highlighted deaf students' lack of focus during classes, as they choose to play games on their devices instead of paying attention to the ongoing lesson. We believe that such issues require solutions to facilitate deaf education.

- **Education system related challenges.** We also aimed to identify those challenges that were related to learning and the educational system. We found that, while most of the students got adjusted fairly quickly to the remote online system [55], this became a huge barrier, especially for deaf students [191, 44]. Even in a typical classroom setting, D/DHH students generally attend classes with the support of a special education team due to their special needs [201]. However, working from home, this new adjustment, in reality, created substantial barriers for deaf and hard-of-hearing students [196]. Researchers [134, 265, 47] underscored that lack of sign language interpreter hinders the understanding of deaf students with inadequate vocabulary knowledge. Even though the interpreters were present during the online classes, however, due to small visuals, it became challenging for the students to understand. Alsindi et al. [50] highlighted that in addition to miscommunication between teacher and student, the lack of interpreter's knowledge regarding art and design hindered the performance of students.

Deaf students have also suffered from a lack of access to education and welfare services, such as inadequate sign language interpreting avenues, the difficulty of lip-reading when teachers are wearing masks, limited direct support by teachers, among others [297, 285, 155, 232]. Unfortunately, the COVID-19 pandemic has worsened social exclusion among deaf students, especially with the disruption of daily interactions with other people, lack of access to information, and inadequate sign language interpreters [290, 46]. The exclusion is caused by lack of internet access, poor infrastructure, poverty that impedes lack of access to high-quality educational materials, barriers relating to lack of accessible learning management systems (LMSs), inability to use the LMS to access the content, and LMS systems that do not cater for the needs of deaf students [193].

Deaf education in some countries has been affected by a lack of resources in public schools, unpreparedness among teachers of deaf children, greater exclusion, and obstacles such as lack of real-time transcription services, technical issues, and unavailability of subtitles on videos [196, 232, 44]. The suggested problems call for improvement of the education system to make it more conducive for deaf education.

- **Physical accessibility challenges.** Adequate, accessible experience for students with hearing disabilities was an unattainable goal [152, 46, 134], even though distance learning equipment and technologies such as video-conference technologies, different websites, electronic platforms, applications, and/or various databases became available for most disabled students [236, 222]. This was not the case for underdeveloped countries [207]. Furthermore, some students with hearing disabilities lived in areas where there was hardly any access to the internet [305, 8]. To add another layer of barriers, some students with disabilities did not possess even basic technologies [285]. Hence, without physical attendance, remote learning for students with auditory access needs became a huge struggle for students with disabilities [152].

  We also established that, during the current pandemic, wearing masks seemed to have become the major impediment for students who were deaf or with hearing impairments [281]. Indeed, face masks became the worst enemy for hard of hearing and deaf students. Most importantly, a cloth face mask inhibited speech reading and blocked muffling sound [152]. They even prevented students from reading lips. The other concern pertained to both audibility and intelligibility of speech. Due to the wearing of masks, students with hearing issues found the teachers voices completely diminished through the use of masks and shields [205]. This made the student-instructor communication poor and inaccessible. Physical distance also became a significant obstacle between students and faculty only because this unconventional communication reduced speech audibility and intelligibility [199, 208].

- **Health related challenges.** The other most critical challenge pertains to the mental health of disability students [329]. Students with hearing disabilities showed four times more than other students increased symptoms of anxiety, depression, and emotional challenges compared with the general population [71]. We established that some health-related issues that emerged during and before the pandemic were affecting deaf education. For instance, deaf students have faced emotional challenges due to isolation from their classmates, and lack of access to important information during the pandemic [284, 321]. The fact that most deaf students experience impractical delays has also led to emotional and social issues among them [168, 237]. All these challenges have led to numerous mental health problems and unforeseen psychological impacts [108].

RQ$_2$: What are emerging solutions to better handle challenges faced in deaf education during the COVID-19 pandemic?

In this section, we will discuss the proposed solutions to the most prominent issues that have been identified in the previous section. The technology used in deaf education must ensure that the audio is clear and with self-explanatory images, the activities taught should be easy due to the online learning challenges, and there should be concerted efforts from all stakeholders [171]. Furthermore, deaf students should be given mental health services, training on pragmatic skills, be provided with hearing aids, be encouraged to read, and also be facilitated to gain information during the pandemic [284, 321, 168, 237]. It is also recommended that parents look for suitable online educational programs, find opportunities for exposure to deaf students, communicate with deaf and hard-of-hearing students, enable deaf and hard-of-hearing students to socialize, and assist them in getting the services they need [169]. A combination of government-led and community-led responses has also provided greater educational and social support for deaf students [297]. It is also proposed that recognition of group rights will lead to greater inclusion for deaf students, so that cultural and linguistic accessibility can be offered to the population [290]. For example, sign language should be considered and recognized as a language like any other.

It is also important to develop videos with captions and interpretations, whether offline, online through YouTube or cloud-based Zoom recordings, which are especially useful to the deaf community [119, 280, 222, 289, 191, 232, 44]. An important aspect of the videos is that they must be thoroughly tested for validity to ensure their effectiveness, and revisions are done in order to improve the quality of the videos. Sutton [289] also notes the importance of providing interpreters and speech-to-text capabilities for deaf students during the pandemic to aid their learning. It is suggested that governments should utilize inclusive educational models, improve the accessibility of deaf students to various services, provide deaf-friendly masks, expand television programming and hire more teachers in order to have a favorable number of staff assisting deaf students [196, 285]. Low-income families should be given financial assistance to purchase electronic equipment for their children as recommended by Alqraini and Alasim [44]. The authors also proposed that a quiet environment should be created for the students during their lessons. [212] suggested the provision of standard educational technologies to teachers and students, proper training of teachers [8, 47, 54], hosting workshops concerning deaf culture, and video translation of textbooks in sign language to ensure the effectiveness of distance learning.

Karampidis et al. [154] recommended that distance learning platforms should be integrated with "Hercules", a bi-directional translator that translates five languages, including Greek, Cypriot, British, German, Slovenian, and Portuguese, to their respective sign languages and vice versa. Institutions must incorporate a better approach to provide accessible technology that individuals with diverse needs can adopt during the pandemic [21]. Another study [248] suggested the use of ICT (Information Communication Technology) to conduct online classes during the pandemic. The uninterpreted-learning ICT models were preferred by the participants of the case study [8] over Zoom classes. Mathews et al. [200] stated that to address the communication gap in distance learning, interpreters have had to employ a variety of specialized expertise, interact with one another, and actively involve both their hearing clients and Deaf communities in diverse settings. The study also recommended vocabulary development of the interpreters to convey the lessons more conveniently. Alshawabkeh et al. [47] suggested that deaf students must be trained by an IT professional with a sign language interpreter prior to the initiation of distance learning. Students, teachers, and interpreters should collaborate in order to present material simultaneously. They also proposed that teachers involve deaf students in planning the online class before it begins. Institutions should continuously evaluate deaf student's feedback to enhance the quality of distance learning. Moreover, the existing LMSs must be provided with additional features for the DHH students [50].

Our study has also shown that governments should also put in place inclusive emergency plans and improve access to telecommunication services such as the internet to deaf students [193]. It is also proposed that policy changes should be made to enable deaf adults to participate in early intervention teams and greater collaboration from multi-agency teams in order to have professional teams working towards inclusion and education of deaf children in the pandemic [236]. Deaf students should have a conducive environment at home, support from parents, online instructional content, access to specialist support, and good access to instructions to mitigate the connectivity challenges [152, 295]. It is evident that collaborations from a wide range of stakeholders will provide the necessary support and resources needed for improving deaf education.

4.5. Discussion

Our literature review provides an elaborate overview of the challenges that deaf students have been facing in education during the course of the current pandemic. Furthermore, we also reviewed potential solutions that can be enforced and incorporated by different authors. In this section, we provide notable takeaways from our study.

Takeaway 1: Provide necessary equipment and technology. We have established that a lack of equipment such as hearing aids and inaccessibility to the internet are major obstacles impeding deaf education during the COVID-19 pandemic. The problem is worse in rural areas and those with high levels of poverty [193]. As further indicated by Paatsch and Toe [237], global research has shown that many deaf students attend mainstream classes that do not have adequate support for the difficulties that such students face. It has also been demonstrated that deaf students face challenges when using Zoom platforms, especially given that the platform has a steep learning curve and its features are not easily understood by all students [295]. One of the technologies lacking for many deaf students is Remote microphone (RM) hearing assistive technology (HAT), which should be customized to the needs of every student [152].

TABLE 4.3. Detailed information regarding the 34 papers selected: These publications report major challenges that students with special needs, specifically deaf and hard-of-hearing students faced in academic institutions during the COVID-19 pandemic.

Studies	Purpose	Year	Category	Method	Source of Info	Method of collecting Data	Participants	Sample Size	Study Location
[193]	Provides alternative educational methods aimed for Deaf students	2021	Deaf	Semi-Structured Interview	Ministry of Education	WhatsApp, Audio Records, Phone Call Interviews	Teachers	18	Saudi Arabia
[8]	Exploring the emergency-remote teaching of natural sciences to deaf learners	2022	Deaf & Hard-of-Hearing	Semi-Structured Interview	Four schools in the of KwaZulu-Natal	Zoom	Teachers	7	South Africa
[21]	Investigating the e-learning experiences of deaf students	2021	Deaf & Hard-of-Hearing	Survey, Interview	Technical & Vocational Training Corporation(TVTC)	Zoom	Students	65	Saudi Arabia
[44]	Exploring the challenges and support methods for D/DHH students during their distance education	2021	Deaf & Hard-of-Hearing	Semi-Structured Interview	20 Elementary Schools	Phone Call Interviews	Parents	37	Saudi Arabia
[46]	Challanges of teaching deaf students	2020	Hard-of-Hearing	Interview	King Saud University (KSU)	Unstructured Phone Interviews	Lectures	11	Saudi Arabia
[47]	Investigating the technological instruction provided to deaf students in online learning	2021	Deaf	Semi-Structured Interview	Al Ain University	MS Teams	Students, Teachers	15 Students 3 Teachers	United Arab Emirates
[50]	Investigating the challenges of virtual learning faced by art and design D/DHH students	2021	Hearing Impairments	Semi-Structured Interview, Observing	University of Bahrain	MS Teams, WhatsApp	Students	105 Males 5 Females	Bahrain
[54]	Challenges faced by the teachers while teaching students with hearing impairment during a pandemic	2021	Hard-of-Hearing	Survey, Interview	Special and Inclusive Schools of Punjab	Social Media, Google Forms	Teachers	87	India
[152]	Challenges and unexpected benefits with remote learning	2020	Deaf & Hard-of-Hearing	Observational Study	United States	UNK	Students	UNK	United States
[103]	Addressing the content of Organic Chemistry in a contextualized for D/DHH Students	2022	Deaf & Hard-of-Hearing	Observational Study	Federal Institute of Paraiba	Quiz	Students	1 Deaf 8 Hard-of-Hearing	Brazil
[119]	Focuses on the development of voter education Videos	2020	Deaf	Interview	GERKATIN	WhatsApp	Students	33	Indonesian
[289]	Accommodation Strategies for Deaf Student	2020	Deaf & Hard-of-Hearing	Guideline	Centers for Disease Control & Prevention	Guideline Documents	Students	UNK	United States
[134]	Exploring the experiences and barriers by students with disabilities in online learning	2021	Deaf & Hard-of-Hearing	Survey, Interview	UIN Sunan Kalijaga	Phone Call Interviews, WhatsApp	Students	34 Total 15 Deaf Students	Indonesia
[154]	Challenges and barriers in learning environment of deaf students	2021	Deaf	Observational Study	Greece	UNK	Students	UNK	Greece
[168]	Challenges affect the communication and mainstreaming process	2020	Hearing Impairment	Interview	MySkill Foundation	WhatsApp	Students	3 Males 7 Felmales	Malaysia

Table 4.4. Continued detailed information regarding the 34 papers selected.

Studies	Purpose	Year	Category	Method	Source of Info	Method of collecting Data	Participants	Sample Size	Study Location
[169]	Recommending parents about educating DHH children	2020	Deaf & Hard-of-Hearing	Survey	United States	UNK	Parents	133	United States
[284]	Barrier to access an appropriate information	2020	Deaf & Hard-of-Hearing	Survey	Lee University	Social Media & Email	Students	19 Deaf 17 Hard-of-Hearing	United States
[171]	Presenting the remote teaching experience	2020	Deaf	Survey	Scuola Audiofonetica	UNK	Students	233	Italy
[191]	Challenges and success in Teaching Chemistry for deaf students	2020	Deaf & Hard-of-Hearing	Observational Study	NTID	Social Media (Chat, Zoom)	Students	UNK	United States
[196]	Difficulties and barriers deaf and hard of hearing individuals	2020	Deaf & Hard-of-Hearing	Survey	Greek Ministry of Education	Social Media	UNK	UNK	Greece
[200]	Examining the experiences of sign language interpreters during the COVID-19	2022	Deaf	Semi-Structured Interview	Council of Irish Sign Language Interpreters	UNK	Interpreters	16	Ireland & United Kingdom
[212]	Investigating how an emergent system of e-learning affects the linguistic access of deaf students	2021	Deaf & Hard-of-Hearing	Semi-Structured Interview	Two Deaf Primary Schools	UNK	Student, Teachers Interpreters Parents	4 Student, 4 Teachers 2 Interpreters, 3 Parents	Trinidad and Tobago
[119]	Develops a Tech Media for Students with Hearing Impairments	2020	Hearing Impairment	Exploratory Study	GEKARTIN (Videos)	Social Media	Students	8	Indonesia
[155]	Implications of deaf students in medicine	2020	Hearing Impairment	Observational Study	United Kingdom	UNK	Students	UNK	United Kingdom
[236]	Provides sources of prevention of deafness to support services for deaf children	2021	Deaf	Survey	United States	Text Apps, Signed Languages	UNK	UNK	United States
[232]	Exploring the effectiveness of the assistive istening device system in online learning contexts	2022	Deaf	Semi-Structured Interview	Scottish Sensory Centre	Zoom or Teams	Students, Parents Leaders	3 Students, 13 Parents 3 Leaders	United Kingdom
[248]	Exploring the online learning process using computer information technology media	2021	Deaf	Survey	Vocational Schools	UNK	Teachers	50	Indonesia & Malaysia
[265]	Investigating the accessibility of Deaf Students During	2021	Deaf	Interview	Brawijaya University Malang, Dinamika University Surabaya, Widya Mandala Catholic University	WhatsApp Video Call	Students	4	Indonesia
[280]	YouTube Instructional Videos accessible to Deaf of Hearing (DHH students)	2020	Deaf & Hard-of-Hearing	User Study	TWUFCL	Social Media YouTube Channel	K-12 Students	4	United States
[285]	Adoption of video captions understandable for deaf viewers	2020	Deaf	Survey	United States	UNK	Children	UNK	United States
[290]	Impact on deaf adults, children, and their families in Ghana, focusing on issues of inclusion	2020	Deaf	Interview	Ghana	WhatsApp (Chat, Call, Video)	Teachers, Leaders, Students	5 Males 1 Female	Ghana
[295]	Challenges and improvements to ASL online teaching	2020	Deaf & Hard-of-Hearing	Observational Study	RIT	Social Media (Zoom)	Students	10	United States
[297]	Exploring the impacts of the recent pandemic crisis	2021	Deaf	Exploratory Study	Literature Review	UNK	UNK	UNK	Italy
[321]	Observations on mental health for students with hearing loss	2021	Hard-of-Hearing	Survey	Higher Education Institutions & Special Education Schools	Questionnaires	Students	1100	China

Takeaway 2: Improve accessibility and usage of learning materials. We have noted that many institutions have digitized their content, however, it is still inaccessible due to a lack of captioning and unclear audio, among other issues. Such a finding is consistent with Fernandes et al. [119], who found that learning materials for deaf students should meet the validity and effectiveness so that they can be of help to deaf students. However, it is not translated even when such content is accessed, and there are no speech-to-text services. Furthermore, deaf students find it hard to follow the teacher during virtual classes, when several faces are appearing on the screen simultaneously, or when captions' speed is fast [169]. The lack of self-explanatory images, the presence of background music, and the inclusion of unnecessary decorative details also make the accessibility of learning materials difficult [171]. It is important to provide visual materials and techniques that will help deaf students learn more effectively [46]. Another accessibility challenge during the COVID-19 pandemic is that the use of face masks by teachers on online platforms makes it hard for deaf students to read lips, which is a major challenge in their learning that should be overcome by using clear masks [289]. The provision of accessible learning materials will be very important in improving deaf education.

Takeaway 3: Improve collaboration and partnership. It has been clear that all stakeholders should be involved in improving deaf education. The proposed solutions indicate the important role played by the government, teachers, parents, and specialists in improving education outcomes for deaf students. Using the example of Saudi Arabia [193], governments can play a crucial role to help in creating a conducive environment for deaf education. Furthermore, in Italy, Tomasuolo et al. [297] explains the crucial role of stakeholder lobbying by deaf organizations such as the World Federation of the deaf (WFD), and the Italian National Deaf Association, among others. It is noted that collaboration between deaf community members, deaf organizations, scholars, and activists in many countries around the world has led to greater access to education, improved use of captions, greater use of Text apps, broadcasting of content that considers the deaf community, utilization of clear masks, among others [285, 236]. Therefore, such collaborations and partnerships provide important opportunities for improving the quality of deaf education during the current pandemic.

Takeaway 4: Cater for the mental health needs of deaf and impaired students. We have found that some students developed mental health issues during the pandemic, while others already had them prior. As explained by Krishnan et al. [168], such a situation has been brought about by the social distancing and related protocols during the COVID-19 pandemic, which has added to their isolation and lack of social interactions. Swanwick et al. [290] indicate that deaf students faced social exclusion even before the pandemic, but the current situation has exposed and deepened the issue. The pandemic has also led to negative emotional responses from deaf students because the pandemic has led to the school closing, fear of illness, and social distancing, among other family problems [284]. It has been noted that deaf students are psychologically resistant to the effects of the pandemic but show less mental resilience compared to normal hearing students [321].

Takeaway 5: Simplify the LMS systems. Our study has shown that the mere availability of the LMS systems does not guarantee quality online education for deaf students. Indeed, the switch to online learning has been abrupt due to COVID-19, and most deaf students faced tremendous challenges in accessing the content on LMS platforms [193]. It has also been observed that there were predominant challenges in ensuring an uninterrupted-learning environment via video conferencing, for example, whether Zoom could adequately display LMS-located content or not [191]. Such systems need to be simplified and customized to improve their usability features and look and feel for deaf students. LMS systems are extremely important for remote access to materials and learning for deaf students.

4.6. Conclusion

In this chapter, we conducted a comprehensive literature review with the aim to investigate the chief challenges that education has faced recently by deaf and hard-of-hearing students during the COVID-19 pandemic. In summary, our research contributions provide substantial evidence of the immediate need to investigate the barriers that we emphasized in the previous section. Furthermore, this early contribution of the present work opens an opportunity for the research community and the educational sector to address these needs broadly and globally with similar interest and care. Additionally, our work directly contributes to the literature by providing a detailed analysis of online learning challenges for deaf and hard-of-hearing students. Most critically, it brings forward attention to recommending educational systems to be more accessible during pandemic crises and leverage teaching strategies that can be easily incorporated even in the face of environmental crisis. In addition, we have also disseminated our data as a supplementary electronic file for the research community to engage more extensively in a similar line of research and replicate our work for further advancement of SLR research.

4.7. Chapter Summary

: This chapter analyzed the challenges deaf and hearing-impaired students had during the COVID-19 pandemic. In addition, we provide critical findings from the reviewed study. As a result, our comprehensive literature study revealed that online education for deaf and hard-of-hearing students was not universally available. It revealed a lack of comprehension in the middle-eastern nation regarding the obstacles experienced by deaf students.

In the next chapter, we investigate the e-learning experiences of deaf students during the COVID-19 era, with a particular focus on the Technical and Vocational Training Corporation (TVTC) in the Kingdom of Saudi Arabia (KSA).

CHAPTER 5

I CANNOT SEE YOUTHE PERSPECTIVES OF DEAF STUDENTS TO ONLINE LEARNING DURING COVID-19 PANDEMIC: SAUDI ARABIA CASE STUDY

5.1. Introduction

The COVID-19 pandemic has necessitated the introduction of various public health measures to control its spread, including social distance measures. Such policies have affected nearly every sector of the economy, including education. Unfortunately, Gleason et al. [128] indicated that People With Disabilities (PWD) are often disproportionately affected in times of drastic and unintended changes. In the case of COVID-19, PWD is facing challenges in education because social distance measures have forced education institutions to shift from face-to-face learning to e-learning. As noted by Hanjarwati and Suprihatiningrum [134], some of the challenges faced include a lack of support, expensive internet access, and the inability to work with the e-learning system, among others. It is important to raise awareness of how inclusivity in education can be achieved during the COVID-19 era, such as by promoting the use of blended learning, providing sign language options, and improving support for disabled persons [301]. It is also important to resolve barriers to education for disabled students, which include technical problems, time, and absence of simultaneous translation, among others [46, 21].

In the last few years, studies on challenges associated with e-learning depended on the evolution as well the development of the e-learning system [312, 13]. There are usually three types of interactions in e-learning systems, i.e., teacher-to-learner, learner-to-course contents, and learner-to-learner interaction [283]. Several studies have been conducted in Saudi Arabia to analyze the impact of COVID-19 on multiple factors, such as financial, psychological, political, and societal attitudes [324, 252, 19, 110]. Furthermore, a number

---

This entire chapter is reproduced from Aljedaani, Wajdi, Mona Aljedaani, Eman Abdullah AlOmar, Mohamed Wiem Mkaouer, Stephanie Ludi, and Yousef Bani Khalaf, "I cannot see you—the perspectives of deaf students to online learning during COVID-19 pandemic: Saudi Arabia case study," Education Sciences 11, no. 11 (2021): 712, https://www.mdpi.com/2227-7102/11/11/712.

of studies investigated the use of online learning within Saudi Arabia during the COVID-19 period. For instance, research by Almekhlafy [38] focused on online learning of English courses using blackboard, and [118] examined the student satisfaction with the teaching quality of case-based discussion (CBD) sessions. Another study by Alshehri et al. [49] investigated the online learning facilitated syllabus delivery and assessments during COVID-19 and found that it was important to improve IT infrastructure, teacher training on online education, and student engagement, whereas [45] found that it was challenging to teach complex scientific concepts through online means, and there was low interaction between students.

However, studies performed in Saudi Arabia on disability, specifically with deaf students, are limited. There are only two studies that investigated deaf education during the pandemic. Madhesh [193] investigated deaf students' situations through 20 ministries of education channels that were utilized during the locked-down period. The goal of the study is to provide an alternative educational method aimed at Deaf students. The second study was conducted by Alsadoon and Turkestani [46], where they investigated the obstacles that the instructors faced while they were teaching online classes. Both studies were conducted on teachers of deaf students, but they did not examine the deaf students' challenges and concerns during the sudden shift to online learning. Our study is the first to focus on the challenges deaf students have faced when transitioning to online learning during the pandemic. More specifically, this study is unique since, compared to other countries, online learning is not very established in Saudi Arabia, and its implementation has mostly been heightened by COVID-19. Saudi society is also traditional and conservative, wherein the deaf culture is still new and not well-established [39]. Thus, such students may have low self-esteem in communication [10], which may affect how they learn using the e-learning platforms. Furthermore, the context of this research is unique in terms of its gender focus, given that the Technical and Vocational Training Corporation (TVTC) only admits male students, unlike the participants in other studies that were both male and female. Therefore, the context of this study is very unique and its findings will be a great contribution to the body of research on the subject.

The aim of the current study is to explore the e-learning experiences of deaf students during the COVID-19 period, focusing on the Technical and Vocational Training Corporation (TVTC) in the Kingdom of Saudi Arabia (KSA). To explore the e-Learning experiences of deaf students during the COVID-19 era at TVTC, a mixed-methods approach was used. First, we conducted an interview to collect preliminary insights. The interviews were performed with eight deaf students who voluntarily involved in the study. Then, we perform a survey in order to obtain the views of deaf students whose education had been disrupted by the pandemic. The survey helped in discovering new insights and estimating the prevalence of some aspects using a larger population, as well as providing explanations for support or opposition to some questions. Since the education of deaf students via online learning has not been previously investigated, this study will shed light on issues and challenges that can occur for deaf students while learning online. In this study, we investigate the following research question:

RQ_1: What are the challenges and concerns that deaf and hard-of-hearing students are having with an online education during COVID-19 pandemic?

This RQ will guide this research by investigating the difficulties, challenges, and concerns of deaf students during the pandemic period. We will answer this question by exploring the students' perspectives of TVTC college through interviews and surveys investigating the learning processes during COVID-19.

The contributions of this chapter are:

- To explore the challenges faced by deaf students during the pandemic.
- To identify how issues faced by deaf students during the COVID-19 pandemic can be solved.

5.2. Materials and Methods

This section presents the approach of our study, information about the participants engaged in the study, the data collection process, details about the procedures that were followed in interviews and surveys, and analyzes the data to address our research question.

5.2.1. Study Approach

This research was carried out in several stages as follows. Firstly, the survey and interview guides were created based on the research questions and a preliminary review of the literature on the subject. Secondly, the survey was administered, and interviews were conducted by the researcher. Thirdly, the survey was analyzed, and transcripts were coded. Finally, thematic analysis was used to create a theme map, which was followed by the analysis of results.

To explore the e-Learning experiences of deaf students during the COVID-19 era at TVTC, a mixed-methods approach was taken. The specific methodologies used were case study and survey [64]. We considered two approaches to be appropriate for this study because of several reasons. The first reason is that the case study methodology allows us to investigate a particular phenomenon in its natural environment [328], which also applies to deaf students in the TVTC. Given that the COVID-19 situation is of a worldwide nature, we deemed it fit to use a survey in order to obtain the views of deaf students whose education had been disrupted by the pandemic. The survey technique was conducted using two data collection methods, namely surveys, and interviews.

The nature of the current study is that it is both descriptive and exploratory qualitative. The descriptive aspect provided observations on how deaf students are e-learning in the current pandemic, while the exploratory qualitative aspect sought to identify their experiences in the COVID-19 era [130, 259]. In this study, we followed the case study guidelines by Runeson and Höst [259] and survey guidelines by Kitchenham and Pfleeger [162]. To analyze qualitative data, we combined our methods with a deductive thematic analysis [74, 258, 75]. The thematic analysis was selected for this research [75]. The reason for selecting thematic analysis was to enable the researcher to identify themes in the study that could help interpret interviews and derive meanings. Various prior studies conducted on deaf and hard of hearing have found this method to be adequate [93, 66].

One of the advantages of thematic analysis is flexibility, and it was selected in this study because it can follow a given theoretical framework, unlike grounded theory [228]. The theoretical framework [75] employed in this research is deaf and hard of hearing challenging (described in detail in Section 10.3) stages, meaning that the thematic analysis approach used will be deductive. The researchers assumed a connection between the respondents' replies and the meanings. Hence, the essentialist/realist thematic analysis approach was adopted [304]. The directions given by Braun and Clarke [75] guided the thematic analysis technique in this paper.

5.2.2. Data Collection

This study's data were collected in two steps. First, we conducted interviews to collect preliminary insights, similar to the empirical approach of collecting evidence through surveys. The interview responses' patterns gave crucial insights into e-learning experiences for deaf students. Secondly, from the findings of the interviews, we designed a survey and distributed it to deaf students. The reason for using the survey was to corroborate the data from the interviews with a higher sample size. We conducted a survey of deaf students in the TVTC. By conducting the survey, an in-depth investigation of the research question could be explored comprehensively and systematically.

5.2.3. Interviews

The researcher conducted interviews to explore the general experiences of deaf students learning at the TVTC during the pandemic. The following sections provide the interview protocol, participants, and analysis of the interview data.

Protocol

To ensure that researchers received both structured and unstructured responses, a semi-structured format was used in creating the interview schedules. The interviewer used the funnel method so that the interviews would look like conversations [76], as opposed to a question-and-answer format. Such an approach encouraged the interviewees to speak their minds freely, although the researcher ensured that the topics of interest in the discussion were addressed. The mentioned approach allowed the researcher to meet the exploratory and observational objectives of the study. To ensure the validity of the interviews, investigator triangulation was conducted [146], where the questions were thoroughly discussed, and interviews were conducted by three researchers. It was generally agreed that the questions were sufficient in collecting information about the experiences of deaf students during the COVID-19 pandemic.

The interview consisted of 25 questions that asked various aspects that were in line with the objectives of this study. Given that a semi-structured approach was used, the interview questions acted as a guide for the researcher. The questions were used as conversation starters, after which the conversations flowed without disruptions. Table 6.2 presents the set of interview questions. Interviews were conducted by the researcher via the Zoom platform using the Arabic language. All the respondents were native Arabic speakers; hence, the choice of the interviewing language was made. Given that Arabic is the first language in Saudi Arabia, it enabled the researcher to easily interact with the interviewees and obtain more insight from them. It is crucial to mention that the students were speaking sign language, and an interpreter translated the signed language into the Arabic spoken language. We considered this accommodation vital for the smooth running of the interviews, and since none of the researchers was conversant with sign language, we hired an interpreter.

TABLE 5.1. Present the participants Demographics information. Each participant (P#) answered the interview questions.

Participant	Age	Major	Year	Derive Used	Received Support Yes/No	Prefer Learning Online Yes/No
P1	23	Computer Technology	3	Laptop	Yes	No
P2	22	Computer Technology	2	Laptop	No	No
P3	22	Business	2	Laptop	No	No
P4	21	Business	4	Mobile Phone	No	No
P5	24	Business	1	Mobile Phone	Yes	No
P6	20	Computer Technology	3	Desktop Computer	No	Yes
P7	22	Business	3	Mobile Phone	No	Yes
P8	23	Computer Technology	4	Desktop Computer	No	No

Participants The voluntary response sampling method explained by Murairwa [220] was used because the researcher wanted to include only those deaf students that were willing to share their experiences. Therefore, out of all the participants that were willing to take part in the research, only those who volunteered were interviewed. The interview stage was exploratory, and therefore, the researcher was not concerned about non-generalizability of results because of using the voluntary response sampling method.

The number of interviewees that agreed to take part were 8, out of a population of 80 deaf students that had been contacted. The interviewed students were all male because the college admits male students only. The individuals were contacted via the students' emails, and their responses towards the participation request were noted down. All of the 8 students were male, 4 in each of the 2 majors at TVTC (computer technology and business). Table 6.1 presents the demographic summary of the participants. The equal splitting between the two majors was conducted in order to have a good overview of each category. The interview duration was between 20 and 30 min. We compensated all participants with a $25 prepaid gift card.

TABLE 5.2. Presents the set of interviews questions.

First **Background and Demographics**	***Fourth*** **Challenges and Concerns**
Years of age, and study major	What was your distractions while you learning online?
Do you have access to a device for learning online?	What was your most challenges during online learning?
What device did you use for online learning?	How was your learning environment at home?
Second **Generic Views**	How did you communicate with your teachers?
How would you describe your experience in learning online?	Were you able to access the class materials via Blackboard?
What type of device did you use for online learning?	How you ever encounter any barrier or issue communicating your teachers or department?
How did you study the class that are needed hardware equipment?	Did all videos support the subtitles?
Based on your experience in online learning, what do you prefer now?	Do you have a printed text transcript of audio content on the website?
Third **E-learning Tool**	***Fifth*** **Students recommendations**
What is your perspective on Blackboard platforms?	What do you feel are the benefits of online courses, such as those provided during Covid?
Did you used any other e-learning tools? why?	What are the things that you would like to change in online learning?
Did you train on Blackboard? if not, did teachers and department shared with you resources?	

Data Analysis The interviews were transcribed, which prepared them for the data analysis stage that was conducted through thematic analysis. The first stage of the thematic analysis method is reading the scripts in detail to facilitate coding the interviews in line with the research questions. In this study, codes were used to categorize the responses of the participants according to the selected topics. Subsequently, the codes were utilized in creating a theme map, which would illustrate the results of the study.

Transcript Coding

The interview responses given in the interviews were scanned in order to facilitate the coding process. The researcher assigned codes according to topics that expressed certain opinions, attitudes, and experiences that related to the research question. Given that the interviews were long, the researcher identified only the relevant responses to the current study. In the initial step, the researcher scrutinized the interview scripts and created a list of codes that emerged from the responses. The second step involved evaluating and investigating the codes to ensure that they were representative of the research questions. In the third step, the researcher revised the codes, which involved merging or dividing some of them.

Deducing Themes

The researcher categorized the generated codes into various themes. In this research, a theme is considered a pattern of responses that relate to a given research question. Thematic analysis involves constant revision of themes as the researcher investigates the interview text, which ultimately leads to the creation of a theme map [75]. In this study, a theme map demonstrated the insights derived from the interviews and their relationships. Such an approach ensured that detailed and in-depth descriptions of the research subject was conducted without interference from irrelevant data. Theme mapping was conducted and revised three times by the researcher. An illustration of the research findings of the theme map is given in Figure 5.1.

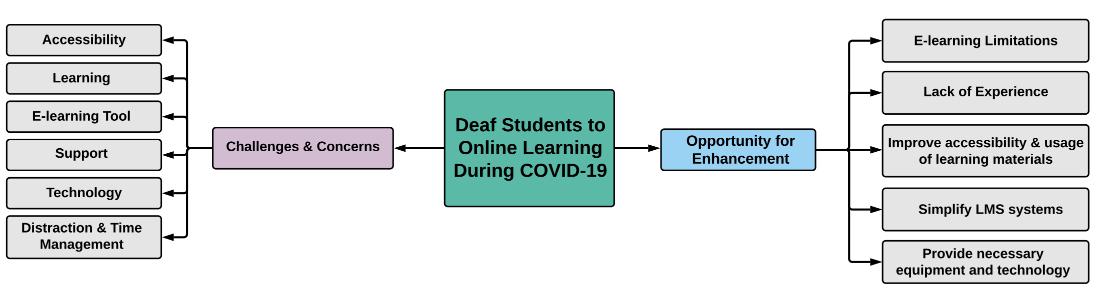

FIGURE 5.1. Thematic analysis findings in the form of a theme map.

5.2.4. Survey

After identifying the relevant topics from the interviews and considering the research questions, the survey was designed. The survey was created to corroborate the interview findings, discover new insights, and estimate the prevalence of some aspects using a larger population.

Design The designing of the survey began with generating 100 questions and statements that were developed by the author. The survey was divided into four sections according to the themes that had been generated in the interviews section. Time was taken to revise the questions in order to remove those that were considered ambiguous, irrelevant, repeated, or of a personal nature. The revision of the questions paved the way for reducing the questions from 100 to 72. To evaluate the survey's effectiveness and overall experience, a pilot study was conducted with five deaf students. The feedback was that the survey was very long, some of the questions were repeated, and that it was important to arrange the questions in a logical manner.

Based on that information, we reduced the number of questions from 72 to 42 and grouped them into 3 sections. Easier questions were given first to encourage the respondents to answer the questions. Demographic questions were also placed at the end of the paper so that they do not lead to no-response if placed at the beginning. The researcher included both closed-ended and open-ended questions in order to provide a chance for the respondents to give their personal insights without restrictions. For our final survey, we included 42 questions that can be broken down as follows: 18 Likert questions, 19 multi-choice questions, and 5 that were open-ended. The Likert questions used a 5-point scale that indicated the extent of interest, concurrence, or importance of an aspect. It was also deemed appropriate to make some of the questions optional so that the respondents do not feel pressure to answer them just for the sake of completing the survey. The survey were designed using Google Forms, which also helped in collecting the data. It is also important to state that the survey was created in the Arabic language, which is the first language in Saudi Arabia, to enable the respondents to understand the questions easily [224]. The survey and interviews were carefully translated into English by three authors. Subsequently, a three-stage process was conducted to check the translated responses for their correctness. In the three-stage process, each of the three authors reviewed the translations and certified that they were accurate. Our survey questions are available in both Arabic and English in [2].

Participants The survey link was shared through the students' emails. Out of the initial targeted number of 80 respondents, the researcher received 65 responses (response rate = 81.25%), which is considered high [282]. All the respondents were male because the college has only male students. From the total number of participants, 26.2% majored in computer technology, while 73.8% majored in business. The respondents were in their first, second, and third years of study.

Data Analysis The results of the survey were analyzed by first merging some of the responses. For instance, strongly agree and agree were combined to give the general agreement rate. A weighted average response was developed to simplify the recording and analysis of the responses. For example, the disagreement percentage was the proportion of the responses that strongly disagreed or disagreed with a certain question. Analysis of quantitative data was conducted using R Language, which is a statistical computing package.

To facilitate the understanding of the quantitative data, it was corroborated with qualitative data. Such an approach helped in providing explanations for support or opposition to some questions. Several quotes were also provided, which were retrieved by reviewing the themes and codes that had been generated in the thematic analysis stage.

Privacy and Data Protection We considered several privacy and data protection aspects. For instance, we anonymized all responses in order to hide the identity of the respondents that participated in the study. Furthermore, all the research materials, including responses received from participants, were secured in the researcher's laptop using passwords. Prior to participation, we requested consent from all potential respondents, who allowed us to use their information for the research.

Rationale behind the Interview and Survey Questions

We grouped our questions into five sections, as follows: background and demographics, generic views, e-learning tools, challenges and concerns, and students' recommendations. We created the questions based on the insights that had been gained from the related studies on the perspectives of deaf students in various other places. The rationale of creating and framing the questions the way we did was to obtain a broad picture of the challenges of online students within our research context.

5.3. Study Results

This section presents the findings of our study.

RQ_1: What are the challenges and concerns that deaf and hard-of-hearing students are having with an online education during COVID-19 pandemic?

A survey and interviews were conducted in order to obtain both quantitative and qualitative data. We have grouped our findings into seven challenges that will be discussed in this section.

As shown in Figure 5.2, we report the main challenges faced by deaf and hard-of-hearing students with an online learning education. The majority of the students (62 respondents (96.9%)) communicated that they were having network issues or unreliable internet access at home. Forty-two students (75.4%) revealed that they have no access to tools to help facilitating the study and the many type distractions at home, such as the distraction of smartphones and televisions in the same room. A moderate subset of 34 students (52.3%) were concerned about the difficulty of communicating with the instructors and the interactions were not feasible, whereas two students (3.1%) were concerned about the collaboration with their fellow students. Twenty-five students (38.5%) found that COVID-19 makes fast internet connections more critical. Twelve students (18.5 %) mentioned that the development of the COVID-19 pandemic has resulted in life-altering employment shifts across Saudi Arabia. Five students (7.7%) found that maintaining an unstructured work schedule can be difficult and hard to adjust to, whereas one of the students (1.5%) found that the challenge is centered around the lack of interactions and feelings of isolation.

In the rest of this subsection, we provide more in-depth analysis of these challenges.

Select the top three most significant challenges you face while learning from home?

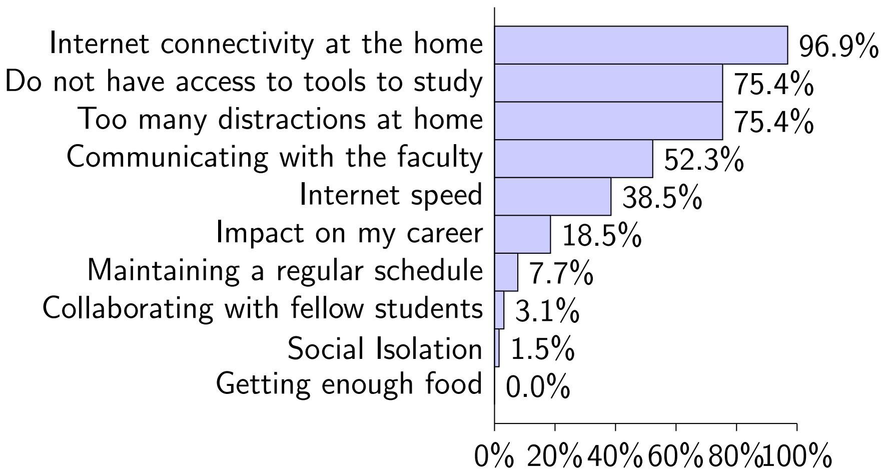

FIGURE 5.2. Presents the most challenges deaf students faced.

(1) Issues associated with accessibility:

In this section, we wanted to know whether students were accessing all the information on Blackboard, and our results are given in Figure 5.3. It was apparent that 47 students (72%) faced challenges in accessing information on the platform. **P8** noted that:

> at first, using Blackboard was extremely difficult and causing problems for getting course material and navigating the platform. Furthermore, there were different opinions because the department encourages us to use Blackboard, whereas the teachers encourage us to use different sites, so sharing class materials were very hard between the teacher and students. We ended up using social media application 'WhatsApp' to share the class materials.

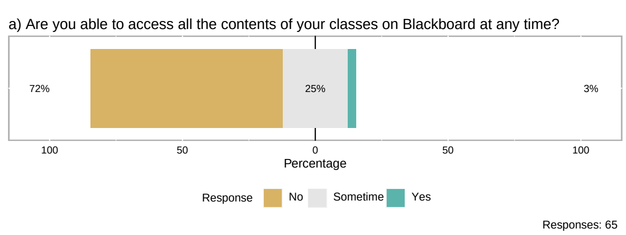

FIGURE 5.3. Accessibility of coursework materials.

We also wanted to know whether the coursework materials were accessible, and our results are given in Figure 5.4. We determined that most of the students indicated that the materials were not easily accessible. **P5** explained that:

> Blackboard was not friendly interface. I had an issue locating the exam component since there are a lot of headers and sub-headers in the navigation bar, and the font was very small hard to read.

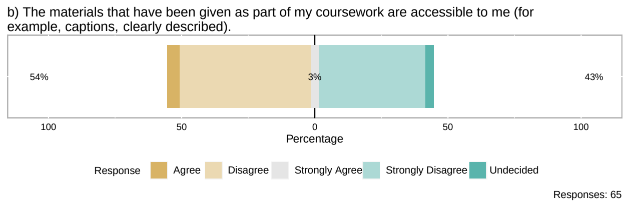

FIGURE 5.4. Accessibility of materials on Blackboard.

We also asked students whether the blackboard pages were easy to navigate, whether the videos had subtitles, whether they had enough time to complete the assignments, and whether the text sizes were easily seen. Our results are given in Figure 5.5, where a majority of the students indicated a lack of accessibility in all four aspects mentioned. **P3** explained that:

> Blackboard was in English interface, and it was hard for me to switch it to the Arabic language without any assistance. I missed many classes for this reason, and teachers were not recording the classes.

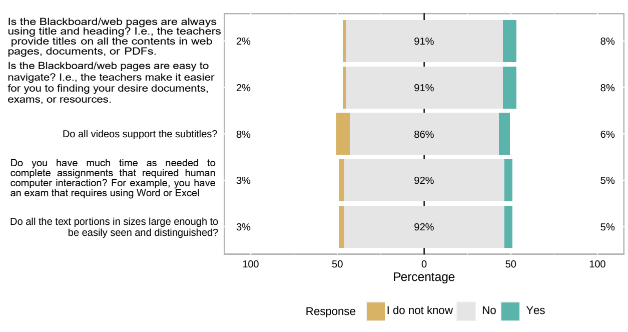

FIGURE 5.5. Responses to questions regarding to accessibility aspect.

(2) Learning problems:

In this sub-section, we wanted to identify the kind of learning challenges that the students faced during the pandemic. One of the questions we asked the respondents was whether the online learning was stressful during the pandemic, where 54 students (83%) noted that it was extremely stressful. The results are given in Figure 5.6.

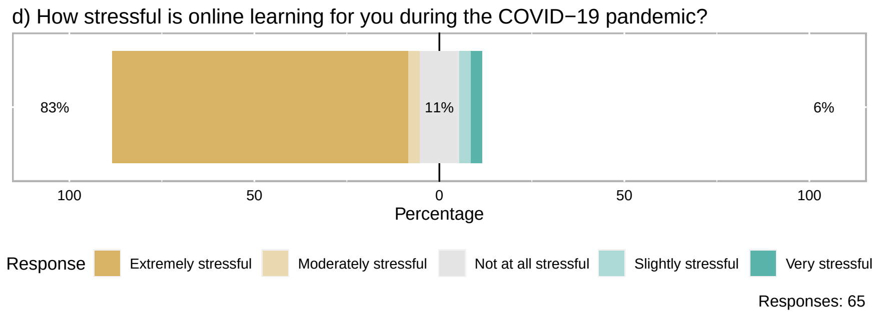

FIGURE 5.6. Stress in online learning during COVID-19 pandemic.

We also wanted to know whether online learning has been effective during COVID-19, and our results are presented in Figure 5.7. It was unfortunate that 40 (62%) of the respondents indicated that their learning was not effective at all during the pandemic.

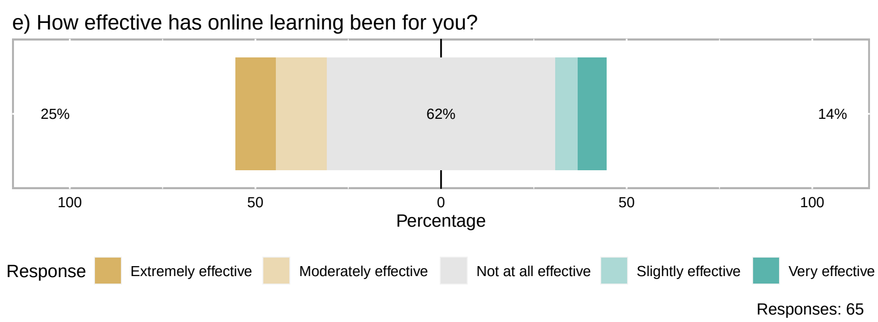

FIGURE 5.7. Effectiveness of online learning during the COVID-19 pandemic.

We also wanted to know how the educational performance of deaf students had been affected during COVID-19, and our results are given in Figure 5.8. Approximately 60 (92%) of the respondents were very worried about their performance during the period. Such outcomes were caused by teaching and learning challenges in learning that deaf students have faced in the pandemic. **P4** said that:

Less than half of the materials were covered because the time was short, and we faced difficulties understanding the material. It has to be recorded for us to see it again, but there was nothing recorded.

Such situations affected the educational performance of students.

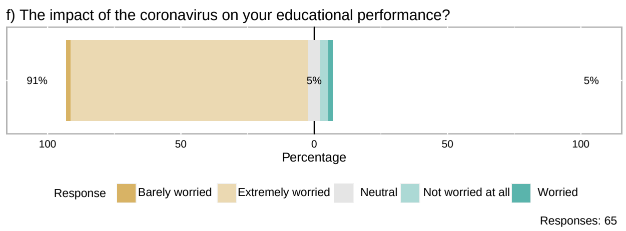

FIGURE 5.8. Impact of COVID-19 on educational performance.

(3) Challenges associated with e-learning tools:

Using online e-learning as a tool for teaching is one of the challenges and factors influencing the acceptance and use of e-learning tools, and these tools have become a key part of pandemic life. The rate of participants agreement on the usage of Google Meet, Zoom, and Backboard was 49 (75.4%), 13 (20%), and 3 (4.6%), respectively. From Figure 5.9, it is evident that Google Meet was the most preferred platform, followed by Zoom. A closer introspection reveals a shortcoming of Zoom and Blackboard over Google Meet that is the limitation of the visibility of the camera. We report samples of the participants' comments (**P2**) below to illustrate this challenge:

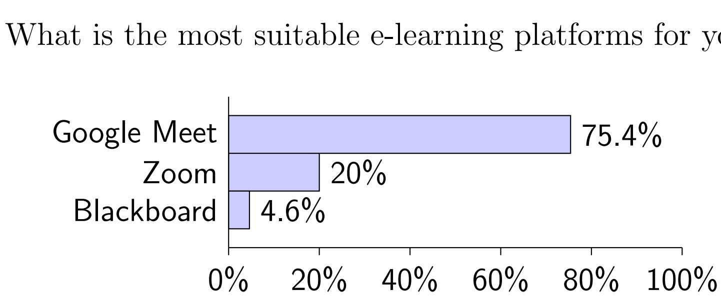

FIGURE 5.9. Presents the most suitable e-learning platforms students experienced.

> Zoom and google meet. Google meet was the best because all cameras were visible to us, whereas in Blackboard, we can only see four cameras.

(4) Problems with communication:

In this section, we wanted to know how helpful deaf teachers were to their students. Most of the students indicated that the teachers have not been helpful. Our results are presented in Figure 5.10. For those students that found their teachers not very helpful, they gave their reasons. For example, **P3** said that:

> Communicate was my biggest issue, and there were difficulties in communicating with our teachers and the department. Some teachers take a while to respond, where others not responded at all.

Other challenges were related to the congruence of the technology used on both the students and teachers. **P1** noted that:

> The teacher would call the student and sometimes presses on the name of the student or calls their name or waive at the student but in this case, the student cannot see, because the picture would be apparent only to the teacher but from the student view it was only visible for four students' camera, not all the class. So, the teacher would waive, but the student did not know because of the other four students, so it is always very late for the student to ask, and the teacher would answer: hold on, let me see which student needs me so I can show their camera.

Hence, the communication challenge greatly affected the helpfulness of the teacher towards the students. We also wanted to identify the communication means that students used to communicate with their teachers, and our results are in Figure 5.11. It was established that 60 (93.8%) of the students used WhatsApp, while 13 (20.2%) used Zoom.

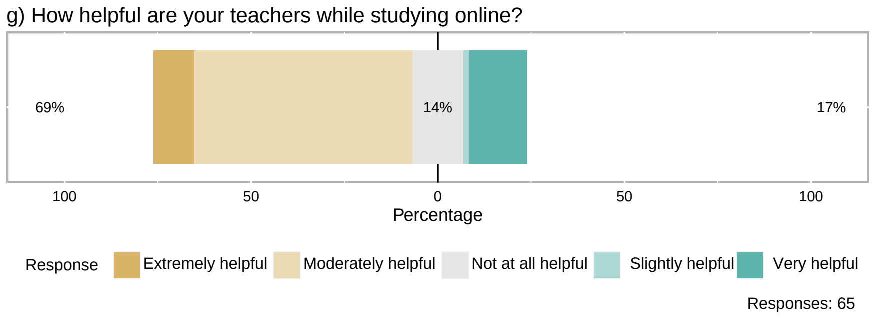

FIGURE 5.10. Teachers' helpfulness during the COVID-19 pandemic.

During your remote training, do you use any communication means, such as social media, video chat, etc. to help you communicate with your teachers or classmates? If yes, what is it?

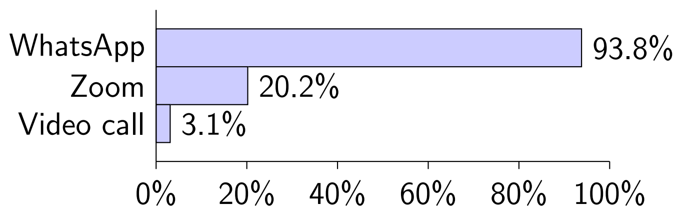

FIGURE 5.11. The communication means that students used to communicate with teachers and classmates.

(5) Inadequate support:

In most cases, deaf students require support in order to assist them in interpreting and understanding content from their teachers. Figure 5.12 shows the results that we obtained. A large majority of the students indicated that the level and kind of support that they received was not helpful. To interpret their suggestions, **P4** said that:

> We do not have an interpreter during COVID-19 as we used before the pandemic. The teacher should have a strong sign language for us to understand them. That is the most crucial thing the deaf needs in learning. The problem we encounter that some teachers sign language is weak. So as a deaf student, if the sign languages were inadequate, there isn't any benefit because the information isn't received correctly and isn't fair.

Support for deaf students is critical in the classroom and indispensable in virtual learning. Such students require a lot of assistance, which has not been forthcoming during the COVID-19 pandemic. Even when offered, it has not been adequate.

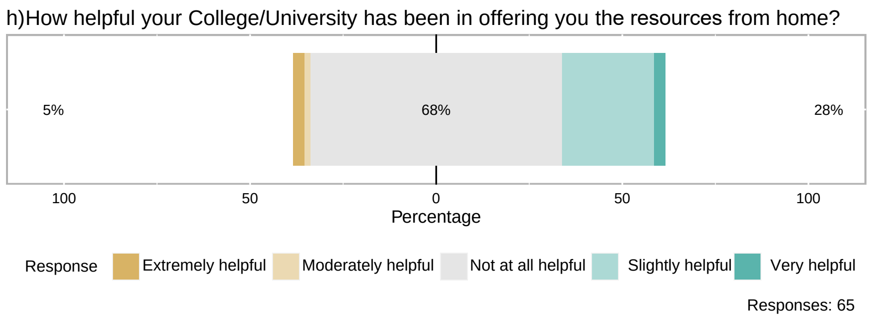

FIGURE 5.12. Institute support provided to deaf students.

(6) Technology problems:

We also wanted to know whether students were satisfied with the software that they were using for learning. Figure 5.13 shows the results that we obtained. We were surprised that 59 (91%) of the respondents were not satisfied with their online learning, while only 3% expressed satisfaction. To explain the situation, **P2** said that:

> When I encounter problems with the laptop, I switch to access the class through the phone because the internet signal was stronger on my phone than the laptop. When a teacher sends a file during the class, it was not easy to see it through a phone, so I switched back to the laptop. Then, I still struggle to get the file due to the weak internet signal.

The COVID-19 pandemic has created a sudden shift from face-to-face learning to virtual learning, and it seems that many institutions, teachers, and their students were not ready for the change. The challenges have been especially worse for deaf students who require a lot of instructional support in their learning. Such a situation may explain the technological difficulties that they have faced.

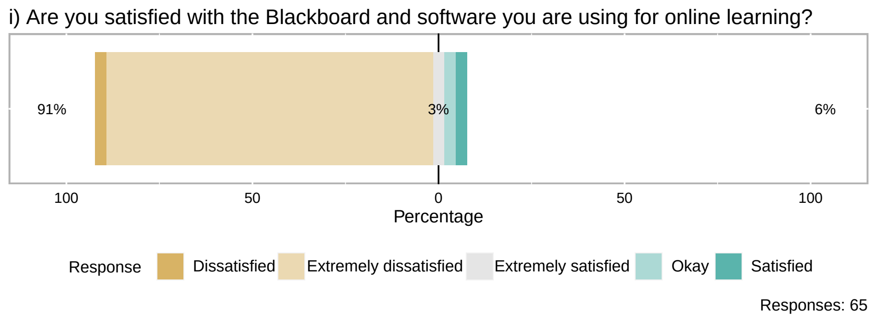

FIGURE 5.13. Students satisfaction on Blackboard platform.

(7) Distractions and Time Management Challenge:

Most of the respondents suggested that their learning during the COVID-19 period was greatly affected by the environment at home. As shown in Figure 5.14, approximately 51 (78%) of the respondents did not have a peaceful time when studying, and the majority of them faced distractions. Such an inconducive environment greatly affected the learning of the respondents at home.

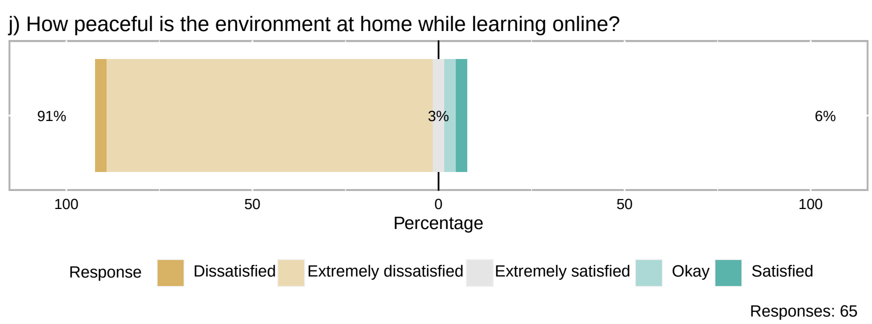

FIGURE 5.14. Students environment at home during COVID-19.

Having established that most of the students were distracted while studying from home due to the pandemic, we wanted to find out the types of distractions that they faced. In Figure 5.15, 56 (86.2%) of the students were distracted by social media, which is a common issue that affects students' learning in the modern world. Another significant issue was disruption due to the people at home, which was reported by 47 (72.3%) of the respondents.

Similar sentiments were also given by the other interviewees, who confirmed that family commitments, as well as disturbance from parents and siblings, also disrupted learning.

It was evident that distractions and time management challenges greatly affected the respondents during their studies at home, as Figure 5.16 indicated 36 (55.40%) of the students had poor time management. We established that the environment was not conducive for them.

What distractions do you face at home?

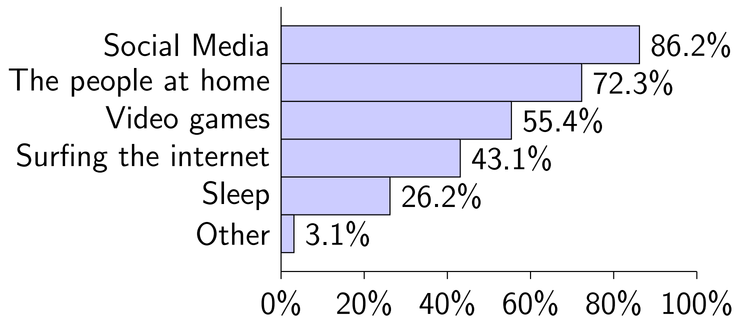

Figure 5.15. The distractions the deaf students faced.

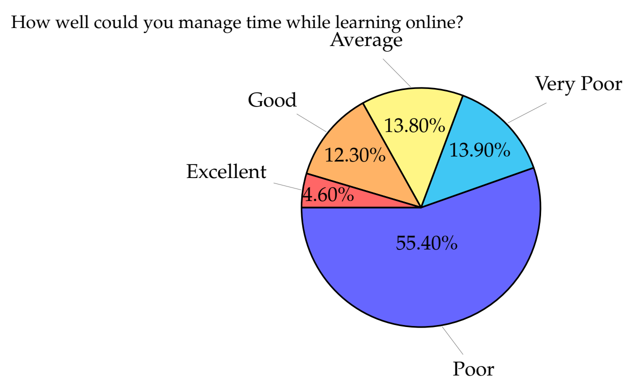

Figure 5.16. Distribution of student's time management during COVID-9.

5.4. Study Discussion

In this section, we provide the most important takeaways from our study. In total, five takeaways are discussed in the sub-sections below.

Takeaway 1: E-learning limitation. We found that the inadequacy of tools with limited cameras that can be visible to teachers and students posed great challenges to deaf education. The tools do not provide subtitles, and Zoom, provide a caption for the stream class, without supporting languages, such as Arabic. Such a finding greatly affected deaf students' learning because they cannot hear what is being said but depend on what they see on the screens. The importance of visual media in education is also indicated by Fernandes et al. [119] in Indonesia, who found that the effective use of videos greatly promoted education. Improvement of such aspects could greatly help in improving deaf education. Future researchers can compare the effectiveness of various e-learning tools to suggest which are more appropriate for deaf students.

Takeaway 2: Lack of experience. It was clear that teachers and students are not trained on the tools or do not even have good documentation to follow. Without such training, there were problems in how both students and teachers used the technology, leading to challenges in education. Deaf students, in particular, have not been trained to use the special tools needed to facilitate their education. Such findings corroborate a similar study by Krishnan et al. [168] in Malaysia that demonstrated issues in e-learning due to a lack of familiarity with technology. Other scholars can investigate the range of training programs, methods, and guidelines that would be useful in enlightening the population on how to undertake online education effectively.

Takeaway 3: Improve the accessibility and usage of learning materials. We noted that there were significant problems in the accessibility and usage of learning materials due to reasons such as a lack of subtitles and captions on videos. The importance of improving accessibility for deaf students is confirmed by studies such as Sommer [284] in the US and Mantzikos and Lappa [196] in Greece. It is important to state that deaf students need such assistance in order to understand the content in videos. More research is needed on these accessibility issues in distance/remote education for deaf students to minimize access challenges in similar contexts.

Takeaway 4: Simplify LMS systems. Our findings showed that most students were facing problems navigating through the LMS systems, the blackboard in particular. For instance, they did not know how to change languages, switch between content, or obtain course materials, among others. The problem is worse for deaf students, who cannot follow audio directions on the systems. Such technical issues were also identified by Alsadoon and Turkestani [46] as significant barriers to e-learning for deaf students. It will be important for software engineers to investigate how LMS systems can be simplified for deaf students.

Takeaway 5: Provide necessary equipment and technology. We established that a lot of equipment was needed for students to communicate with their teachers and access materials from the online platforms. For instance, it is important for students to have computers, fast internet access, among other things. The need to provide such tools and technologies was emphasized in a previous study by Krishnan et al. [168]. It will be important to identify how such software and devices can be availed to students.

5.5. Conclusion

Understanding the challenges that deaf students faced during the COVID-19 period are of paramount importance to the deaf community. In this chapter, we aimed to investigate the e-learning experience of 65 deaf students at the Technical and Vocational Training Corporation (TVTC) in Saudi Arabia. Due to the closure of physical classes, online learning using several devices in synchronous (live) and asynchronous (pre-recorded) environments has become an alternative learning method. However, this alternative learning method becomes challenging to deaf students due to the limited resources and accessibility to online learning.

5.6. Chapter Summary

: This chapter examined the e-learning experiences of deaf students during the COVID-19 era, with a particular focus on the Technical and Vocational Training Corporation (TVTC) in the Kingdom of Saudi Arabia (KSA). Students who are deaf or hard of hearing have reported that the backboard platform and course materials are not easily accessible.

In the next chapter, we analyze students' perceptions of the Blackboard mobile application's compliance with LMS accessibility guidelines.

CHAPTER 6

THE STATE OF ACCESSIBILITY IN BLACKBOARD: SURVEY AND USER REVIEWS CASE STUDY

6.1. Introduction

The use of mobile devices, and particularly smartphones, has risen in the last few years, with an estimated 3.6 billion smartphones in the world in 2020 [89]. As of 2021, the number of mobile cellular network subscribers was 8.6 billion and has risen in subsequent years [286]. With the increase of mobile devices, there has also been a rise in the number of mobile apps, which are defined as "application software designed to run on perspicacious phones, tablet computers, and other mobile devices" [149]. Accessibility in such mobile applications has gained a lot of attention in the last few years [202, 106, 59]. According to Mayordomo-Martinez et al., accessibility is defined as "the extent to which products, systems, services, environments, and facilities can be used by people from a population with the widest range of characteristics and capabilities to achieve a specified goal in a specified context of use" [202, 68]. Accessibility for mobile computing is an important topic, especially for persons with disabilities, given that there are approximately 650 million disabled people globally, representing nearly 10-15% of the world's total population [202, 214, 11]. The need to ensure accessibility in mobile applications has even led to the adoption of the Web Content Accessibility Guidelines (WCAG) 2.0 guidelines, which are used as benchmarks to evaluate how well users can seamlessly utilize a given system or application [101, 59, 230]. Therefore, it has been crucial to collect user reviews and feedback in order to improve the accessibility of mobile applications.

Various studies investigated the accessibility of LMS tools. For example, research by Li [180] assessed students' acceptance of the Blackboard LMS platform in the United States. They found a need for more compatible content and activities to be introduced for mobile learning. Another study by Alkhaldi et al. [36] extracted the benefits of making the Blackboard more accessible to students. Furthermore, a study by Kinash et al. found that students expressed positivity towards Blackboard mobile learning [159]. Yet, little is known about the extent to which Blackboard is successful in meeting the expectations of students with disabilities. With the pandemic constraints, LMS general, and Blackboard, in particular, became a unique medium for students to interact with their learning materials. Therefore, their non-compliance with accessibility guidelines can potentially hinder the learning rate of students in need of such services.

To perform the above-mentioned investigation, the goal of this paper is to gather students' perceptions of LMS compliance with accessibility guidelines, by considering Blackboard as a case study. Our study findings will highlight the unanswered accessibility issues that are being currently faced by users. To do so, we leverage recent Blackboard public user reviews, the official medium for any mobile users to share their feedback with the app maintainers. User reviews represent the *wisdom of the crowd* [40], and various successful apps have been known to interactively respond to their user's feedback, by addressing their concerns in the app's newer releases [306, 58, 209]. To the best of our knowledge, none of the previous studies assessed the accessibility of the Blackboard mobile app platform using user reviews. The research questions that we seek the answer in this study are as follows:

RQ_1: To what extent do students find the Blackboard mobile application easy to use? This research question discovers the extent to which students are able to use the Blackboard application. To do so, we performed a large-scale survey with 1,373 students, among them 65 were deaf. This 4-questions survey targets the usability of blackboard, especially when being the main learning medium, given that most universities are currently offering online courses due to the COVID-19 pandemic. We also conducted follow-up interviews with 8 students to reflect on the findings of the survey.

RQ_2: What accessibility issues are reported by the users of the Blackboard app? Since the findings of our previous research question cannot be generalized, we also decided to explore user reviews for further analysis. To address this research question, we crawled and analyzed 15,478 user reviews, which as publicly posted by Blackboard users on the Google Play Store. We used quantitative and qualitative procedures to filter out these reviews and extract only accessibility-related ones. Our findings will inform app developers of the most common accessibility issues so that they can be resolved in current and future applications. Also, our curated set of reviews is available, as part of our replication package, for reproducibility and extension purposes[1].

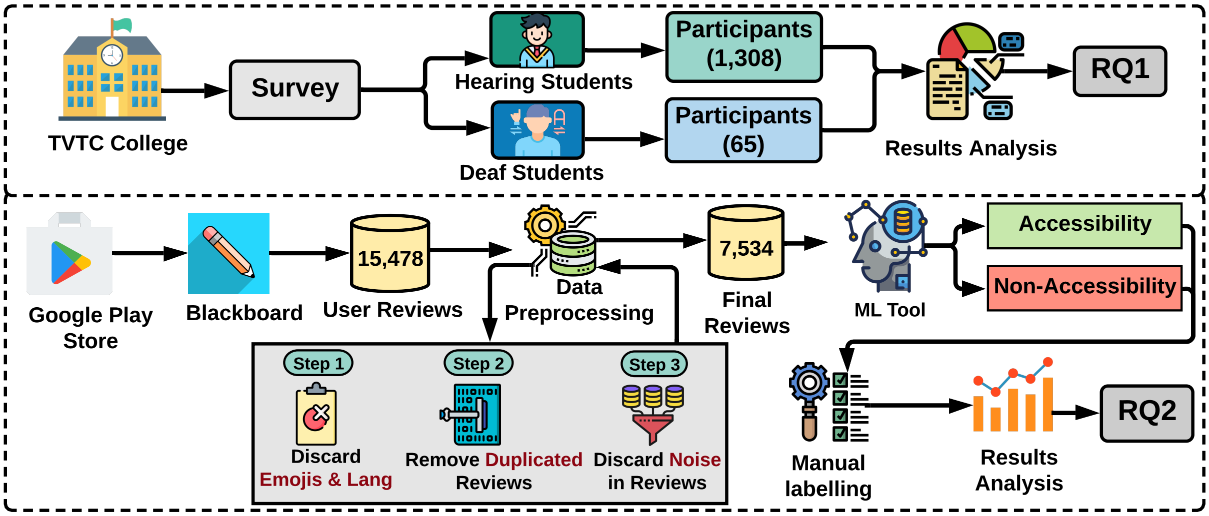

FIGURE 6.1. Approach Overview.

6.2. Study Design

This section presents the details of our approach used in this study, as provided in Figure 6.1. The information covered in this section contains the survey details, interview procedures, and user reviews.

In this section, we provide the details of our survey with 1,373 students. Then we elaborate on the follow-up interviews with 8 students. To get more insight into the users' reports about accessibility-related, we collected all the user reviews related to the Blackboard app. We detail our filtration process to identify if the user reviews were accessible or non-accessibility. We explain our manual analysis to label the user reviews based on the accessibility guidelines.

[1]https://sites.google.com/view/a11yofblackboardapp/home

TABLE 6.1. Participants demographics information. Each participant (P#) answered the interview questions.

Participant	Age	Major	Year	Student Type
P1	22	Electronic Engineering	2	Hearing
P2	24	Mechanical Engineering	4	Hearing
P3	22	Computer Networking	3	Hearing
P4	20	Electronic Engineering	1	Hearing
P5	23	Computer Technology	3	Deaf/Hard-of-hearing
P6	22	Business	2	Deaf/Hard-of-hearing
P7	21	Business	3	Deaf/Hard-of-hearing
P8	23	Computer Technology	4	Deaf/Hard-of-hearing

TABLE 6.2. Set of interviews questions.

***First-* Background and Demographics**
Years of age, and study major
Do you use the Blackboard mobile application on your phone?
***Second-* Generic Views**
How would you describe your experience while using the Blackboard mobile application?
Were you able to access the class materials via the Blackboard mobile application?
How often would you use the Blackboard mobile application?
***Third-* Accessibility Challenges**
How easy was the application to use?
How is the navigation of the Blackboard mobile application?
***Fourth-* Students Recommendations**
Are there any features that you think you need but are missing in the mobile application?
What do you think the Blackboard mobile application should improve on?

6.2.1. Survey

To get an overview of the issues surrounding the accessibility of the Blackboard LMS platform, we conducted the survey at Technical and Vocational Training Corporation (TVTC) college[2], which was the study's location and focus. Our participants were divided into hearing students (1,308 participants) and deaf students (65 participants). The questionnaire was in the Arabic language, which was the native language of the respondents. We asked four questions in the survey, which are given in Table 6.3. We sent the questionnaire using Google forms[3], which made it easier and more convenient to reach the respondents by sending them a link to the form. The analysis of the results was crucial in elaborating the

[2]https://www.tvtc.gov.sa/index-en.html

[3]https://www.google.com/forms/about/

perception of students towards the accessibility of the Blackboard platform.

TABLE 6.3. Set of survey questions.

***Q1-* What is your Gender?**
○ Male
○ Female
○ Other
***Q1-* What is your major?**
○ Computer technology and related fields
○ Business and related fields
○ Mechanical and related fields
○ Electronic and related fields
○ Electrical and related fields
○ Telecommunication and related fields
○ Food Processing Technology and Related to It (Food Processing)
○ Chemical and related fields
○ Tourism and Hospitality and related fields
○ Civil, Architectural and related fields
○ Other
***Q3-* How satisfied are you with using the Blackboard platform on your mobile phone?**
○ Extremely satisfied
○ Satisfied
○ Neutral
○ Dissatisfied
○ Extremely dissatisfied
***Q4-* Based on your experience using the Blackboard application on your mobile phone, how easy and user-friendly is the app for you?**
○ Extremely easy
○ Easy
○ Neutral
○ Difficult
○ Extremely difficult

6.2.2. Interview

To complement the survey data, we conducted interviews so that respondents would give us their views and opinions. Given the importance of validity in interviews, we utilized investigator triangulation. We used a voluntary sample of 8 students, four deaf and four hearing students. We created an interview schedule with open-ended and closed-ended questions that were focused on the research questions. The semi-structured nature of the interviews gave us a chance for respondents to give explanations about the questions they provided, giving more in-depth information. We asked nine questions in the interview, which are given in Table 6.2. We conducted the interviews using the Zoom platform[4] and used the Arabic language, which was the native language of the respondents. We offered a $25 prepaid gift card to motivate the interviewees to participate. It is important to mention that, during the interviews with deaf students, we hired a sign language interpreter as an accommodation for the students because none of us was competent in sign language. The demographic information of the interviewees is given in Table 6.1.

Following the interviews, we transcribed the data and translated it from Arabic to English. The accuracy of the translation was ensured by all the authors separately as follows: when the first author translated the work from English to Arabic, it was passed on to the second author and later to the third author, who separately compared the translated scripts to the original Arabic ones. We utilized thematic analysis to analyze the qualitative data. We perused through the interview transcripts, created codes, revised them, and deduced the themes from the pattern of answers given.

6.2.3. User Reviews Collection and Preprocessing

The first stage in this section was to get user reviews on the accessibility of the Blackboard platform, where we collected the reviews from the Google Play Store [131]. We collected all the reviews relating to Blackboard, and a total of 15,478 reviews were received. The next step was data preprocessing which involved three steps:

- **Step (1)- Discard Emojis & Languages:** We removed any reviews that only contained emojis or images, such as thumbs up and others. These reviews that contain only emojis are not useful and assist us in understanding the accessibility issues in the reviews. We also removed any reviews that were written in any other languages other than English, such as Chinese or Arabic, since our study was in the English language. After mined the collected data, we eliminated 292 reviews containing only emojis or written in a different language than English.

[4]https://zoom.us/

Table 6.4. Present an example of the eliminated reviews.

Step	Example
Emojis	👍👍
language	Algunas veces las videoconferencias no se conectan
Noise	Love it!!

- **Step (2)- Remove Duplicated Reviews:** We removed any reviews that were posted twice or severally by the same user. We eliminated such duplicate reviews because they were only repeating themselves and adding no value to the dataset. In this Step, we eliminated 1,670 reviews, and we only kept the unique reviews in the dataset.
- **Step (3)- Discard Noise in Reviews:** We discarded noise by removing all reviews that were in less than five words format, such as 'super' or 'awesome' or 'great' or 'good app' etc., which were not useful in getting user feedback. This step involved removing 5,982 reviews. Table 6.4 presents an example of three steps of data preprocessing.

User Reviews Filtering

After the data preprocessing, we remained with 7,534 reviews subjected to machine learning to know whether the user reviews were related to either accessibility or non-accessibility. To do so, we used our previous model [40] to help us automatically identify the type of user reviews. This means we put all the Blackboard user reviews as an input of the model, and the model identifies the two subsets of the dataset, accessibility, and non-accessibility as an output. We used this model to reduce the human effort that is needed to filter the user reviews manually. After we utilized the ML model, the model's output was distinguished from 3,813 reviews (50.61%) out of the 7,534 reviews as accessibility.

User Reviews Labeling

Since we are using machine learning to identify accessibility user reviews, there could be user reviews that are not related to accessibility, which can be a false positive of automated detection. Therefore, to address this issue, we performed a manual analysis intended at filtering the false positive reviews and also classifying the reviews based on the accessibility guideline given in Table 8.1. More precisely, we employed three-step iterations, which are described in content analysis method [227, 243] involving two of the authors of this paper, who have complementary expertise in line with the goal of our analysis. The first author is a software engineer with four years working on mining software repositories for mobile and publishing more than two papers in the accessibility field; the second researcher is a bachelor student in computer engineering and has two years of experience in NLP related to the mobile domain.

From this point forward: we introduce both of them as inspectors; they label a total of 3,813 user reviews, each using the approach outlined below:

Iteration (1): In the first stage: the inspectors analyze the 3,813 user reviews individually. The inspectors read all the user reviews during the analysis process and strive to identify any non-accessibility user reviews labeled as accessibility (false positive). After each inspector completed the labeling, the inspectors opened a discussion about the false positive user reviews. In the discussion, the two inspectors intend to stimulate a consensus. Afterward, the inspectors decided to eliminate 1,498 reviews. Thus, the inspectors ended up with 2,315 user reviews.

Iteration (2): In the second stage: the inspectors categorize the 2,315 accessibility user review proceeded from the first iteration. Both of the inspectors aimed to categorize based on the guidelines of the BBC standards and guidelines for mobile accessibility [63] and described in Table 8.1. During the categorizing process, the inspectors allow categorizing the user review and labeling them with one or more guidelines. After each inspector completed the labeling, the inspectors opened a discussion about the process of categorizing reviews. While discussing, the inspectors identified an issue during this iteration.

Review 1. I can't review some content. That's kind of important in school.

Review 2. It doesn't load anything anymore on my phone ever so disappointed

Review 3. The app is really easy and accessible but it doesn't always load the pages that I need

The above examples demonstrate one thing in common: they do not elaborate enough to make them seem like accessibility reviews. Afterward, the inspectors decided these false positives are still non-accessible, even though they are worded in an accessible way. Hence, the inspectors eliminated 13 reviews. So, the inspectors ended up with its iteration with 2,302 accessibility user reviews.

Iteration (3): After the second iteration, where the inspectors categorized 2,302 accessibility user reviews based on the guidelines, they concentrated on dealing with the multi-guideline reviews. There were 164 accessibility user reviews belonging to two guidelines. The inspectors have to decide for each review what is the deciding factor in labeling each guideline. The inspectors opened a discussion about the multi-guideline reviews. The following is an example of one of the user reviews that was labeled as multi-guidelines.

> This app is awful, the last one wasn't great but it was still better than this one. It sends me notifications for the same grade like 10 times, and half of the links don't work, and it's always glitching.

The review listed above has a couple of problems, yet some do not take precedence over others, as this example was labeled notifications and links guidelines. After the discussion, the inspectors decided to label the review to the primary concern of the review, which is link guidelines.

Additional validation: To extend inspection and validate the procedures executed by the inspectors, who individually examined and labeled all the accessibility user reviews relating to Blackboard reviews, the directions of Levin et al. [178] were followed by picking a 9% sample of the entire data set (239 out of 2,302 reviews). The selected sample satisfied the 95% confidence level, while the confidence interval was 6. Then, we randomly selected 239 reviews out of the 2,302 reviews. Afterward, the selected sample was given to two of the authors for labeling them. The selected data were not previously disclosed to the authors. The review procedure lasted for seven days to avoid fatigue. During the labeling process, the authors had the ability to look for terms/keywords online that they could not understand. The labeling of the data was followed by a comparison with the labeled reviews from the original dataset. We investigated the inter-rater agreement level between the two datasets using Cohen's Kappa Coefficient [94], which gave us an agreement level of 0.87. As noted by Fleiss et al. [121], an agreement level between (i.e., 0.81–1.00) implies almost *perfect agreement*.

6.3. Study Results and Discussion

RQ$_1$: To what extent do students find the Blackboard mobile application easy-to-use? In this question, we wanted to find out how easily the students were finding the Blackboard mobile application when using it. Figure 6.3 provides the responses of deaf students.

As shown in the figure, a majority (85%) of the deaf students found Blackboard to be extremely difficult. We found the high rate of dissatisfaction with Blackboard among deaf students to be a major source of concern and sought more explanation from the students. This is particularly interesting since only 12.1% of hearing students have found it to be extremely difficult. One of the most important features of an LMS system is having a proper interface that supports the needs of deaf students. Compared to hearing students, deaf students have e-learning challenges that require LMS systems to have complicity, and consistency, be navigable and have proper typography. This finding has driven our interview to seek more insights about why deaf students experience difficulties when using Blackboard. For instance, participant *P6* had a problem with the Blackboard interface and explained that:

> Blackboard was not a friendly interface. I had an issue locating the exam component since there are a lot of headers and sub-headers in the navigation bar, and the font was very small and hard to read. *(P6)*

Deaf students rely on videos that have been uploaded on Blackboard to learn. Therefore, the system should make it easy for them not only to access but also to download, videos and materials for later studies. In our study, some students could not access the materials. *P8* who faced such a challenge said that:

> I usually like education apps so that it can be more convenient to learn from my phone. However, in the blackboard app, I Can't watch videos or download files. Then, I stopped using the app and switched to the web version. *(P8)*

Deaf students have issues when videos do not have captions. It is important to note that deaf students cannot hear what the teacher is saying and solely rely on captions and, or, the guidelines that they have been given for the lecture. Captions increase the cognitive load of deaf students, their comprehension of the content they are learning, and their motivation in the subject. It is unfortunate that some of the videos on Blackboard did not have captions. *P5* noted that:

> Because I am deaf and rely on visualization, I cannot have the video caption in the app for videos. *(P5)*

The use of a Graphical User Interface (GUI) in LMS systems is particularly important for deaf students because it enables them to visualize what they are learning. It is important to have high-quality images that are informative to students. In our study, deaf students had problems when doing exams due to a lack of pictures, and P7 said:

> I liked the app, and it was easy to use. My issue always was in the exam where the picture is not shown in the exams, and I missed a few questions for that reason. *(P7)*

Although previous studies indicated that deaf students had positive perceptions towards learning management systems [115], the accessibility problems in Blackboard that have been uncovered in this study may change such a perception. We compared the results of accessibility of the Blackboard application among deaf students with hearing students, and the results are given in Figure 6.3 and Figure 6.2.

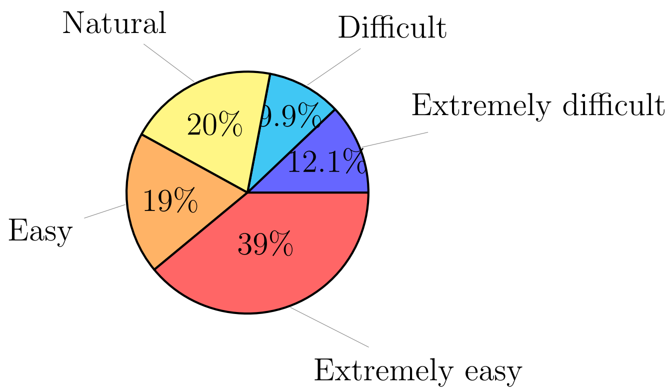

FIGURE 6.2. Percentage of the hearing Students (no. 1308) participated in the survey.

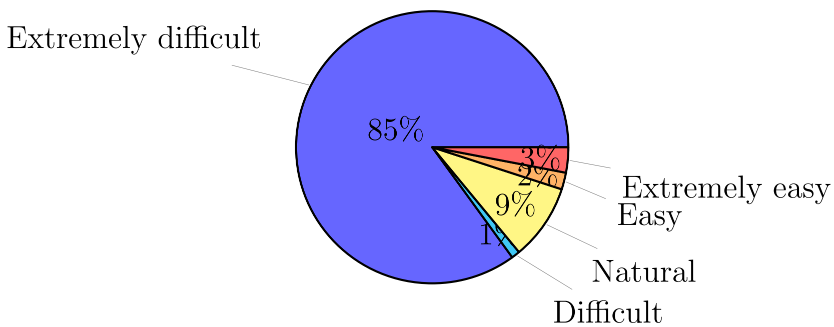

FIGURE 6.3. Percentage of the deaf Students (no. 65) participated in the survey.

Figure 6.2 provides the responses of hearing students. Compared to deaf students, the majority of the hearing students (39%) found it extremely easy to use the Blackboard LMS application, 20% found it natural, and only 10% and 12% found the platform difficult and extremely difficult respectively.

We wanted to identify the reviews of those students who found it difficult to use the platform. It is important to mention that students should be able to see all files that have been uploaded and be able to download them, provide alternative access to learning materials, and use the right format that can be opened using common programs. In our study, some students complained that materials were inaccessible. *P2* said that:

When we locked out in COVID-19 pandemic, I used the application because

> I have no laptop to access. I tried to access the classes material via the Blackboard app, but it is not opening documents. So, I have to ask my colleagues to share me the class materials. *(P2)*

The user-friendliness of an education app is very important because it indicates the ease of use of the application by students. Students should be able to efficiently and effectively utilize the app, attend lessons, take assignments, and even download materials. The site layout of the app, navigation labels, and overall design of the app should make it appealing to users, which was not the case for Blackboard according to the respondents. *P1* noted that:

> I used several education apps to teach my younger brothers, and they were easy and simple to use. For my studies, I used the blackboard. I found it a highly complex app to navigate if you don't know what you're doing. It needs to be more user-friendly since it is an educational app. *(P1)*

P3 also highlighted that some links were not working:

> I love the app, and I used to study my class materials on it. Sometimes links in the app are not working, and the show me page was not found. *(P3)*

It would be plausible to suggest that the app was not user-friendly and easy to use. In a bilingual or multilingual learning environment, app users should be able to switch between languages. For instance, in Saudi Arabia, Arabic and English are often used, even though some people only understand one of the languages. For the case of the students we interviewed, their native language was Arabic and they indicated that changing language in Blackboard was a problem. According to *P4*:

> When I downloaded the app, it was in the English language. I tried to change the language to Arabic, but I could not. I asked my father to change the language for me, but I realized that with the Arabic language the layout of the application was weird and hard to read or navigate. *(P4)*

Table 6.5. Present the results of the accessibility reviews after labeling.

Guideline	# of Reviews	% Percentage
Principle	1062	46.13%
Focus	556	24.15%
Notifications	267	11.60%
Design	145	6.30%
Forms	93	4.04%
Audio/video	69	3.00%
Links	59	2.56%
Dyn.content	38	1.65%
Images	10	0.43%
Editorial	3	0.14%
Structure	0	0.00%
Text equivalent	0	0.00%

These results are consistent with previous studies [5] that have found Blackboard LMS usable and accessible, although improvements to the platform could make it easier to use. Indeed, although the majority of the students are finding it easy to use Blackboard, others have issues relating to navigation, accessibility of materials, and changing language, among others.

RQ$_2$: What accessibility issues are reported by the users of Blackboard app? In this second research question, we were looking at the user reviews given about Blackboard on Google Play Store. Table 6.5 shows the number of reviews and their percentages in relation to the various guidelines.

From the Table 6.5, the majority of the reviews (1062 representing 46.13%) related to Principle, followed by Focus (24.15%), and Notifications (11.60%). Table 6.6 presents an example of accessibility reviews relating to each guideline. The principle guideline requires that apps be easy to operate, accessible to all users, robust in use, and understandable. For students with disabilities and those without, it requires that every user be able to navigate through the platform, know the information that is presented, understand it, and frequent upgrades are made to improve app accessibility. From the reviews we analyzed, the *principle guideline* was not met, and one of the reviews indicated that:

Table 6.6. An example of the accessibility reviews in each guideline.

Guideline	**Example**
Principle	for a student who that is Blind that uses accessibility software,JAWS,a screen reader that the Blind uses to operate on a computer,the Blackboard application does not function correctly when jaws requires that some activation requires the user to do a double tap to activate some functionctions of radio dialogues especialy when taking an exam to open the test and when navigating to the next test question the application did not move to the next question but went and submitted the exam test.no be.
Design	Please add dark mode, plus the feature to customize, organize and categorize modules, really need it. And I need to relogin almost every day, please make the login stick.
Notifications	What's the point if it doesn't even have notifications to remind you of anything? You might as well just use the browser version and auto sign in everytime.
Focus	Hard to navigate and often causes glitches in test submission. Difficult to find what you need.
Forms	After so many updated and this app still has the same problem. It does it open all the files (PDF, PowerPoint) they does not load.
Audio/video	I downloaded this app in the hopes that the inset videos would be easier to watch on my phone, the web browser doesn't allow you to see the whole video, just a cropped amount. Unfortunately, the app does not allow for videos and I just see code instead. But I would really like to see video capability added to this app. I do like that it has due dates listed for assignments, and for that alone, I am happy to keep the app for now. Overall though, it is somewhat unsatisfying.
Links	Absolutely awful. App is entirely unusable. Every link opens in an embedded browser and results in an error.
Dyn.content	Some of the worst ui design I've ever seen in an app. There are so many unnecessary animations tied to small actions; you can't click on a link without something wobbling or showing a folder opening up and papers falling out. The worst part is, the animations clearly take up a huge amount of resources because the app will actually lag before they show, which makes the whole app feel clunky and slow. If they got rid of all of these things the app would at least feel useable.
Images	Unable to view any images sent by the instructor (sent individually or inside a quiz).
Editorial	Activity steam updates after a very long time. Also when opening a new announcement it opens old announcements, not the recent one. The push notifications come on after more than 24 hours. Definitely not pleased about the app and how slow it responds to everything. Also, I can't hear my collaborate sessions although all my microphone settings have bee activated to be on and can be accessed by blackboard.

> For a student who is Blind that uses accessibility software, JAWS, a screen reader that the Blind uses to operate on a computer, the Blackboard application does not function correctly when jaws require that some activation requires the user to do a double tap to activate some functions of radio dialogues, especially when taking an exam to open the test and when navigating to the next test question the application did not move to the next question but went and submitted the exam test.no be.

The *focus guideline* requires that the page be navigable, focusable, and the input type be compatible with the users. All content should be sufficiently described with unique labels, and there should be logical and intuitive navigation order of focusable elements. In relation to the Focus principle, one of the reviews suggested that:

> Not user friendly. Especially disappointed that the word "organization" is spelled incorrectly throughout the app. How can one take an app like this seriously if the designer can't spell, and nobody else corrects them???

On the *Notifications principle*, students should be able to get notifications on updates in the courses, and upload materials, among other things even without their laptops. If the app does not provide notifications, students may miss important communication from the university or lecturers, which means missed learning opportunities. For an application such as Blackboard, which is used by many universities, it was unfortunate that some users said it lacked notifications. One of the reviews said:

> Fix the goddamn notifications I missed requirements because of this!

For the *design guideline*, a learning app should have proper interactive elements, consistent navigation options, an appropriate background color, provide light/dark models, reduce eye strain on the users, and have a good font size. There should also be visual cues, as well as form elements that have clearly associated labels. For students with visual impairments, there should be accommodations made for them so that they are able to access the materials. Some users did not find Blackboard to have a good design, and one of the reviews indicated that:

> It works, but what's with the zany button click animations etc? They're really distracting, I'd like to see them toned down a good bit.

It is important to note that the issues we found affecting accessibility from the user reviews have been reported in previous studies [313, 18]. Grouping them in terms of the guidelines helped us to classify them for easier identification. Even though Blackboard is a very common application, previous studies have shown that even such applications can benefit from improvement if people give feedback on their accessibility [113, 320]. Therefore, it would be important for developers to address the accessibility issues indicated for a better user experience.

6.4. Conclusion

In this study, we explored the student perceptive and user reviews of the Blackboard app, a common education application utilized during the COVID-19 period. We believe that our research would contribute towards the existing literature on the ease of use of educational applications and be the first study to utilize Blackboard user reviews from the Google Play Store to analyze the accessibility of the application. We established that most of the students, especially the deaf ones (85%), found it extremely difficult to use the Blackboard application. Similar results were established when we analyzed user reviews from the Google Play Store, where (31%) of the user reviews were related to accessibility. Some of the reasons that made the Blackboard application inaccessible were the lack of notifications, unavailability of captions, distracting animations, difficulty in changing language, and lack of captions on videos. Our results provide valuable insights for educational application developers on improving the accessibility and usability of the applications.

6.5. Chapter Summary

: This chapter examined how students perceive the compliance of the Blackboard mobile application with LMS accessibility standards. In recent years, the usage of mobile devices, notably smartphones, has increased. This indicates that the mobile application will obtain a large number of reviews. With manual analysis, it becomes exceedingly difficult for engineers to carefully examine every user feedback.

In the next chapter, we use supervised learning to formulate the identification of accessibility reviews as a binary classification issue.

CHAPTER 7

FINDING THE NEEDLE IN A HAYSTACK: ON THE AUTOMATIC IDENTIFICATION OF ACCESSIBILITY USER REVIEWS

7.1. Introduction

Many mobile applications (apps) have poor accessibility which makes it difficult for people with disabilities to use such apps [319, 48, 257, 255]. Researchers presented several methods, tools, frameworks, and guidelines to support developers in creating accessible mobile applications [242, 88, 256, 59, 114, 294, 127, 260, 174]. However, many software developers and designers still do not incorporate accessibility into their software development process due to lack of awareness or lack of resources, e.g., budget and time, [244, 100, 250]. In this paper, we present a method that can help software developers to quickly become aware of specific accessibility problems with their apps that the users encountered. Our method is based on automatically identifying app reviews that users write on app stores, e.g., App Store[1], Google Play[2] and Amazon Appstore[3], where these reviews express an accessibility-related feedback [40].

Analyzing app reviews was used by technology professionals to identify issues with their mobile apps [192, 91, 181]. However, accessibility in user reviews is rarely studied especially for mobile applications [112].

Identifying accessibility-related reviews is currently done using two main methods: manual identification and automatic detection [112]. The manual identification approach is time consuming especially with the vast number of reviews that users upload to the app

[1]https://www.apple.com/ios/app-store/

[2]https://play.google.com/store

[3]https://www.amazon.com/mobile-apps/b?ie=UTF8&node=2350149011

This entire chapter is reproduced from AlOmar, Eman Abdullah, Wajdi Aljedaani, Murtaza Tamjeed, Mohamed Wiem Mkaouer, and Yasmine N. El-Glaly, "Finding the needle in a haystack: On the automatic identification of accessibility user reviews," in Proceedings of the 2021 CHI conference on human factors in computing systems, (2021), 1-15, https://dl.acm.org/doi/abs/10.1145/3411764.3445281, with permission from the Association for Computing Machinery.

stores, and so it becomes impractical. The automated detection method employs a string-matching technique as a predefined set of keywords are searched for in the app reviews [112]. These keywords were extracted from the British Broadcasting Corporation (BBC) recommendations for mobile accessibility [62]. While this method sounds more practical than the manual one, it has its own drawbacks: the string-matching technique ignores that keywords derived from guidelines do not necessarily match the words expressed in reviews posted by users. This mismatch includes but not limited to situations when the keywords are incorrectly spelled by users.

More importantly, the presence of certain keywords in a review does not necessarily mean that the review is about accessibility. For example, consider the following reviews from Eler et al. dataset [112]:

> This is the closest game to my old 2001 Kyocera 2235's inbuilt game 'Cavern crawler'. Everything is so simple and easy to comprehend but that doesn't mean that it is easy to complete right off of the bat. Going into the sewers almost literally blind (sight and knowledge of goods in inventory) is a great touch too. Keep at it. I'll support you at least in donations.

This review contains a set of keywords that could indicate accessibility (e.g., old, blind and sight) but it is not an accessibility review. In this review, the word old refers to a device rather than a person. The words blind and sight refer to knowledge of goods in the game rather than describing a player's vision. Therefore, the discovery of accessibility reviews heavily relies on the *context*, and so, simply searching for their existence in the review text is inefficient. Due to the overhead of the manual identification, and the high false-positiveness of the automated detection, these two methods remain impractical for developers to use, and so, accessibility reviews remain hard to identify and to prioritize for correction. To address this challenge, it is critical to design a solution with *learning capabilities*, which can take a set of examples that are known to be accessibility reviews, and another set of examples that are not about accessibility but do contain accessibility-related keywords, and learn how to distinguish between them. Therefore, in this paper, ***we use supervised learning to***

formulate the identification of accessibility reviews as a binary classification problem. This model takes a set of accessibility reviews, obtained by manual inspection, in a previous study [112] as input, we deploy state-of-the-art, machine learning models to *learn* the *features*, i.e., textual patterns that are representative of accessibility reviews. In contrast to relying on words derived from guidelines, our solution extracts *features* (i.e., words and patterns) from actual user reviews and learns from them. This is critical because there is a *semantic gap* between the guidelines, formally written on an abstract level, and technology-specific keywords. By features, we refer to a keyword or a set of keywords extracted from accessibility-related reviews that are not only important for classification algorithms, but they can also be useful for developers to understand accessibility-related issues and features in their apps. The patterns can be about an app feature that supports accessibility (e.g., font customization, page zooming or speed control); about assistive technology (e.g., word prediction, text to speech or voice over) as well as about disability comments (e.g., low vision, handicapped, deaf or blind). Particularly, we addressed the following three research questions in our study:

RQ1: *To what extent machine learning models can accurately distinguish accessibility reviews from non-accessibility reviews?*

To answer this research question, we rely on a manually curated dataset of 2,663 accessibility reviews, which we augment with another 2,663 non-accessibility reviews. Then we perform a comparative study between state-of-the-art binary classification models, to identify the best model that can properly detect accessibility reviews, from non-accessibility reviews.

RQ2: *How effective is our machine learning approach in identifying accessibility reviews?*

Opting for a complex solution, i.e., supervised learning, has its own challenges, as models need to be trained, parameter tuned, and maintained, etc. To justify our choice of such solution, we compare the best performing model, from the previous research question, with two baselines: the string-matching method, and the random classifier. This research question verifies whether a simpler solution can convey competitive results.

RQ3: *What is the size of the training dataset needed for the classification to effectively identify accessibility reviews?*

In this research question, we empirically extract the minimum number of training instances, i.e., accessibility reviews, needed for our best performing model, to achieve its best performance. Such information is useful for practitioners, to estimate the amount of manual work needs to be done (i.e., preparation of training data) to design this solution.

We performed our experiments using a dataset of 5,326 user reviews, provided by a previous study [112]. Our comparative study has shown that the *Boosted Decision Trees* model (BDTs-model) has the best performance among other 8 state-of-the-art models. Then, we compared our BDTs-model, against two baselines: (1) string-matching algorithm and (2) a random classifier. Our approach provided a significant improvement in the identification of accessibility reviews, outperforming the baseline-1 (keyword-based detector) by 1.574 times, and surpassing the baseline-2 (random classifier) by 39.434 times.

The contributions of this chapter are:

(1) We present an action research contribution that privileges societal benefit through helping developers automatically detect accessibility-related reviews and filter out irrelevant reviews. We make our model and datasets publicly available [4] for researchers to replicate and extend, and for practitioners to use our web service and filter down their user reviews.

(2) We show that we need a relatively small dataset (i.e., 1500 reviews) for training to achieve 85% or higher F1-Measure, outperforming state-of-the-art string-matching methods. However, the F1-measure score improves as we add to the training dataset.

[4] https://smilevo.github.io/access/

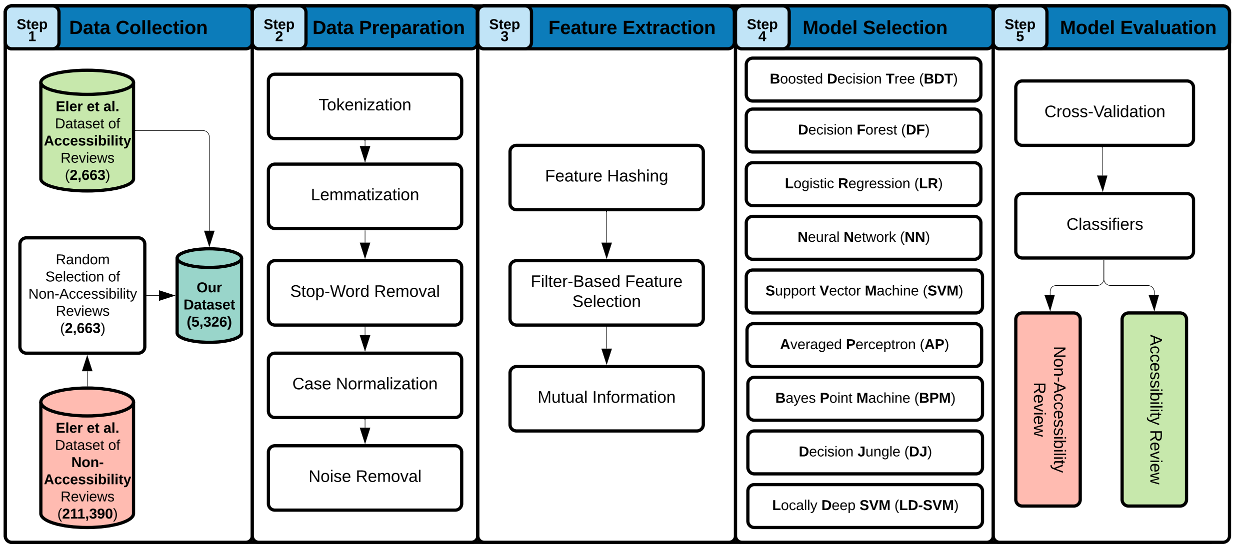

FIGURE 7.1. Accessibility app review classification process.

7.2. Accessibility App Review Classification

The main goal of this work is to automatically identify accessibility-related reviews in a large dataset of app reviews. Our approach takes a set of reviews as input and makes a binary decision on whether the review is accessibility pertaining or not, i.e., classifying app reviews (for simplicity we refer to them as *accessibility reviews* and *non-accessibility reviews*). To be able to do so, we built a classification model using a corpus of reviews and current classification techniques. We then used the classification model to predict types of new app reviews. Figure 10.1 provides an overview of the process used in the detection of accessibility reviews. Our approach follows five main steps:

(1) **Data Collection:** We used a dataset of app reviews along with their ground truth categories previously identified through manual inspection [112] as input for training purposes.
(2) **Data Preparation:** We applied data cleansing and text preprocessing on this set to improve the *reviews text* for the learning algorithms. Some of the text preprocessing procedures we used are namely, tokenizing, lemmatizing, removing stop words, and removing capitalization.
(3) **Feature Extraction:** We used Feature Hashing [314] to extract features (i.e.,

words) from the preprocessed review text to create a structured feature space.

(4) **Model Selection and Tuning:** We examined a total of nine classification algorithms to evaluate the performance of the model for prediction. These classifiers were chosen because they are commonly used for classification of text such as app reviews [148, 163]. After training and evaluating the model, we used a testing dataset to challenge the performance of the model. Since the model has already learned from the N-Gram vocabulary and their weights discussed in Section 7.2.3 from the training dataset, the classifier output predicted-labels and probability-scores for the testing dataset. Since an app review is a plain text in our case, we follow the approach provided by Kowsari et al. [165] that discusses trending techniques and algorithms for text classification, similar to [42, 43].

(5) **Model Evaluation:** We built a training set using the extracted features for the model to learn from.

7.2.1. Data Collection

The dataset, used for this study, and shown in Table 7.1, is a collection of these 2,663 accessibility reviews, manually validated by Eler et al. [112]. The collected reviews are extracted from across 701 apps, belonging to 15 different categories, as shown in Figure **??**. This dataset excluded all apps under the Theming and System categories, since they usually do not have any interface associated with them. Eler et al. [112] started with collecting 214,053 reviews, then they performed the string-matching using 213 keywords to filter down reviews and keep only those who potentially may contains information related to accessibility. These keywords are derived from 54 BBC recommendations proposed for mobile accessibility. The string-matching reduced the reviews from 214,053 to 5,076 candidate accessibility reviews. However, the manual inspection of these candidate reviews found that only 2,663 were true positives.

In order for us to verify the previous manual labeling of the reviews, we followed the process of Levin et al. [179] and randomly selected a 9% sample of reviews, i.e., 243 out of the 2,663 reviews. This quantity roughly equates to a sample size with a confidence level of

95% and a confidence interval of 6. Then we randomly added another 243 non-accessibility reviews, to end up with a total of 486 reviews. Afterward, one researcher labeled them. The selected data was not exposed to the researcher before. The review process was given a period of 7 days, to avoid fatigue, and the researcher had the opportunity to search online for any keywords they could not understand, during the labeling process. Once the data was labeled, we positioned our labeling against the original labeling of the reviews, from the dataset. We used Cohen's Kappa coefficient [94] to evaluate the inter-rater agreement level for the categorical classes. We achieved an agreement level of 0.82. According to Fleiss et al. [121], these agreement values are considered to have an almost *perfect agreement* (i.e., 0.81–1.00).

TABLE 7.1. Statistics of the dataset.

Number of Apps	701
App Categories	15
All Reviews	214,053
Accessibility Reviews	2,663

To prepare training data for the binary classification of app reviews we created two groups of app reviews: (1) reviews indicating accessibility and (2) reviews not related to accessibility. For the accessibility reviews, we used the set of 2,663 reviews previously identified and validated as accessibility reviews through manual inspection by Eler et al. [112]. Since class starvation or an imbalanced training set (i.e., not having equal size of both groups) could decrease the performance of a classification model [176, 179], we need to select an equal number of non-accessibility reviews for the training.

To efficiently train a classifier, it is important for the negative set to be as *close* as possible to the positive set. Therefore, we chose the negative set to be populated using the discarded reviews of the original authors, during their manual process. These discarded reviews tend to contain some keywords that are relevant to accessibility, but they were found to be conveying another meaning, and that is what we want our model to learn. Since the subset of discarded reviews was 2,413, we randomly selected reviews from the Eler et al. [112] remaining reviews dataset, so that these reviews are also extracted from the same apps, and most likely to contain some keywords that overlap with our true positive set.

To decide on the number of reviews necessary for training purposes, we reviewed the thresholds used in several text classification studies. The highest number of text documents used in comparable studies [179, 176, 42] was around 2000 text documents. Since our goal was to provide the model with sufficient reviews that could represent all possible accessibility topics, unlike existing works we chose a total of 5,326 reviews for the model creation and validation. However, we did evaluate our model with different sizes of training sets to understand the size of the training set that yields the best results. We report the results of our evaluations with regard to the testing of different training sizes in Section 7.3.

7.2.2. Data Preparation

Upon completion of the data collection phase, we applied a common approach explained in [165] for text preprocessing, similar to [42, 43]. For a model to classify text documents correctly, the text needs to be cleaned and preprocessed. To preprocess the app reviews text, we used natural language processing techniques, built-in the Microsoft Azure [56], such as tokenizing, lemmatizing, removing stop words, and removing capitalization.

Tokenization: is the process of splitting natural text data into tokens, or meaningful elements, that contain no white space. We tokenized app reviews by breaking them into their constituent set of words.

Lemmatization: is the process of getting the basic form of a word by either removing the suffix of a word or replacing the suffix of a word with a different one. It is also the process of reducing the number of unique occurrences of similar words. We used this preprocessing technique to represent words in their canonical form in order to reduce the number of unique occurrences of similar text tokens.

Stop-Word Removal: We removed words such as (*is, am, are, if, for, the*, etc.) that do not play any good role in classification.

Case Normalization: Since we wanted the same words with different font cases (e.g., Accessibility and accessibility) to be treated as the same word, we converted original review texts to lower case. This type of text cleansing helps us avoid having repeated features differing only in the letter case. We realize that in some cases a user can identify themselves as 'Deaf' with uppercase 'D' to express their cultural identity in their review which is different from 'deaf'. However, as our classifier is a binary classifier that only distinguishes accessibility reviews from the rest, the words 'Deaf' and 'deaf' will yield the same classification result. Hence, case normalization in this context is safe and will not overrule users' expressions.

Noise Removal: We removed any noise that could deteriorate classification performance and confuse the model when learning. Examples of the noise we removed include removing special characters, numbers, symbols, email addresses and URLs.

7.2.3. Feature Extraction

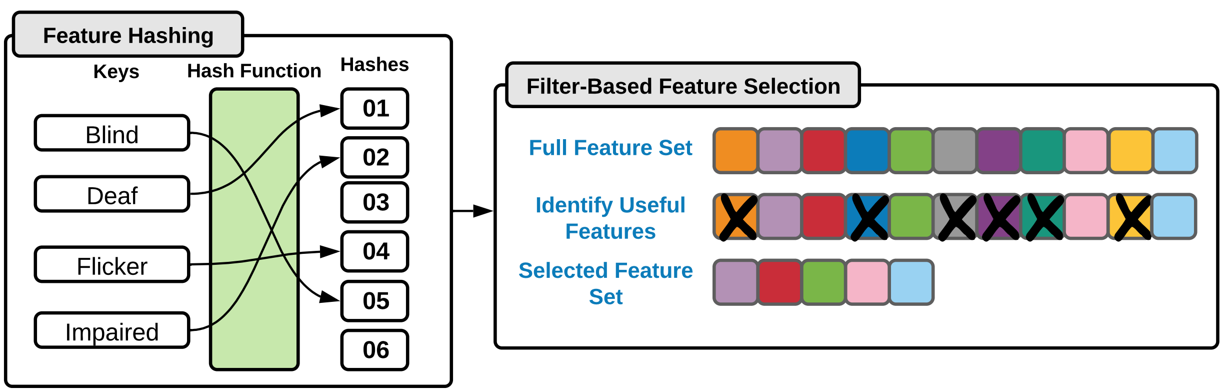

FIGURE 7.2. An example of feature hashing and feature selection process in feature extraction stage.

After cleansing and preprocessing the reviews text, we extracted features from the preprocessed text that matter the most in distinguishing between the two classes in classification. Particularly, we used the ***Feature Hashing*** technique for feature extraction. Feature Hashing is a technique that operates on high-dimensional text documents used as input in a machine learning model, to map string values directly into encoded features and represent them as integers [274, 314]. This technique helps to reduce dimensionality and to make the feature weights lookup more efficient. Internally, the Feature Hashing technique creates a dictionary of N-Grams. We used bigrams in our classification since it greatly improves the performance of text classification [292]. Generally, N-Grams have more meaning and semantic than isolated words. For example, the word font does not provide enough information by itself. However, when N-Gram features extracted from reviews, e.g., small font, font customization, font size, etc., the word font can indicate accessibility reviews. We discuss in details the features of our model (i.e., keywords and bigrams) in Section 7.3. We used Mutual Information filter-based feature selection. Mutual Information is a technique that measure how much a variable contributes towards reducing uncertainty about the value of another variable in order to identify features with the greatest predictive power. In fact, this feature set is the training set that the model learns from. In Figure 7.2, we illustrate how Feature Hashing applied to the text which was being transformed to a dictionary, as well as the process of the filter-based feature selection.

7.2.4. Model Selection and Tuning

Selecting an appropriate classifier for optimal classification is a challenging task by itself [120]. In this study, we are tackling a two-class classification problem as we are categorizing app reviews into two groups, accessibility and non-accessibility. Because we already have a predefined set of classes, our approach relies on supervised machine learning algorithms to assign each review into one of the two categories. We tested nine different classification algorithms as to see which one provides the best results in the context of accessibility and app reviews classification. The tested classifiers are: Logistic Regression (LR), Decision Forest (DF), Boosted Decision Tree (BDT), Neural Network (NN), Support Vector Machine (SVM), Averaged Perceptron (AP), Bayes Point Machine (BPM), Decision Jungle (DJ), and Locally Deep SVM (LD-SVM). We adopted these classifier algorithms because they are commonly utilized in the literature of software-related text classification [175, 132, 184, 234, 42]. Below is a brief description of each of the classification algorithms used in this study.

- **Logistic Regression (LR)**[53] is a linear classifiers that predicts the probability of an outcome by fitting data to a logistic function.

- **Decision Forest (DF)**[249]: is a tree-based learner that builds many classification trees. A specific classification is associated with each tree produces. To classify a new object, DF chooses the classification that has the most votes over all other trees.
- **Boosting Decision Tree (BDT)**[123]: is an ensemble learning method in which the second tree corrects for the errors of the first tree, the third tree corrects for the errors of the first and second trees, and so forth. Predictions are based on the entire ensemble of trees together that makes the prediction.
- **Neural Network (NN)**[135]: is a set of interconnected layers. The inputs are the first layer that are connected to an output layer by an acyclic graph.
- **Support Vector Machine (SVM)** [318]: is a learner that constructs hyper-plane(s) in n-dimensional space.
- **Averaged Perceptron (AP)**[95] is a simple version of Neural Network. The inputs are classified into several outputs based on a linear function, and then combined with a set of weights that are derived from the feature vector.
- **Bayes Point Machine (BPM)**[143]: is an algorithm that uses a Bayesian approach to linear classification called the Bayes Point Machine. This algorithm approximates the optimal Bayesian average by choosing one average classifier, the Bayes Point.
- **Decision Jungle (DJ)**[276]: is a recent extension to decision forests. It consists of an ensemble of decision directed acyclic graphs (DAGs).
- **Locally Deep SVM (LD-SVM)**[153]: is a classifier that has been developed for an effencient non-linear SVM prediction.

We compared all the nine classifiers based on their common statistical measures such as precision, recall, accuracy, and F1-measure. These experiments were performed on the Azure ML platform because it provides a built-in web service once the classification model is deployed. We report the results of our classifier comparison and evaluation in Section 7.3.

We use grid search cross validation [269], a tuning method that performs exhaustive search over specified parameter values for an estimator, for tuning of our selected ML models. In order to facilitate the replication of our results, we provide the selected main parameters for ML techniques as shown in Table 10.3.

7.2.5. Model Evaluation

We assess the performance of our selected models based on the following four measurement aspects:

- **Precision** = $\frac{tp}{tp+fp}$: is a statistic that calculates the accurate number of correct predictions out of all the input sample.
- **Recall** = $\frac{tp}{tp+fn}$: is a statistic that calculates the accurate number of positive predictions that was actually observed in the actual class.
- **Accuracy** = $\frac{TP+TN}{TP+TN+FP+FN}$: is a statistic that calculates the accurate number of
- **F1-measure** = $\frac{2 \cdot P \cdot R}{P+R}$: is a a statistic that calculates the accuracy from the precision and recall.

Here TP denotes True Positive, TN denotes True Negative, FP denotes False Positive, and FN denotes False Negative. These metrics participation in measurement for a classifier's output.

- **True Positive (TP):** This parameter determines the predictions labeled correctly by the classifier as positive.
- **True Negative (TN):** This parameter determines the correct number of negative predictions.
- **False Positive (FP):** This parameter determines the number of instances (negatives) that were presumed as positive instances by the classifier by mistake.
- **False Negative (FN):** This parameter determines the number of positive instances that were falsely assumed to be as negative instances by the classifier.

TABLE 7.2. Summary of the hyperparameter in machine learning algorithm.

Classifier	Hyperparameter	Default	Description
	optimiz˙tol	1E-07	Optimization tolerance
LR	1˙weight	1	L1 regularization weight
	L2˙weight	1	L2 regularization weight
	memory˙L˙BFGS	20	Memory size for L-BFGS
	n˙estimators	8	Number of decision trees
DF	max˙depth	32	Maximum depth of the decision trees
	n˙samples˙leaf	125	Number of random splits per node
	min˙samples˙split	1	Minimum number of samples per leaf node
	max˙n˙leaf	20	Maximum number of leaves per tree
BDT	min˙samples˙leaf	10	Minimum number of samples per leaf node
	learning˙rate	0.2	Learning rate
	n˙tree	100	Number of trees constructed
	n˙nodes	100	Number of hidden nodes
	learning˙rate	0.1	Learning rate
NN	n˙learning˙rate	100	Number of learning iterations
	learning˙rate˙weights	0.1	Initial learning weights diameter
	momentum	0	Momentum
	n˙iter	1	Number of iterations
SVM	Lambda	0.001	Lambda
	learning˙rate	1	Learning rate
AP	m˙iter	10	Maximum number of iterations
BPM	n˙training˙iter	30	Number of training iterations
	n˙estimators	8	Number of decision directed acyclic graphs
DJ	max˙depth	32	Maximum depth of the decision directed acyclic graphs
	max˙width	128	Maximum of the decision directed acyclic graphs
	n˙optimiz	2048	Number of optimization steps per decision directed acyclic graphs layer
	max˙depth	3	Depth of the tree
	lam˙weight	0.1	Lambda weight
LD-SVM	n˙theta	0.01	Lambda Theta
	n˙theta˙Prime	0.01	Lambda Theta Prime
	n˙sigmoid	1	Sigmoid sharpness
	n˙iter	15000	Number of iterations

Cross-Validation. We applied a 10-fold cross-validation technique to evaluate the variability and reliability of our models. For each model, we split our dataset into 10 folds containing the equal size of app reviews. Then, we performed 10 evaluations with various testing datasets wherein each evaluation 9 folds were used as a training dataset and the other fold was used as a testing dataset. Put differently, unlike other approach that is dependent on just one train-test split, when evaluating our model using 10-fold cross-validation, we train on multiple train-test splits in which one fold is left as a holdout data set, so it is unseen during the training. This approach is considered the preferred method as it gives us a better indication of how well our model performs on unseen data. We aggregated the results of the 10 evaluations and reported the average performance tested with multiple models.

7.3. Experimental Results and Evaluation

In this section, we review the results of our experiments to evaluate the performance of our approach. For evaluating various accessibility classification models, we used standard statistical measures (*Precision*, *Recall*, *Accuracy*, *F1-measure*). Using the evaluation results, we provide answers to our research questions.

RQ1. To what extent machine learning models can accurately distinguish accessibility reviews from non-accessibility reviews?

We conducted an experiment to determine if the automatic classification of user reviews using machine learning techniques can be performed with high accuracy. We wanted to understand the opportunities and limitations of the machine learning technique in automatically detecting accessibility reviews.

We compared the nine classification algorithms tested in this study with respect to precision, recall, accuracy and F1-measure and reported the results as shown in Figure 7.3. The accuracy and F1-measure of the Boosted Decision Trees model (BDTs-model) is clearly higher than its competitors for the classification of accessibility reviews. The BDTs-model with the accuracy of 90.6% and F1-measure of 90.7%, outperformed other classification algorithms. Figure 7.3 also shows that the Bayes Point Machine (BPM) and Averaged Perceptron (AP) with F1-measure of 88.7% and 88.3% respectively, yielded higher predictive power after the Boosted Decision Trees.

The fact that BDTs-model achieved top performance rate can be explained by the fact that a boosted decision tree aggregates several learnings since it is an ensemble learning method. In the ensemble method, the errors of the first tree are fixed by the second tree, and the errors of the second tree are fixed by the third, and so on. In this method, the entire ensemble trees together form the prediction.

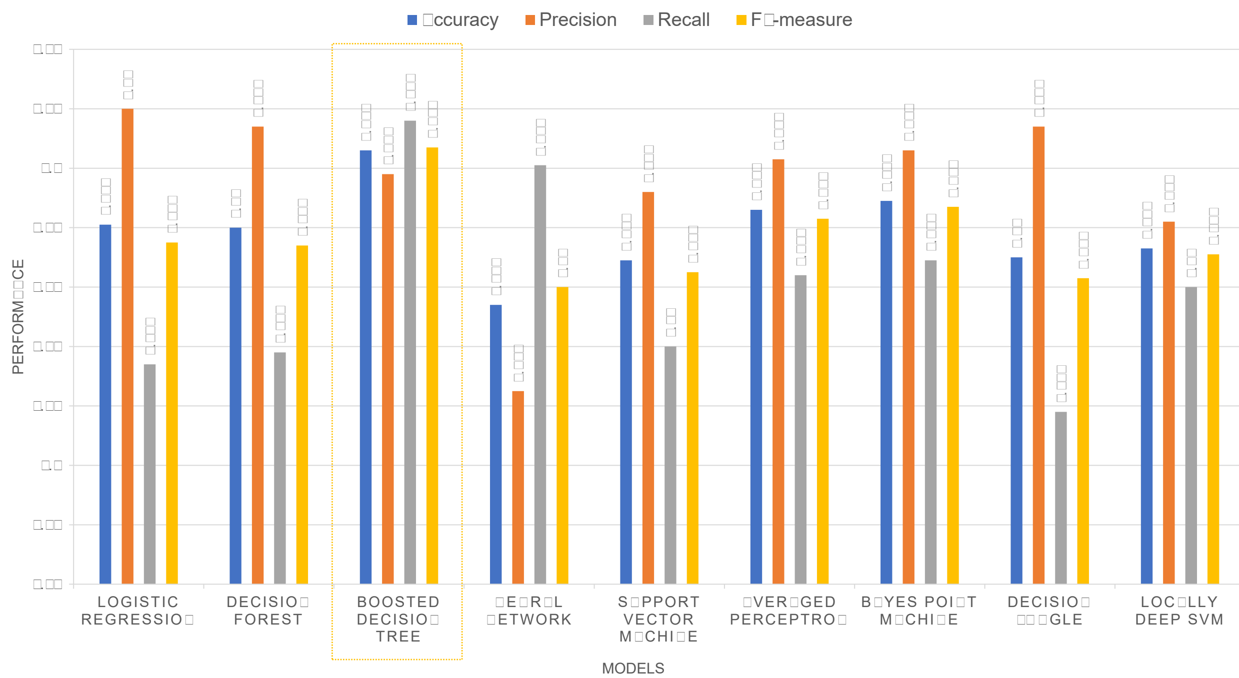

Figure 7.3. Comparison between binary classifiers, in terms of precision, recall, accuracy, and F1-measure.

To further understand how these models distilled the text of the reviews into features, we extract keywords that were trending in our dataset, that we enumerate in Table 7.4. It is important to note that the majority of these keywords were identified by the BBC recommendations for mobile accessibility, however, not all of these keywords were found to be useful for our best performing classifier, i.e., BDTs-model. In Table 7.4, we report in bold, the features that were influential in increasing the accuracy of the trained Boosted Decision Trees. Such finding does not necessarily deny the relevance of the remaining keywords in describing accessibility related issues, but the fact that they were not selected, indicates their existence in non-accessibility related reviews. Keywords such as dark mode or mute, while being used in the BBC guidelines, are also known to be used in general usability contexts. For example, the keyword mute tends to be frequently used in reviews related to media and video players, where sound is one of the main features of the app.

Further, on a more qualitative sense, we examine the set of frequently occurring bigrams for the keywords (reported in Table 7.4) that are strongly correlated to the accessibility review . Bigram corresponds to a sequence of two adjacent words in a sentence to help better understanding the context for the given terms. By analyzing the natural language in the accessibility review, we obtain more specific accessibility review-related terminology. Table 7.3 presents the frequently occurring bigrams in the review. Looking at these terms, we see that developers are either commenting on the features of the apps (e.g., easily accessibile, good text reflow, great for visually impaired), or they are discussing accessibility issues with their products pointing out that the apps need to be improved (e.g., terribly hard to see, no visual cue, cant read).

The findings, illustrated in Tables 7.4 and 7.3 indicate a potential variation of how users typically state their accessibility needs. While it seems intuitive, there are no studies that focused on extracting such information in a structured manner to facilitate the identification of such accessibility problems by the app maintainers.

Although a high classification performance of our BDTs-model has been demonstrated in Figure 7.3, there are some limitations that lead BDTs-model to output some misclassified reviews as illustrated in Table 7.5. According to our thorough analysis, we notice that the misclassification of our model can be related to:

- False positive instances caused by the format of reporting user perspective of the apps. The examples in the table show that different expression about the apps like simple or headache can be confusing to the classifier and hence it misclassified these reviews.
- False negative instances caused by the format of reporting a specific feature of the apps. As shown in the table, the users commented on a specific feature such as

functioning reader and caller ID. The BDTs-model will wrongly classify it because these could be seen as an accessibility-related features.

It is worth noting that the above misclassifications do not have a large influence on the overall performance of the BDTs model. Only a small number of reviews are wrongly classified by our model.

TABLE 7.3. A sample of frequently occuring bigrams for the keywords that are strongly correlated to accessibilty review by our model.

Bigram			
cannot see	**accessibility**	**readable**	**hard to see**
cannot see anything	easily accessible	readable text	very hard to see
cannot see worksheet	more accessible	document reader	too hard to see
cannot see number	great accessibility	easier reading	really hard to see
cannot see status	accessibility suite	can read	terribly hard to see
still cannot see	accessibility screen	cant read	hard to see theme
blind	**header**	**flicker**	**voice command**
blind user	theme header	screen flicker	voice command search
color blind	custom header	flicker taskbar	use voice command
supports blind	size header	flicker background	voice commands works
impaired / blind	adjust header	heavy flickering	simple voice command
totally blind	transparent header	constant flickering	custom voice command
text-to-speech	**screen reader**	**impaired**	**text reflow**
verbose text-to-speech	screen reader accessibility	visually impaired	text reflow feature
text-to-speech works	accessibility screen reader	vision impairment	activate text reflow
text-to-speech feature	talkback screenreader	visual impairment	good text reflow
text-to-speech news	small-screen reader	great for visually impaired	has text reflow
transcript	**visual cue**	**navigable**	**audio description**
transcript title	no visual cue	navigable bar	turns on audio description
recording / transcription	some visual cue	navigable button	
zooming and transcript	provide a visual cue	navigable app	
transcription not found		easily navigation	

TABLE 7.4. List of keywords trending in the 5326 reviews. Keywords in **bold** are found to be strongly correlated to accessibility reviews by our model.

Keywords				
(1) dark mode	(16) adjustable	(31) **voice command**	(46) colour coding	(61) captcha
(2) zoom	(17) **blind**	(32) **text-to-speech**	(47) **transcript**	(62) **audio description**
(3) customization	(18) **header**	(33) eyestrain	(48) default language	(63) container
(4) font size	(19) overlap	(34) strain	(49) older device	(64) distinguishable
(5) volume	(20) pause button	(35) background image	(50) **visual cue**	(65) input type
(6) **cannot see**	(21) **flicker**	(36) **screen reader**	(51) grouped	(66) keyboard language
(7) **accessibility**	(22) spacing	(37) change language	(52) seizures	(67) page refresh
(8) **readable**	(23) migraine	(38) small widget	(53) select language	(68) page title
(9) change font	(24) input method	(39) stop button	(54) understandable	(69) sign language
(10) **hard to see**	(25) autoplay	(40) **impaired**	(55) vibration feedback	(70) svg image
(11) background color	(26) metadata	(41) **text reflow**	(56) actionable	(71) switch device
(12) light mode	(27) too bright	(42) timeout	(57) audio cue	(72) touch target
(13) mute	(28) haptic	(43) consistency	(58) missing label	(73) adjust size
(14) contrast	(29) scaling	(44) epilepsy	(59) **navigable**	(74) adjust colour
(15) subtitle	(30) control key	(45) assistance	(60) verbose	

Summary. The Boosted Decision Trees model, with an accuracy of 90.6% and an F1-measure of 90.7%, is the best performing model in the binary classification of accessibility reviews.

RQ2. How effective is our machine learning approach in identifying accessibility reviews?

The main goal of this study is to propose an automatic approach for identification of accessibility reviews that can effectively outperform current state-of-the-art baselines: Keyword-based (i.e., also called pattern-based or string-matching) [112] and Random classifier [195]. Existing studies that have applied machine learning techniques in similar contexts (i.e., text classification) usually evaluate their approach using different classifiers. To compare their approach against others, they consider the keyword-based approach. To our knowledge, the only study that considers additional approach (i.e., random classifier) is the study by Da Silva et al. [195]. Thus, we consider keyword-based and random classifier to compare against our approach. Answering this question is important to understand if the detection of accessibility reviews is a learning problem. We hypothesize that learning algorithms can outperform string-matching algorithms. To examine if the hypothesis holds true, we chose to investigate the following two baselines, and compared them with our BDTs-model.

Table 7.5. Examples of the misclassification case of our BDTs-model.

Type	Example
False Positive	Simple and easy to use This app works well - especially lucid dream - i still remember my dream last week. Amazing! But i dont like the side effects - like headache and other emotional thing.
False Negative	Beautiful Functioning Reader Thank you for all your hard work in making this app for us to use. And to offer it to us for free is amazing. I use this app everyday, I got all my friends and family using it too. Thank you so *much! I can only think of one thing that could make this app better, if you could add caller ID* with name, and make it so users could turn it on or off, this would be great. Even without that, this app is great.

Baseline 1. Keyword-based Approach. The keyword-based (string-matching) approach for identifying accessibility reviews is suggested by Eler et al. [112]. In their work, they inspected 214,053 user reviews to identify the reviews pertaining to accessibility. Their string-matching approach classified a total of 5,076 reviews as accessibility reviews. However, manual verification of the 5,076 reviews later found that only 2,663 of the reviews were correctly identified [112].

To calculate statistical metrics for baseline 1, we used a set of 5,326 reviews (cf., set of 2,663 accessibility reviews, from Table 7.1, and another 2,663 non-accessibility reviews, selected from the same apps). Then, we manually inspected these reviews to determine true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). True positives are when the keyword-based approach correctly detected accessibility reviews, and true negatives are when non-accessibility reviews are correctly identified.

False positives are the reviews identified as accessibility reviews while they are not; and false negatives are the reviews identified as non-accessibility reviews while they are accessibility reviews. Since we already had the reviews labelled, we were able to count *TP*, *TN*, *FP* and *FN*.

Baseline 2. Random Classifier. Similar to Da Maldonado et al. [195], we consider Random classifier as one of the baselines to compare our approach to. The precision of the random classifier technique is calculated by dividing the number of accessibility reviews by the total number of user reviews (i.e., $\frac{2663}{214053} = 0.012$). When it comes to recall, there is only 50% probability for a review to be classified as an accessibility review since there are two possible classifications available. Finally, the F1-measure of baseline 2 is calculated as $2 * \frac{0.012*0.5}{0.012+0.5} = 0.023$.

Using the values of *TP*, *TN*, *FP* and *FN*, we calculated the Precision, Recall, and F1-measure, for both baselines. Table 7.6 shows the standard statistical measures of the three approaches, also the performance improvements achieved by our BDTs-model compared to the other two methods.

As can be seen from Table 7.6, F1-measure obtained by the machine learning approach is much higher than the other methods. F1-measure achieved by the machine learning approach is 0.90, while F1-measure values using keywords and random classifier are 0.576 and 0.023 respectively. Table 7.6 shows that our approach outperforms the keyword-based approach by 1.574 times and the random classifier by 39.434 times when identifying accessibility reviews. To better understand the performance of the string-matching method, we have extracted examples reviews that were wrongly classified, as accessibility:

Review 1. Good to have your files easily accessible. Would like integration of caldav/ carddav

Review 2. Very useful application. Gmail users must go for it blind eyes

The existence of keywords such as accessible and blind eyes, are string-matched to the keywords considered as accessibility by the guidelines, and so, the keyword-based approach will flag their corresponding reviews as accessibility. However, the first review (i.e., Review 1) refers to the new feature that allows user files to be accessible more efficiently and requests the integration of a protocol for the synchronization of calendars. Similarly, the second review (i.e., Review 2), is praising an app that synchronizes Gmail calendar with Outlook calendar, and the user's expression of "going with blind eyes", refers to their satisfaction, and not to what would be considered by the string-matching method as an accessibility issue.

To determine the different cases of when the keyword-based approach fails, we evaluated 592 reviews, a statistically significant sample with a confidence level of 99% and a confidence interval of 5%. By analyzing the selected reviews, we identified the following reasons behind the failure cases of the string-matching approach:

- **Keyword Misspelling.** This category depicts the case when accessibility aspects of the mobile application are addressed by the users using misspelled keywords. This case can be illustrated in the following example: Font size of lowercase letters is sooosmall! How to change it? It should be like on google keybord when you change capital/lowercase mode - lowercase letters have almost the same size as capital. It's much easier for your eyes!. The keyword matching approach can miss any word with a typo or with inproper spacing, such as keybord or sooosmall. Misspellings are frequent in app user reviews, since mobile writing is known to be more prone to typos.
- **Keyword Variation.** This category shows the case in which users use different part-of-speech (POS) of the accessibility-related keywords reported in Table 7.4. As shown in the following review: very accessible as a blind user thank you, the user used the adjective form (accessible) of the word accessibility.
- **Expression Variation.** This category represents cases in which users use different expressions of the keywords listed in Table 7.4 to address accessibility aspects of the apps. This case is best illustrated in the following accessibility review: still getting responses from the wrong people and noticed that when in night mode with pure

black background - when you try to delete a message the yes option is completely black so impossible to see. As can be seen, the expression impossible to see is used instead of the keyword cannot see to represent the user perspective on the problem.

Summary. The Boosted Decision Trees model outperforms the current state-of-the-art approaches in the classification of accessibility reviews. We obtained an F1-measure score of 90.7% with an improvement of 1.574x and 39.434x over the keyword-based and random classifier approaches respectively.

TABLE 7.6. Comparison in approaches used to the baselines in our study.

	Our approach			**Keyword-based**			**Random classifier**		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Classification	0.898	0.916	0.907	0.996	0.405	0.576	0.012	0.500	0.023
Improvement	–	–	–	0.901 x	2.261 x	1.574 x	74.833 x	1.832 x	39.434 x

RQ3: What is the size of the training dataset needed for the classification to effectively identify accessibility reviews?

So far, we showed that our machine learning approach can accurately identify user reviews that pertain to accessibility. However, the performance of a classifier relies on the size of the training data. At the same time, creating a training dataset is a challenging and time-consuming task. Thus, the question is: What is the size of the training dataset needed to effectively classify user reviews? If an approach requires a very large training dataset than it will require a considerable time and effort to be applied to other similar contexts. However, if less training dataset is required to effectively classify accessibility reviews, then our approach can be applied and extended with little efforts.

To answer this research question, we incrementally added reviews to the training dataset and evaluated the performance of the classification. We began by creating a large training dataset that contains equal size of accessibility reviews and non-accessibility reviews.

Then, we used cross validation technique, which is a technique that partitions the original dataset into a training set to train the model, and a test set to evaluate it using number of folds [164]. In this study, we divided the dataset into 10 folds making sure they contain equal size of both classes. Next, we tested our approach using a 10-fold cross-validation technique using 9 folds for training and 1 fold for testing. Since we wanted to monitor the performance of our classifier as the training dataset size increased, we incrementally added batches of 100 reviews until we used all of our training data (e.g., 5,326 reviews). It is important to note that we considered the equal size of accessibility reviews and non-accessibility reviews with batches incrementally added to the training dataset. We computed the F1-measure value for each iteration (e.g., after adding batches of new reviews to the training set). We recorded the number of reviews needed to achieve at least an F1-measure of 80% to 90%.

Figure 7.4 shows F1-measures calculated when detecting accessibility reviews, while incrementally adding batches of reviews to the training dataset. Our results show that the highest F1-measure (i.e., 0.907) was achieved with 5,326 reviews (our total training dataset) and the lowest F1-measure value (i.e., 0.630) was achieved with 100 reviews. Our results also show that 80 to 90 percent F1-measure is achieved with 400 to 5000 reviews in the training dataset. Such that, we need only 400 reviews to get around 80% F1-measure and we need at least 1500 reviews to get 85% or higher, while with 5000 reviews we got around 90% F1-measure. Finally, we found that the F1-measure score improves as we add to the training dataset.

Summary. We find that we need a relatively smaller training dataset (i.e., 1500 reviews) to get 85% or higher F1-measure. The F1-measure score improves as we add to the training dataset.

7.4. Discussion

We presented a new approach that identifies app reviews with accessibility concerns. We compared our new approach to the current state-of-the-art methods. Based on these findings we discuss implications that can be theory-based and practice-based. Theory-based implications show how this study can further advance the research on accessibility reviews. Practice-based implications show how our model supports our community in building and maintaining accessible mobile apps.

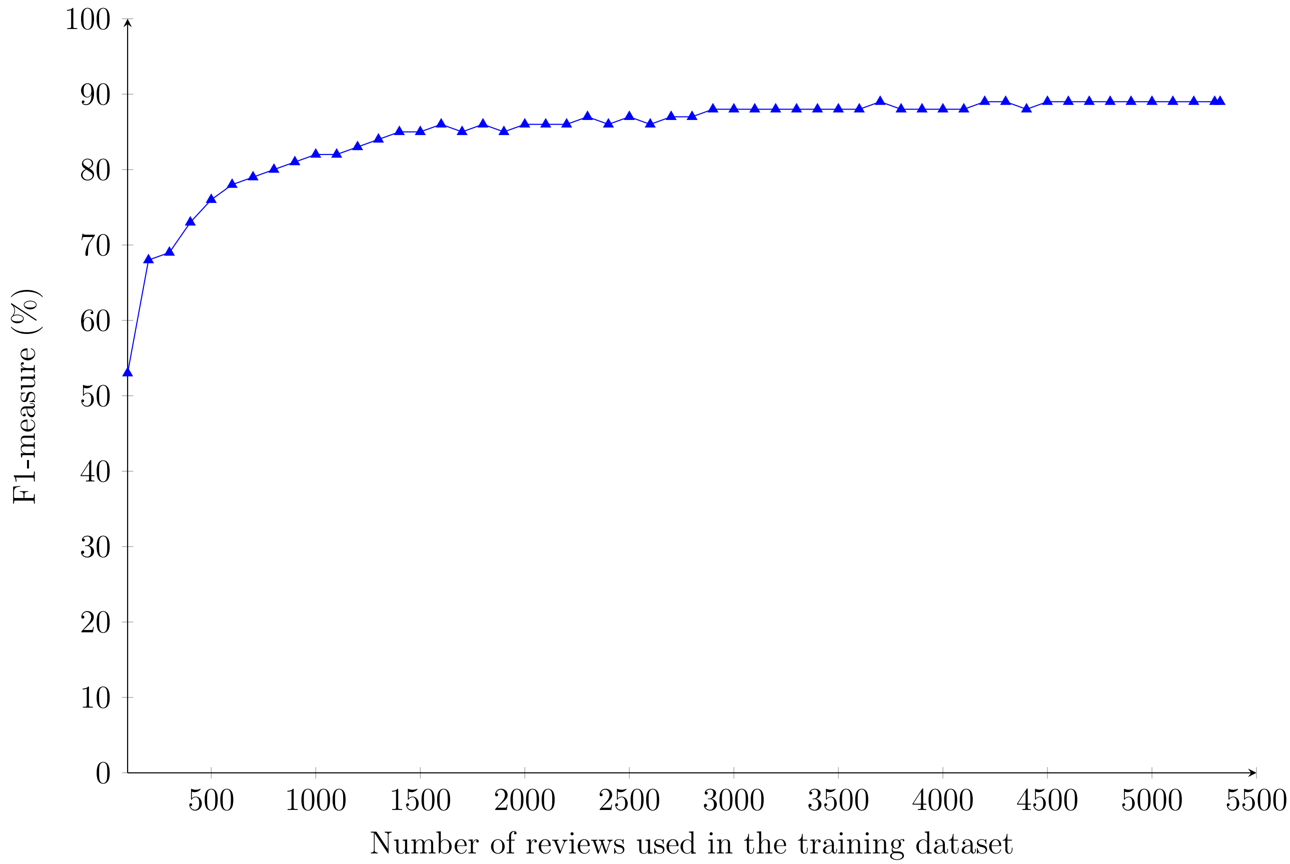

FIGURE 7.4. F1-measure achieved by incrementally adding training data size for binary classification.

Implication 1: App reviews are rich source of information that can be mined to identify specific accessibility problems with the mobile app. There are so many accessibility guidelines that developers and designers can find it difficult to test for all of these guidelines. Additionally, adhering to these guidelines does not necessarily guarantee the accessibility of the said app. Also, usability testing with different groups of people with disabilities, e.g., blind or deaf, can be infeasible especially for medium and small-scale companies. One way to discover accessibility problems which prior testing did not reveal is to listen to the users and learn from the reviews they wrote. Our approach can aid technology professionals to quickly spot accessibility problems with their app.

Implication 2: Accessibility as part of mobile apps maintenance and evolution. There exist accessibility testing tools and methods that are designed to support the implementation and testing phases of the software. However, there are no tools, to the best of our knowledge, that supports software accessibility in the maintenance phase. With changes made to an app, either for adding a feature or fixing a bug, accessibility can be at risk. Also, with updates made to the phone's operating system or the installed assistive technology, the accessibility of an app may deteriorate. We call for innovative methods that can support technology professionals in maintaining the accessibility of their app after its release. Our approach in analyzing app reviews offers an opportunity for developers and designers in detecting accessibility pitfalls based on their users' written feedback. However, with the tremendous number of reviews developers receive on a daily basis, it becomes impractical to manually read through them and identify potential issues related to their new release. Adding our model to the pipeline, will alleviate the manual overhead of looking up accessibility related reviews, and so developers can quickly locate their corresponding issues, and add them to their maintenance pipeline.

Implication 3: Understanding users' language in expressing their accessibility concerns. When we compared our BDTs-model to the keyword-based detector, we found that some accessibility reviews did not contain the accessibility keywords that were driven from accessibility guidelines [112]. This indicates that users voice their accessibility feedback using user taxonomy which may or may not echo the technical and professional terms used in accessibility standards. Further research is needed to understand how users describe mobile accessibility issues. By learning the accessibility user taxonomy, we can improve our BDTs-model, which will lead to enhanced discovery of accessibility reviews.

Implication 4: The interplay between developers and designers, accessibility experts, and users. Accessibility experts establish guidelines and design methods in support of creating accessible software. Technology professionals often are not able to digest all these guidelines and often find existing resources lacking. This situation yielded to the existence of software products that are inaccessible to people with disabilities. The effective involvement of people with disabilities in this process can help bridging the communication gap between accessibility experts and developers and designers. By giving users the opportunity to lead the prioritization of accessibility issues based on their usage experience, mobile apps accessibility can be improved in a more meaningful way for people with disabilities. Analyzing app reviews is one way to give users the lead in determining which accessibility issue should be fixed in the next release. Analyzing app reviews can also offer insights to accessibility experts on users' accessibility needs right from the field, which will be more realistic than results collected from controlled lab studies.

Implication 5: Direct and immediate apps filtering benefit for end users. People find online reviews helpful in making purchase decisions [57]. Peer comments help users become aware of the limitations of reviewed products [217]. Currently, on mobile applications stores, e.g., App Store and Google Play, users can read all reviews, sort them by most helpful or most recent. However, mobile application stores provide no means to filtering reviews based on relevance to specific quality metrics, e.g., accessibility. This lack of filtering pushes users to download the app first and then experience its accessibility, leaving no room for benefiting from peer comments. Sometimes, apps suffer from accessibility regression giving users an unpleasant surprise with an updated app that is less accessible than its former version [299]. We call on mobile application stores to take action and allow users to filter reviews based on relevance to accessibility.

Implication 6: Pushing the boundaries of Accessibility testing. Current accessibility testing strategies are human intensive, and therefore become expensive and impractical, as most developers struggle to find the appropriate testers who can evaluate the compliance of their apps to accessibility guidelines. Existing accessibility scanners are tailored for the web, and they cannot be applied to the mobile environment. In this context, online user reviews, offer a rich source of scenarios, which can be coupled with the app's current version, to create test cases of practically captured anomalies.

7.5. Conclusion

This chapter presents an approach that automates the classification of app reviews as accessibility-related or not so developers can easily detect accessibility issues with their products and improve them to more accessible and inclusive apps utilizing the users' input. As Hayes pointed out:In Action Research, the goal is ultimately to create sustainable change. That is to say, once the research facilitators leave, the community partners should be able to maintain the positive changes that have been made. [139]. Our goal is to create a sustainable change, by including a model in the developer's software maintenance pipeline and raising awareness of existing errors that hinder the accessibility of mobile apps, which is a pressing need [244].

As we develop our model, we conducted an evaluation of nine different classifiers using an existing dataset of manually validated accessibility reviews. Our evaluation shows that the Boosted Decision Tree classifier offers higher accuracy than the other approaches in the classification of app reviews. Additionally, we compared our approach with two baselines, namely a keyword-based approach, and a random classifier. The results indicate that our approach outperforms the two state-of-the-art approaches with the F1-measure of 90.7%. Finally, we conduct an experiment to evaluate the impact of training data sizes on our classifier's accuracy.

7.6. Chapter Summary

: This chapter developed a supervised learning classifier to formulate the identification of accessibility reviews as a binary classification issue. Although automatic identification is convenient, its major disadvantage is that it does not capture words in user reviews that are not in the accessibility guidelines.

In the next chapter, we propose a classifier that automatically classifies user reviews, into what type of accessibility guidelines they are referring to (Principles, Audio/Video, Design, Focus, Forms, Images, Links, Notifications, Dyn.content, Structure, and Text Equivalent). This will allow developers to easily identify accessibility-related issues.

CHAPTER 8

AUTOMATIC CLASSIFICATION OF ACCESSIBILITY USER REVIEWS IN ANDROID APPS

8.1. Introduction

Many users find it challenging to get complete benefits from mobile applications (apps) having poor accessibility [319, 48, 257, 255, 4]. To address this challenge, researchers have offered a variety of methodologies, techniques, frameworks, tools, and guidelines to guide the development of building mobile applications with better accessibility[242]. It is unfortunate that due to a lack of understanding or resources (e.g., funding and time), many mobile application developers and designers continue to neglect to include accessibility in their mobile app development process [244]. In this chapter, we seek to develop a multi-class method that can help app developers to distinguish between the type of accessibility user reviews easily. The proposed method will automatically classify user reviews derived from app stores, such as Google Play[1], and Apple Appstore[2], into the accessibility type based on the accessibility guideline [62, 28].

There are many challenges with detecting accessibility related to user reviews. One of the most common ways of improving applications is by analyzing the feedback given by the users [91, 23, 181, 31]. In many cases, accessibility user reviews for mobile applications are overlooked [113]. It is vital to mention that accessibility user reviews can either be detected automatically or manually [113]. Given a large number of app users' reviews, manual identification becomes more tedious and time-consuming, meaning that automatic identification is often preferred. Automatic identification of reviews means that the system looks for certain keywords in the user reviews that relate to accessibility [113]. The British Broadcasting

[1]https://play.google.com/store

[2]https://www.apple.com/ios/app-store/

This entire chapter is reproduced from Aljedaani, Wajdi, Mohamed Wiem Mkaouer, Stephanie Ludi, and Yasir Javed, "Automatic classification of accessibility user reviews in Android apps," in 2022 7th International Conference on Data Science and Machine Learning Applications (CDMA), pp. 133-138. IEEE, 2022, https://ieeexplore.ieee.org/abstract/document/9736367, with permission from IEEE.

Network (BBC) accessibility guidelines provide the keywords used for automatic identification [62]. Although automatic identification is convenient, its major disadvantage is that it does not capture words in user reviews that are not in the accessibility guidelines. In addition, even when the keywords are in a certain user review, it is not a guarantee that the review concerns accessibility. In past studies [113], researchers found phrases in user reviews with keywords from the guidelines, which were not necessarily about accessibility. Hence, researchers should be careful to consider the context of the review so that identification will be effective. Such a challenge can be overcome by introducing learning capabilities, which are trained to know the difference between accessible user reviews and those that are not, even if they seem similar. Furthermore, not all accessibility problems uniformly occur, and therefore, some accessibility violations tend to be more frequent than others. This can potentially be another challenge for any automated solution that tries to identify them since one category will be more popular (better represented by data) than another.

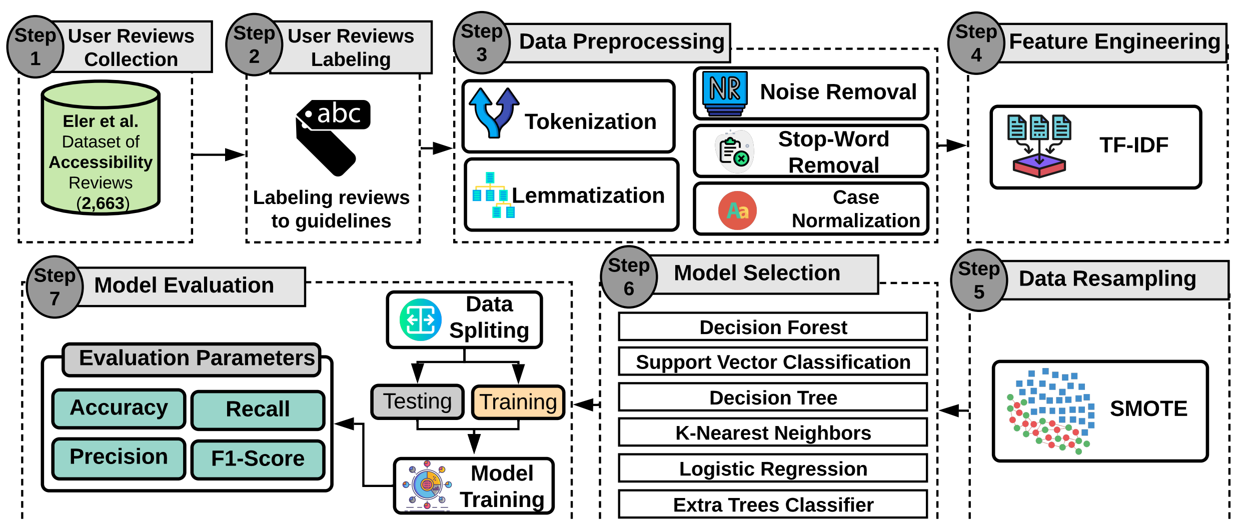

FIGURE 8.1. Overview approach of our study.

To address the above-mentioned challenges, the goal of this paper is to help developers automatically classify user reviews, into what type of accessibility guidelines they are referring to (Principles, Audio/Video, Design, Focus, Forms, Images, Links, Notifications, Dyn.content, Structure, and Text Equivalent). This will help developers quickly distinguish accessibility-related problems, and address them in a timely manner.

To design our solution, we rely on supervised learning techniques to effectively enable app developers to correctly identify the underlying accessibility problem in the user reviews. We analyze a corpus of user reviews, extracted from open-source apps [113] to extract accessibility problems. Since our goal is to design a model that distinguished between types of accessibility issues, we manually categorized the user reviews based on accessibility guidelines [62]. Then, we employ emerging machine learning techniques, known to perform best in text classification [166], to know the features of the reviews that can help in their identification. Our proposed approach gets acquainted with the keywords and patterns that are unique to a given type of accessibility guideline, and transforms them into features to identify a given class (type of accessibility guideline). Determining unique features is crucial for classification algorithms. The outcome of the algorithm (labeled review) is important for app developers to understand accessibility issues and improve on them.

The following are the key contributions of our research:

- We tackle the identification of accessibility user reviews as a multi-class classification problem, where we analyze the extent to which, machine learning models can accurately distinguish between types of accessibility reviews.
- To handle the unbalance between the number of reviews belonging to each category, we adopt the state-of-the-art re-sampling technique SMOTE. This chapter also showcases the potential of class re-balancing on supporting the representation if minority classes (i.e., categories with fewer reviews).

8.2. Study Design

This section describes the proposed method to classify the type of accessibility in review on the selected dataset. Figure 9.1 presents an overview approach of our study.

8.2.1. Step (1): User Reviews Collection

In this study, we used a corpus of user reviews, collected from various popular open-source projects [113]. These reviews are collected from thousands of users from all over the globe. The dataset contains 2,663 reviews that have been manually inspected for containing a problem related to accessibility. The reviews were gathered from 701 Android applications, belonging to 15 different categories, as shown in Table 9.1.

TABLE 8.1. Summary of accessibility guidelines with corresponding description, relevant keywords, and the number of labeled reviews. We followed the BBC standards and guidelines for mobile accessibility [62].

Guideline	Description	Relevant Keywords	# of Labelled Reviews
Principles	These guidelines require a focus on three principles of developing usable and inclusive applications. First, developers should utilize all web standards as required. Secondly, there should be the utilization of interact controls. Thirdly, content and functionality in the app should support native features of the app.	Accessibility, disability, operable, screen reader, blind talkback, , impaired, impairment	664
Audio/video	Applications should provide alternative formats such as transcripts, sign language, or subtitles. Autoplay should be disabled, and the user should be provided with play/pause/stop or mute buttons to control audio. There should be no conflict between audio in application media of native assistive technology.	Subtitle, sign language, transcript, audio description, autoplay, mute, volume, can't hear	311
Design	The color in the app background should have appropriate contrast, and touch targets must be large enough to be touched effectively. Visible state change should be experienced in every item in the app that has been focused on. Unnecessary or frequent flickering of content must be avoided.	Contrast, background color, flicker, font size, visual cue, dark/light mode, eyestrain, seizure, can't see, overlap	1,328
Focus	There should be a logical organization of items, and users should be offered alternative input methods. Interactive and inactive elements should be focusable and non-focusable, respectively. Keyboard traps should be eliminated, and focus should not change suddenly when the app is utilized.	Focusable, control focus, focus, keyboard trap, navigable, order, input/type	122
Forms	Every form of control must have a label. All labels must have a logical grouping, and a default input format must be given. Labels should be close to their form controls.	Unique label, missing label, layout, voice-over, visible label	53
Images	Text images should not be included. Any background images that have content should have another accessible alternative.	Image of text, hidden text, background image, text alternative	86
Links	Any navigation links must indicate the function of the link. If a link to an alternative format is clicked, the user should be notified of the redirection to the alternative. Several links that redirect to the same sourceshould be put together in one link.	Link description, unique desc., duplicate link, alternative format	35
Notifications	Error messages should be clear. Any notifications given must be easily seen or heard. There should be standard system notifications where necessary.	Operating inclusive, vibration, feedback, alert dialog, understandable, unfamiliar	49
Dyn. content	Applications should be made in a progressive manner that enables every user to benefit from them. Appropriate notifications should be given for automatic page refreshes. Flexible interaction input control must be given.	Animated content, page refresh, automatic, refresh, timeout, adaptable, input sign	15
Structure	Every page on the application should be uniquely identified. Content should be arranged in a hierarchical and logical manner with appropriate headings. One accessible component should be used to group interface objects, controls or elements.	Page title, screen title, heading, header, unique descriptive	0
Text equivalent	Applications should give the objective of a specific image or its editorial aim. Also, visual formatting must be complemented by other ways to give meaning. There should be no conflict between decorative images with assistive technology. Every element must have well-placed and effective a11y properties.	Alternative text, non-visual, content description, decorative content, no-text-content	0

8.2.2. Step (2): User Reviews Labeling

We need to categorize the dataset based on the mobile accessibility guideline [62]. To do so, two authors performed the manual categorization of all user reviews in the dataset. The process of categorizing was spread across seven days to prevent human fatigue. The authors were also provided with the chance to search online for unfamiliar keywords during labeling. After the authors finished the manual categorizing procedure, the dataset was validated using the process of Levin et al. [178] by randomly choosing a 9% sample of accessibility reviews. The sample size was 243 out of the 2,663 accessibility reviews. This number is about equivalent to the size of the sample based on a 95% confidence level with a 6 confidence interval. Following that, the third author categorized them. The chosen reviews were not exposed to the author before. Then, the categorical reviews were evaluated using Cohen's Kappa coefficient [94] with respect to inter-rater agreement level, and the "0.87" agreement level was attained. As per Fleiss et al. [121], (i.e., 0.87–1.00) are perfect values for agreement, and our agreement values are considered nearly *perfect agreement*. Table 8.1 presents the accessibility guidelines and the distribution of the categorized user reviews per guideline.

TABLE 8.2. Statistics of the dataset.

Number of Apps	701
App Categories	15
All Reviews	214,053
Accessibility Reviews	2,663

8.2.3. Step (3): Data Preprocessing

We used a textual preprocessing strategy after finishing the process of collecting data. It is vital to preprocess and clean the document adequately so a model can execute text categorization successfully. In our method, we combined NLP methods employing the NLTK python (Natural Language Toolkit) to preprocess the reviews of the app. Among the techniques based on NLP are:

- **Tokenization:** This technique involves splitting natural texts into tokens without any white space throughout this procedure. Tokenizing app reviews involves breaking them down into constituent words set.

- **Lemmatization:** Throughout this procedure, a word's suffix is replaced or removed so its basic form can be obtained. It also lowers the unique occurrence's' count of words that are similar. This approach is used in the suggested strategy for word pre-processing in their canonical format in order to limit the unique occurrences count of identical text tokens.
- **Stop-Word Exclusion:** Stop-words are words that do not assist to the process of classification, such as the, am, and so on.
- **Case Normalization:** Because precise words having various font cases must be treated in a similar way, such as "accessibility" & "Accessibility," the entire text must be converted to lowercase letters. It is commonly referred to as data cleansing because it aids in minimizing the repetition of similar features that vary only in regards to case sensitivity. A person might identify himself as "Deaf" using a capital letter 'D' with in setting of reviews related to accessibility to convey his cultural background within reviews. Because we have multi-class classifier, the classification outcome for "deaf" and "Deaf would be identical, and case normalization would be secure, there would be no overruling of expressions by the user.
- **Noise Removal:** This stage removes any noise that could degrade the performance of the classification or cause the model to become confused during learning. Special characters, numeric data, email id, are examples of noise types deleted in this phase.

8.2.4. Step (4): Feature Engineering

Feature engineering helps the model learn patterns for each class it is trying to distinguish, through allocating appropriate unique key words that appear specifically for one category. We intend to train our model and find these unique keywords and use them as features to properly distinguish between classes. The following is feature engineering method that was employed in this study:

TF-IDF is among the most commonly employed scoring metrics for summarization and information retrieval. It is utilized to convey the significance of the term within each text. The TF-IDF extraction function takes two inputs: IDF and TF. TF-IDF provides tokens that seem to be uncommon within the dataset. When uncommon words appear in multiple documents, their relevance grows.

$$tfidf_{t,d,D} = tf_{t,d} \cdot idf_{t,D} \tag{8.1}$$

where $\boldsymbol{t}$ denotes terms, $\boldsymbol{d}$ denotes each document, and $\boldsymbol{D}$ is the documents set. The parameter, n-gram range, is used in conjunction with TF-IDF. TF-IDF is used to compute word weights, which offer corpus weights for any given the word. The weighted word matrix is the output. Using TF-IDF vectorizer, an increase within meaning is proportionate of the count, although word frequency in the corpus assists in controlling it. The TF approach is frequently explored for extracting features and therefore is widely utilized for text categorization. During classifier training, the incidence frequency of terms' is used as a parameter. TF function doesn't quite take into account the popularity of a word, contrasting the TF-IDF, which gives less weight to more frequent terms.

8.2.5. Step (5): Data Re-Sampling

Imbalanced dataset issues and problems can be resolved through methods of data re-sampling. The problem that can arise in an imbalanced dataset is that it has an uneven ratio of the target classes, which results in the models over-fitting on the majority class during classification [231]. For this, the technique and strategy for re-sampling the dataset have been proposed. Throughout this study, the technique on re-sampling that has been utilized is over-sampling.

Synthetic Minority Over-Sampling Technique (SMOTE) In over-sampling, the sample of the class that is a minority increased in the ratio of the majority class. This enlarges the size of the dataset and provides more features that can train the model and improve its accuracy. Over-sampling is implemented in this research using SMOTE, known as the synthetic minority over-sampling technique. SMOTE is a modern approach and was presented in [87] to figure out how over-fitting in the imbalanced dataset could be overcome. It selects the smaller class at random and locates the K-nearest neighbors for each of these classes. The K-nearest neighbor is used to analyze the samples that are chosen to create a new minority class. Figure 8.2 shows the distribution of the accessibility reviews before and after using SMOTE.

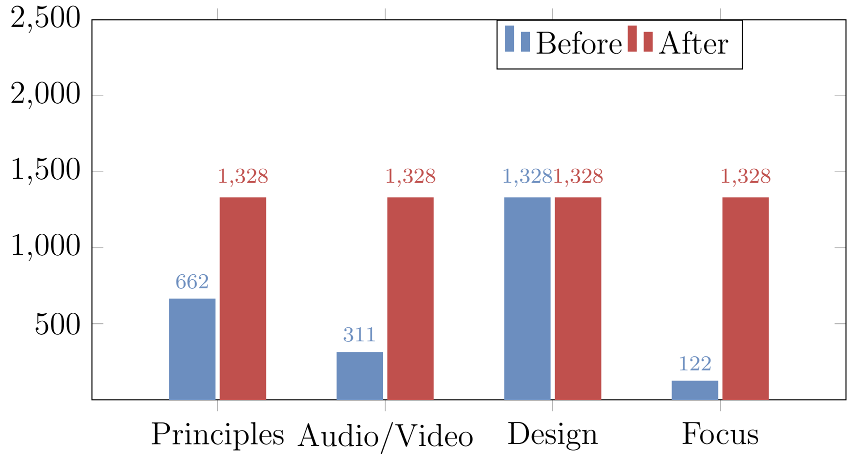

FIGURE 8.2. Distribution of reviews before and after SMOTE.

8.2.6. Step (6): Model Selection

In order to build our model, we used the following six learning algorithms:

- **Random Forest (RF):** is a tree-based classifier that constructs a large number of classification trees. Each tree gives a distinct classification [52]. RF selects the classification with the most votes out of all possible trees to classify a new item.
- **Support Vector Classification (SVC):** is a well-known machine learning classifier for tackling linear & non-linear issues. It's suitable for a variety of practical applications [264]. SVC generates a hyperplane or line that splits the data in the section. Higher-dimensional space is obtained by transforming low dimensional input space using the Kernel function; this process changes non-separable problems into separable problems. It primarily aids in the resolving of non-linear differential issues. SVC divides the data into categories according to labels.
- **Decision Tree (DT):** learns basic decision rules to forecast the class. DT utilized node and leaf by descending from the root and utilizing the sum of product representation [77].

TABLE 8.3. Detailed classification metrics (Accuracy, Precision, Recall, and F1-Score) of each classifier with TF-IDF feature.

Random Forest (RF)				Support Vector Classification (SVC)				Decision Tree (DT)			
Category	**Precision**	**Recall**	**F1**	**Category**	**Precision**	**Recall**	**F1**	**Category**	**Precision**	**Recall**	**F1**
Principle	0.89	0.88	0.89	Principle	0.88	0.86	0.87	Principle	0.90	0.69	0.78
Audio/Video	0.94	0.97	0.96	Audio/Video	**0.95**	**0.98**	**0.96**	Audio/Video	**0.94**	**0.91**	**0.92**
Design	0.87	0.84	0.85	Design	0.85	0.82	0.84	Design	0.63	0.85	0.72
Focus	**0.97**	**0.98**	**0.97**	Focus	**0.95**	**0.98**	**0.96**	Focus	0.89	0.83	0.86
Average F1	0.92	0.92	0.92	Average F1	0.91	0.91	0.91	Average F1	0.84	0.82	0.82

K-Nearest Neighbors (KNN)				Logistic Regression (LR)				Extra Tree Classifier (ETC)			
Category	**Precision**	**Recall**	**F1**	**Category**	**Precision**	**Recall**	**F1**	**Category**	**Precision**	**Recall**	**F1**
Principle	0.50	0.99	0.66	Principle	0.88	0.86	0.87	Principle	0.92	0.89	0.90
Audio/Video	0.98	0.92	0.95	Audio/Video	0.94	**0.98**	**0.96**	Audio/Video	0.94	1.00	0.97
Design	**1.00**	0.03	0.05	Design	0.84	0.82	0.83	Design	0.89	0.85	0.87
Focus	0.97	**0.97**	**0.97**	Focus	**0.95**	0.97	**0.96**	Focus	**0.97**	**0.99**	**0.98**
Average F1	0.86	0.73	0.66	Average F1	0.90	0.91	0.90	Average F1	0.93	0.93	0.93

- **K-Nearest Neighbors (KNN):** Identifies the similarities between new and existing samples and assigns the new data to a group with a high degree of similarity [104]. The similarity between both the new data and the existing classifications is determined by computing the distance between the two sets of data.
- **Logistic Regression (LR):** is a statistical model which is similar to linear regression and based on the probability concept. By fitting data to something like a logistic function, LR predicts the likelihood of the events.
- **Extra Tree Classifier (ETC):** is a collection of classification algorithms teaching method in which the results of numerous de-correlated random forests gathered within a ”forest” gets merged to provide identifications' outcomes.

We chose these algorithms because they are commonly used for a variety of classification issues [234], and they are able to operate effectively with imbalance datasets and NLP in the literature [125, 167].

Table 8.4. Summary of performance measures, formulas, and definitions.

Measures	Formula	Definition
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	Calculates the closeness of a measured value to the standard value.
Recall	$\frac{TP}{TP+FN}$	Calculates the exact number of positive predictions that are actually observed in the actual class.
Precision	$\frac{TP}{TP+FP}$	Calculates the exact no. of correct predictions out of all the input sample.
F1-score	$\frac{2 \cdot P \cdot R}{P+R}$	Calculates the accuracy from the precision and recall.

8.2.7. Step (7): Model Evaluation

The four assessment aspects indicated in Table 9.4 below are used to evaluate the performance of our chosen models:

- **True Positive (TP):** Parameter determines positive predictions identified accurately using the classifier.
- **True Negative (TN):** Parameter determines whether or not the classifier accurately labels negative predictions.
- **False Positive (FP):** Parameter determines the quantity of negative cases incorrectly assumed to be positive via the classifier.
- **False Negative (FN):** Parameter determines the quantity of positive instances that the classifier incorrectly interprets as the negative instances.

8.3. Study Results

RQ_1: To what extent can machine learning models accurately distinguish different types of accessibility reviews?

In this study, we have used six models, namely, Extra Tree Classifier (ETC), Random Forest (RF), Support Vector Classification (SVC), Decision Tree (DT), K-Nearest Neighbors (KNN), and Logistic Regression (LR) for automated classification of accessibility app reviews in four different categories. These categories include Principle, Audi/Video, Design, and

Focus. Four metrics, i.e., Accuracy, Precision, Recall, and F1-Score, along with TF-IDF features, are employed for each classifier. The results of detailed classification metrics of each classifier with TF-IDF features are presented in Table 8.3.

In the case of RF and ETC, the Focus category achieves the highest recall, accuracy, and F1-Score. In the case of RF and ETC, the highest recall, precision, and F1-Score are obtained in the Focus category. SVC classifier exhibits the same trend in Audio/Video and Focus category while DT classifier performs the best result in Audi/Video category. KNN classifier outputs the highest precision in the Design category, while the Focus category produces the highest recall and F1-score. Lastly, LR gives the highest precision in the Focus category, recall in Audio/Video, and the same F1-score in the Audio/Video and Focus category. Overall, the highest precision (1.00) is achieved by KNN in the Design category, and the highest recall (1.00) in the Audio/Video category and F1-Score (0.98) in the Focus category is achieved by ETC. The ETC classifier obtained the highest average F1-score (0.93), while KNN showed the lowest average F1-score (0.66). It can be seen that classifiers have performed well in the Audio/Video, Design, and Focus categories, whereas the lowest results are obtained in the Principle category.

8.4. Discussion

We chose the four categories/guidelines (principles, audio/video, design, and focus) because they are well-represented, and we know that choosing the others would result in under-representation and an inability to categorize them accurately. At the same time, this is acceptable since the four chosen categories are popular categories that most accessibility user reviews will fit. As a result, the fact that we did not categorize all categories is a limitation. However, we believe that we have captured the primary categories of interest to developers. From the findings of this research, this section presents a discussion of the study takeaways.

Takeaway 1- App reviews represent a valuable source of information which once gathered can give detailed problems related to the accessibility of the mobile app: Accessibility guidelines are numerous, and mobile app designers and developers are in

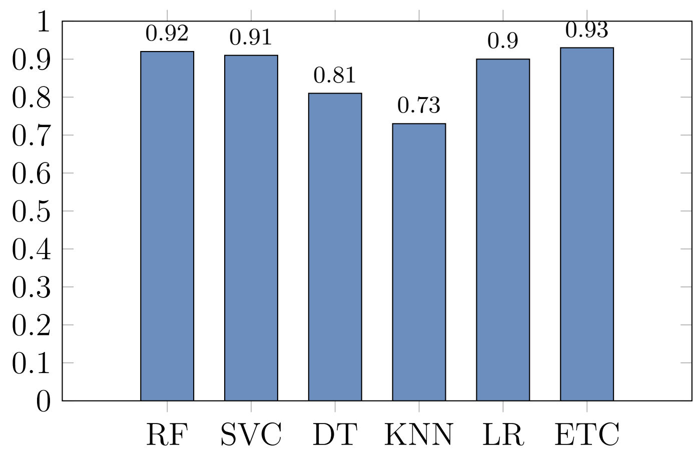

FIGURE 8.3. Comparison of accuracy of all models.

shortage of tools to prevent their appearance. In addition, observing all these guidelines does not always warrant accessibility to the app. Moreover, it is often impractical to undertake usability testing with users with disabilities, e.g., deaf or blind users. A key gap that is not addressed by existing research and testing approaches is listening to user reviews and evaluating them. This new approach is valuable and practical as it allows developers to identify accessibility problems with the app in question.

Takeaway 2- Improving accessibility testing: The current approaches and strategies for accessibility testing are mainly manual. As a result, developers spend a considerable amount of time and cost in identifying the most appropriate people to test their apps on how well they adhere to accessibility guidelines. Accessibility scanners are already available in the market. However, they are only fit for the web and not the mobile environment. In light of this challenge, online user reviews offer new possibilities in capturing anomalies with the app from a more practical perspective. Using the reviews enables developers to identify test cases they wish to undertake in case of app upgrades. In addition, given the dynamic nature of the mobile environment, recent user reviews can indicate any new anomalies in the recently released apps.

8.5. Conclusion

This chapter presented an automated approach for classifying accessibility app reviews in four categories, i.e., Principles, Audio/Video, Design, and Focus, for helping the developers detect app issues and performance improvement by considering user reviews. An existing dataset that comprises manually validated accessibility app reviews has been employed in our work. We employed six classification models, namely Extra Tree Classifier, Random Forest, Support Vector Classification, Decision Tree, K-Nearest Neighbors, and Logistic Regression. To evaluate their performance, we used four classification metrics, i.e., Accuracy, Precision, Recall, and F1-Score for measuring their performance. Evaluation results have shown that KNN exhibits the least accuracy while the ETC model outperformed other models in overall accuracy with TF-IDF features. In the future, we intend to increase the keywords and sample size to improve the selection and analysis process of accessibility reviews and provide a mechanism to check whether the developers have addressed the users' concerns in the subsequent releases by implementing the required features.

8.6. Chapter Summary

: This chapter proposed a classifier that automatically classifies user reviews into what type of accessibility guidelines they are referring to (Principles, Audio/Video, Design, Focus, Forms, Images, Links, Notifications, Dyn.content, Structure, and Text Equivalent). The significance of mobile app development raises several challenges for a deeper understanding of user reviews that are focused on accessibility concerns.

In the next chapter, we propose an approach based on machine learning and NLP methods for Emotion detection automatically tag user reviews as positive, negative, or neutral. The supervised learning method is employed for the annotation of the corpus.

CHAPTER 9

LEARNING SENTIMENT ANALYSIS FOR ACCESSIBILITY USER REVIEWS

9.1. Introduction

Web and mobile applications are common means of engaging with information and services. It is also crucial for these technologies to be accessible to have equal access to people with different abilities. However, in most mobile applications, there is little attention given to accessibility which results into several difficulties to appropriately utilize such applications by people with disabilities [48, 40, 35, 254]. Software application stores like Google Play, App Store and Amazon are available for searching and downloading mobile apps. Most of these platforms freely provide features for user reviews where users can write a review and/or give a star rating. Users' experience provides a valuable knowledge and can be studied by developers, designers, and analysts for the identification of issues in the applications with the help of user reviews [192, 92]. User reviews can be related to requests for features, troubleshooting, compliments, complaints, and dissatisfaction. Reviews can be categorized at higher levels, e.g., how good or bad a feature is. In addition, this division of the high level of app reviews can also be related to the design and usability aspects. Reviews comment on accessibility as well, i.e., the accessibility of apps to the disabled people on their mobile devices [311, 33].

While several studies have addressed various problems related to user reviews [192, 272, 310, 239, 246, 182], user reviews related to accessibility in mobile applications are under-studied [113]. Although the growth of mobile app development is substantial, there is still a lack of mobile-based accessibility-related research, and associated guidelines as compared to web-based accessibility [247]. The significance of mobile app development raises several challenges for a deeper understanding of user reviews that are focused on

---

This entire chapter is reproduced from Aljedaani, Wajdi, Furqan Rustam, Stephanie Ludi, Ali Ouni, and Mohamed Wiem Mkaouer, "Learning sentiment analysis for accessibility user reviews," in 2021 36th IEEE/ACM International Conference on Automated Software Engineering Workshops (ASEW), pp. 239-246, with permission from IEEE.

accessibility concerns. Multiple challenges are associated with the study of user reviews related to accessibility. Prior work has analyzed an enormous amount of reviews and analysts with little impact on the field. For example, there is the possibility of bias in manually identifying the accessibility reviews.

The Internet has become an effective tool through which people communicate their feelings, emotions, and ideas [124]. Business analysts use this data for monitoring people's perceptions and opinions about their products. Natural Language Processing (NLP) based methods have been widely used for the automatic detection of data contents from the text [90]. Artificial Intelligence (AI) based approaches have gained prominence for the development of sentiments or emotion-based systems [198]. In state-of-the-art Sentiment Analysis techniques, the issue is that they access the response in the context of positive or negative aspects but not the specific feelings of the customer and the intensity of their response. To deal with these issues, we present a sentiment analysis based method for identifying accessibility-related problems in mobile apps. The proposed approach is based on machine learning and NLP methods for Emotion detection and automatically tags user reviews as positive, negative, or neutral. A supervised learning method is employed for the annotation of the corpus. After performing annotation of the corpus, the labeled corpus is fed into the model for detecting emotions/sentiments from the user reviews. The proposed system comprises different stages, including data pre-processing, extraction of features, and the prediction model.

Following are the key contributions of our research:

- Use of sentiment analysis for tagging the accessibility-related user reviews.
- Improvement of prediction accuracy of user-reviews tagging.
- We perform a comparison of TextBlob and VADER sentiment analyzers.
- A replication package of the dataset for extension purposes [1].

9.2. Study Design

The main goal of our study is to automatically identify user reviews related to accessibility from the application reviews dataset. Reviews are provided as an input to our proposed approach, and then it performs sentiment analysis on the reviews, i.e., whether the review is positive, neutral, or negative. For this purpose, we generate the classification features using bag of words, extracted using TF-IDF, similarly to previous studies processing user reviews [190, 138]. We build our classification model using corpus reviews and current classification techniques. We then utilized the classification model to predict the types of new app reviews. The overview of the whole process is depicted in Figure 9.1. The key steps of our proposed approach are as follows:

Step (1) - Data Collection: For training, the dataset including the app reviews and their categories are identified through manual inspection [113].

Step (2) - Data Preprocessing: To improve the reviews of the proposed learning algorithms, data cleansing and pre-processing techniques, i.e., tokenizing, lemmatizing, stop words removal, and capitalization removal, are utilized [31, 25].

Step (3) - Sentiment Analysis: To tag the user reviews, we used two sentiment analyzers TextBlob [189] and VADER [147].

Step (4) - Feature Engineering: To create a structured feature space, TF-IDF and BoW [326] techniques are used on preprocessed review text.

Step (5) - Model Selection: We used six classification models for performance evaluation of the proposed prediction model, *i.e.*, LR, SVC, ETC, GNB, GBM, and ADA. The algorithms that are most commonly used for text classification were selected [218, 151]. The performance of the model is validated after training and evaluating the model. We have followed the approach provided by Kowsari et al. [166] which discusses state-of-the-art techniques and algorithms similar to [40] since the app reviews are in plain text.

Step (6) - Model Evaluation: We evaluated the performance of our selected models based on four parameters: accuracy, precision, recall, and F-score [117, 322].

9.2.1. Step 1: Data Collection

In our approach we use a dataset that contains 2,663 manually verified reviews related to accessibility by Eler et al. [113], as shown in Table 9.1. The collected reviews have been extracted from 701 applications which fall under 15 different categories. Eler et al. [113]

first collected 214,053 app reviews, then used 213 keywords for string matching and filtering down the reviews and kept only those reviews that contain accessibility-related information. 54 the British Broadcasting Corporation (BBC) recommendations [63] for accessibility are used for derivation of keywords. After this step, 5,076 potential accessibility reviews were selected after performing string matching.

Table 9.1. Statistics of the dataset.

Number of Apps	701
App Categories	15
All Reviews	214,053
Accessibility Reviews	2,663

Manual inspection showed that 2,663 are true positive. The process of Levin et al. [178] is followed for verification of previous manually labeled reviews, and 243 out of 2,663, *i.e.*, we randomly selected 9% of sample reviews. This value is approximately equal to the sample size by 95% confidence level and 6 confidence interval. Then, 243 non-accessibility reviews were randomly added, and we had 486 total reviews. After that, their labeling was done by another researcher, and the data were kept confidential before. To avoid fatigue, 7 days were given to the review process, and the researcher was given the opportunity to perform online searching of keywords that they were unaware of during the labeling process. After completing the data labeling process, we validated them against the originally labeled reviews. Cohen's Kappa coefficient [94] was utilized to evaluate the categorical classes in terms of inter-rater agreement level, and an agreement level of "0.82" was achieved. The perfect agreement values are 0.81˘1.00 , and our agreement values are considered to have an almost perfect agreement according to Fleiss et al. [121]. The highest number of documents used in the related studies [178, 177] was approximately 2000. In contrast to the existing studies, we have selected 5,663 model creation and validation reviews as our aim was to provide sufficient reviews to the model that could signify all potential accessibility topics.

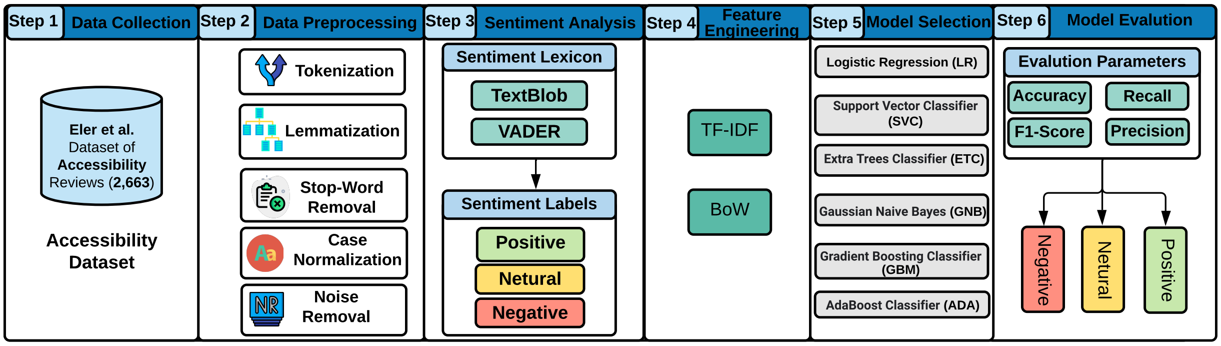

FIGURE 9.1. Overview Approach of Our Study.

9.2.2. Step 2: Data Preprocessing

After completing the data collection process, we selected a text pre-processing approach from [166], which is similar to [40]. In order to perform text classification accurately by a model, it is necessary to clean and pre-process the document properly. For pre-processing the app reviews, NLP techniques using the Python natural language toolkit [308] have been used in our approach. These NLP based techniques include:

- **Tokenization:** In this process, the natural text is split into tokens that do not contain white space. The app reviews are tokenized by splitting them into a constituent set of words.
- **Lemmatization:** In this process, the suffix of a word is removed or replaced in order to get its basic form. It also reduces the count of unique occurrences of similar words. In the proposed approach, this technique is employed to pre-process the words in their canonical form to reduce the count of unique occurrences of similar text tokens.
- **Stop-Word Removal:** Words that do not contribute to the classification process, e.g., am, the, etc., are removed.
- **Case Normalization:** As the exact words with different font cases need to be treated similarly, e.g., Accessibility" and "accessibility", it is required to convert the whole text in lower case. It is generally known as a type of data cleansing which helps in avoiding repetition of the same features that differ only in terms of case

sensitivity. In the context of accessibility-related reviews, a user can identify itself as "Deaf" with upper case 'D' for expressing his cultural identity in the reviews. Our classifier is binary; therefore, it will produce the same classification result for "Deaf" and "deaf" and the case normalization will be safe, and no overruling of users' expressions will be done.

- **Noise Removal:** Any noise that can deteriorate the classification performance and create confusion for the model while learning is removed in this step. Noise types that are removed in this step include numeric data, email id, special characters.

9.2.3. Step 3: Sentiment Analysis

We selected TextBlob and VADER tools in our study for the analysis. We used TextBlob because it is a higher accurate tool for sentiment analysis than other tools [262]. In comparison with the TextBlob, we used the VADER, a most use-able tool for sentiment analysis in recent studies [73]. We used both sentiment analysis tools for fair comparison analysis with multiple techniques.

TextBlob is a widely used lexicon-based method[1] that performs different tasks related to natural language processing (NLP) on raw test [189]. TextBlob algorithm is implemented with a Python library named TextBlob that works as a programming interface for processing text data. With the help of TextBlob, different tasks can be performed, e.g., analysis of sentiments in text, creation of POS tags, extraction of noun phrases, etc. [308]. There are several built-in functions in TextBlob that help in performing different language processing tasks. TextBlob can work in different languages, e.g., Spanish, English, etc. According to research, [219], TextBlob helps in sentiment analysis of tweet data with positive, negative, or neutral polarity. TextBlob library works on top of Natural Language Toolkit NLTK, and its algorithm for sentiment analysis works together with NLTK and pattern processing [170]. Its dictionary includes around 2918 lexicons. In TextBlob, polarity calculation is done on two bases, i.e., objectivity (facts) or subjectivity (personal opinions). The

[1]https://github.com/sloria/TextBlob

sentiment analyzer returns a sentiment score that comprises polarity and subjectivity score. The sentiment scoring range of TextBlob is shown below:

Table 9.2. TextBlob sentiment score range

Negative	Polarity score < 0
Neutral	Polarity score $= 0$
Positive	Polarity score > 0

For subjectivity, the facts-based sentiments have scores below 0.0, while the personal opinion-based sentiments have scores above 1.0.

VADER (Valence Aware Dictionary for Sentiment Reasoning) is a lexicon-based approach[2] that works on gold standard heuristics. It has English language-based sentiment lexicons and is scored and validated by a human. To improve the performance of sentiment analyzer, these lexicons utilize qualitative methods. KirliA et al. [161] suggested that scoring done by VADER sentiment analyzer and human raters hold equal results. Multiple datasets are combined in VADER's corpus. Compared to the previous corpus that focuses on sentiment polarity, VADER also includes the intensity of the polarity score. Slang words and abbreviations that collectively make more than 7500 lexicons are present in its corpus. Scores range between -4.0 to +4.0. The score below -4 specifies the sentiment as negative, whereas the score above +4 indicates the sentiment as negative. Vader's output is depicted in different terms, i.e., neg, neu, pos, compound. Compound output is the aggregation of lexicon scores of complete sentences or a text and ranges from -1.0 to +1.0. The sentiment scoring range of VADER is shown below:

Table 9.3. VADER sentiment score range

Negative	compound score $<=$ -0.05
Neutral	compound score $>$ -0.05 to compound score $<$ 0.05
Positive	compound score $>=$ 0.05

To represent the sentiment intensity and polarity, the VADER algorithm includes a

[2] https://github.com/cjhutto/vaderSentiment

sentiment lexicon approach and grammatical rules and syntactic conventions. The VADER lexicon approach contains different lexical features that include acronyms and emoticons. Hence around 7,500 sentiment features are present in its dictionary. To determine the sentiment intensity of a word, grammatical rules are considered, which can cause variation in the sentiment score of a word.

9.2.4. Step 4: Feature Engineering

Feature Engineering is a method of discovering significant characteristics from data to efficiently train machine learning algorithms or develop features from the main features [72, 37]. These features are being used to enhance the performance of machine learning algorithms [140]. This study used two methods of feature engineering as follows:

TF-IDF For information retrieval and summarization is one of the most used scoring metrics is TF-IDF. It is used for the representation of the term's significance in each text. TF and IDF are given as input in the extraction function of TF-IDF. Tokens that are infrequent in the dataset are given by TF-IDF. The significance of unusual words increases if it appears in two documents.

$$tfidf_{t,d,D} = tf_{t,d} \cdot idf_{t,D} \tag{9.1}$$

where terms are indicated by $\boldsymbol{t}$; each document by $\boldsymbol{d}$; set of documents by $\boldsymbol{D}$. Along with TF-IDF, the "n-gram range" parameter is employed. Words' weight that provides the weights of corpus for any word is calculated with the help of TF-IDF. The output is the word matrix being weighted. An increase in meaning is proportional to the count with the TF-IDF vectorizer, but the word frequency within the corpus helps to manage it. The TF technique is often considered for features extraction and is commonly used for the purpose of text classification. The terms' incidence frequency is employed as a parameter for classifier training. Unlike TF-IDF, in which less weight is given to more common terms, the TF function does not consider if a word is popular or not.

BoW One of the methods used for the simple representation of Natural Language Processing (NLP) and information retrieval is Bag-of-Words, commonly known as BoW.

It is the easiest and flexible way for obtaining the features of a document. In BoW, the histogram of the words in the text is looked at. To train the set, the words' frequency is used as a function. In this research, the CountVectorizer function using the pythons' Scikit-learn library is utilized to implement the BoW method. Vectorization is the process of converting a set of textual data into numerical vectors. Words' frequency helps in the operation of CountVectorizer, and it shows that counting of tokens is done and generation of the limited token matrix is completed [116]. The BoW is a list of features and terms that allocates a significance to each attribute which reflects the particular features' frequency [145].

9.2.5. Step 5: Model Selection

In our study, we considered the following learning algorithms to build our model:

- **Logistic Regression (LR):** It is a statistical model that is based on the concept of probability and is akin to linear regression. LR performs prediction of the outcomes' probability by fitting the data to a logistic function [53].
- **Support Vector Classifier (SVC):** It is a popular ML classifier for solving linear and non-linear problems. It works well for several practical applications [318, 264]. A line or hyperplane is generated by SVC, which separated the data into a section. Low-dimensional input space is transformed with the help of its Kernel function into higher dimensional space. This transformation means that non-separable issues are converted into separable ones. It mainly helps in solving non-linear differential problems. SVC separates the data based on labels and performs complex data transformations [65].
- **Extra Tree Classifier (ETC):** A combination of classification algorithms teaching approach in which outcomes of several de-correlated random forests collected in a "forest" are combined for generation of identifications' outcome, is Extra Tree Classifier (ETC). In principle, it is very similar to a Random Forest Classifier but differs from it in other ways, like a decision tree algorithm is built throughout the forest. In ETC, the Previous training dataset in the ET Forest is used for the

creation of decision models [267].

- **Gaussian Naïve Bayes (GNB):** It employs the Gaussian distributions for the handling of continuous attributes in the Naive Bayes classification and represents the features' likelihood based on the classes [211]. GNB assigns each data point to its nearest class. It considers the distance from the mean point as well as performs its comparison to the class variance [143]. Moreover, GNB exhibits faster performance as compared to other algorithms [251].
- **Gradient Boosting Classifier (GBC):** Decision trees are widely utilized for performing gradient boosting. As they have shown significant results in the classification of the large system, GB frameworks have gained importance in machine learning [225].
- **Ada Boost Classifier (ADA):** uses a linear combination of "weak" classifiers for constructing a "strong" classifier, like GBC. The "weak" classifier can be considered a simple threshold operation on a specific feature category. Weak classifiers' training process is known as "WeakLearn". Ada Boost consumes less memory and has fewer computational requirements.

9.2.6. Step 6: Model Evaluation

The performance of our selected models is measured using the four measurement aspects listed in Table 9.4 below where:

- Positive Predictions labeled correctly by the classifier is determined by the True Positive (TP) parameter.
- Negative Predictions labeled correctly by the classifier is determined by the True Negative (TP) parameter.
- Number of negative instances mistakenly presumed as positive instances by the classifier are determined by the parameter False Positive (FP).
- Number of positive instances mistakenly presumed as negative instances by the classifier is determined by the parameter False Negative (FP).

TABLE 9.4. Summary of performance measures, formulas, and definitions.

Measures	Formula	Definition
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	Calculates the closeness of a measured value to the standard value.
Recall	$\frac{tp}{tp+fn}$	Calculates the exact number of positive predictions that are actually observed in the actual class.
Precision	$\frac{tp}{tp+fp}$	Calculates the exact no. of correct predictions out of all the input sample.
F1-score	$\frac{2 \cdot P \cdot R}{P+R}$	Calculates the accuracy from the precision and recall.

9.3. Study Results

This section discusses the results of our research work in light of the proposed research questions.

RQ_1: How do users express their sentiments in their accessibility app review?

As the expression of users' thoughts regarding the apps, reviews are used as a tool. If the accessibility features address the users' needs, the user reviews are written with positive sentiments. On the other hand, if the accessibility features are not meeting user requirements, then attention is needed by the developers. These reviews reflect negative sentiments. Therefore, a review serves as a way to measure user satisfaction or dissatisfaction about the accessibility, and the negative reviews help identify accessibility topics that need to be fixed. In Figure 9.2, we present the comparison of sentiment analysis results between TextBlob and VADER. According to the results of TextBlob, 72.66% users have positive reviews, 15.88% have negative while the remaining 11.45% have the neutral review. On the other hand, results generated by the VADER approach show that 79.30% users have positive reviews, 10.75 are negative, and the remaining 9.95 are neutral reviews. Although there are some differences in the results of both sentient analyzers, they show a similar trend, *i.e.*, most of the users have positive reviews about the apps' accessibility, few users have negative reviews, while the least number of users have neutral views about.

RQ_2:How effective is our proposed sentiment analysis-based approach in the identification of accessibility reviews?

To analyze the sentiments of accessibility app users, we used two sentiments analyzers, i.e., TextBlob and VADER. Both sentiment analyzers help in the automatic prediction of emotions from user reviews. We also used six different machine learning models, i.e., SVC, GNB, GBM, LR, ADA, and ETC, along with TF-IDF and BoW features with the sentiment analyzers to categorize the sentiments based on the result of RQ1. Furthermore, we used four statistical measures for evaluating the proposed approach. We select the best hyper-parameters setting using the hit and trial method. During tuning each time, we split the dataset and change the model's hyperparameters values. We have done this tuning between parameter values range such as for n_estimator in RF we start from 50, and we end up at 500 while our best value was 300.

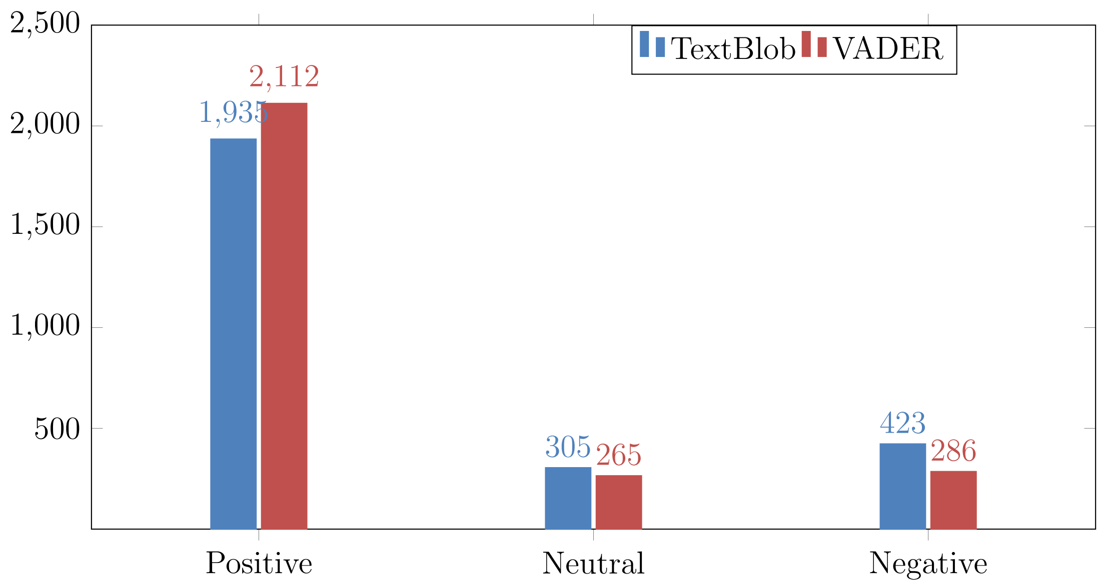

FIGURE 9.2. Comparison of TextBlob and VADER sentiment analysis results.

The results of both sentiment analyzers, i.e., TextBlob and VADER, with TF-IDF and BoW in terms of accuracy, precision, recall, and F-measure, are presented in Table 9.5. We observe that when we used the TextBlob method with the TF-IDF technique, we found that LR and ETC exhibit the highest accuracy, i.e., 0.83. While for precision and F1-Score, ETC outperforms the remaining five classifiers. In terms of a recall measure, GBM performs better when compared to other techniques with TextBlob. For the BoW technique, TextBlob with LR and SVC achieved the highest accuracy (0.86) and F1-score (0.77). TextBlob with the ETC classifier attained 0.86 precision and recall of 0.78 by using the SVC classifier.

Overall results show that TextBlob with the BoW method shows better accuracy, recall, and F1-score as compared to the TF-IDF method. On the other hand, the TF-IDF-based method outperforms BoW in terms of recall measure.

To measure the efficiency of TextBlob, which is a lexicon-based technique, we used the VADER technique on the same dataset. Based on the subjectivity and polarity, TextBlob performs assignment of the polarity score to each word ranging from -1 and 1. While VADER's performance depends on the mapping of lexicon features into sentiment scores done by a dictionary [147]. For the given dataset, the best accuracy (0.84), recall (0.64), and F1-score (0.65) for VADER with TF-IDF is achieved by GMB while ETC outperform in terms of precision (0.80). For VADER with BoW features, SVC outperform other models in accuracy (0.82), recall (0.65), and F1-score (0.65). VADER with ETC exhibits the similar trend in precision but with a lower value as compared to TF-IDF features. Overall, results show that TextBlob performs better than VADER with BoW as well as TF-IDF features.

9.4. Discussion

In this study, we applied sentiment analysis to identify the emotions of reviewed users towards the accessibility of apps. To facilitate the sentiment analysis, we used TextBlob and VADER, which are both popular lexicon-based methods. We wanted to know whether the two techniques could detect users' feelings towards accessibility in their apps based on machine learning techniques. The results of this study showed that sentiment analysis was crucial in identifying users' reviews towards the accessibility of apps, especially those that are disabled.

For many persons with disabilities (such as those who are deaf or blind), expressing their reviews towards various apps can be challenging. However, they can express their emotions (positive, neutral, or negative) towards an app, which may help developers understand whether it is accessible or not. We felt that disabled persons were not given much attention when collecting app reviews, probably because of the complexity involved in getting and analyzing their feedback. This study has shown that sentiment analysis could be the solution for determining the emotions of people with disabilities towards the accessibility of mobile devices. The findings of this study are important for software developers because it enables them to know whether disabled persons are satisfied or dissatisfied with the accessibility of their apps. Thus, developers can make any necessary changes to facilitate the use of the apps by persons with disabilities.

TABLE 9.5. Models performance comparison for TextBlob and VADER with TF-IDF and BoW features.

Sentiment Analyzer	**Model**	**TF-IDF**				**BoW**			
		Accuracy	**Precision**	**Recall**	**F1-Score**	**Accuracy**	**Precision**	**Recall**	**F1-Score**
TextBlob	**LR**	0.83	0.84	0.59	0.65	**0.86**	0.80	0.76	**0.77**
	SVC	0.81	0.80	0.54	0.60	**0.86**	0.76	**0.78**	**0.77**
	ETC	0.83	**0.88**	0.60	**0.66**	0.85	**0.86**	0.68	0.72
	GNB	0.65	0.55	0.48	0.50	0.65	0.55	0.47	0.50
	GBM	0.80	0.68	**0.66**	0.64	0.80	0.68	0.66	0.66
	ADA	0.71	0.57	0.64	0.59	0.73	0.61	0.69	0.64
Vader	**LR**	0.80	0.73	0.45	0.48	0.81	0.66	0.59	0.61
	SVC	0.80	0.72	0.43	0.46	0.82	0.65	0.65	0.65
	ETC	0.80	0.80	0.43	0.46	0.81	0.73	0.52	0.56
	GNB	0.68	0.45	0.42	0.43	0.68	0.44	0.42	0.43
	GBM	**0.84**	0.66	0.64	0.65	0.80	0.62	0.58	0.59
	ADA	0.74	0.46	0.42	0.44	0.72	0.49	0.49	0.49

9.5. Conclusion

In this chapter, we presented an automated sentiment analysis-based approach for the classification of accessibility-related app reviews to help the developers detect these issues and improve their app's performance in light of user's reviews. We employed an existing dataset that is composed of manually validated accessibility reviews. Two sentiment analyzers, namely TextBlob and VADER using TF-IDF and BoW, are utilized with TF-IDF and BoW features. Both of the analyzers are coupled with six classifiers, namely LR, SVC, ETC, GNB, GBM, and ADA. Evaluation is done using four measures, i.e., Accuracy, Recall, Precision, and F1-Score, and the results show that TextBlob outperforms VADER in the classification of app reviews. Overall, results show that the ETC classifier performed best in TF-IDF features while svc is most efficient in BoW features. Sentiment analysis results also show that most of the users have given positive reviews about the accessibility of an app.

9.6. Chapter Summary

:

This chapter proposed an approach based on machine learning and NLP methods for Emotion detection automatically tag user reviews as positive, negative, or neutral. However, having accessibility-related bugs can have severe impacts on their lives that can go from preventing them from participating in new activities, to threatening their lives in critical situations due to the sensitive nature of disabled people. Therefore, identifying and prioritizing these bugs are of crucial importance. yet, the manual identification of these bug reports is time-consuming, human-intensive, and error-prone.

In the next chapter, we present a classification-based approach for the automatic detection of accessibility bug reports to support software developers with the correction of accessibility errors in their systems.

CHAPTER 10

ON THE IDENTIFICATION OF ACCESSIBILITY BUG REPORTS IN OPEN SOURCE SYSTEMS

10.1. Introduction

Open source and industrial software utilize bug-tracking systems — also called issue-tracking systems — such as Bugzilla [79]. These tracking systems are used to help developers maintain the software by allowing the end-users to submit the issue description they faced while they are using the software. Bug reports can describe accessibility issues that could have prevented or limited users with a disability, special needs, or functional constraints [29].

People with disabilities or special needs rely heavily on accessibility software applications in their everyday life (find accessible location, customized UIs, voice translation, communication, driving, shopping, etc.). Having accessibility-related bugs can have severe impacts on their lives that can go from preventing them from participating in new activities, to threatening their lives in critical situations due to the sensitive nature of disabled people. Therefore, identifying and prioritizing these bugs are of crucial importance. yet, the manual identification of these bug reports is time-consuming, human-intensive, and error-prone. The textual nature of bug reports adds another layer of challenge related to the meaning ambiguity of these natural language descriptions. To illustrate this problem, let us consider the following two examples:

> **Example 1:** Missing labels on the buttons in the "Select how you want to use Weave [1]

This entire chapter is reproduced from Aljedaani, Wajdi, Mohamed Wiem Mkaouer, Stephanie Ludi, Ali Ouni, and Ilyes Jenhani. "On the identification of accessibility bug reports in open source systems," in Proceedings of the 19th International Web For All Conference, pp. 1-11, 2022, https://dl.acm.org/doi/abs/10.1145/3493612.3520471, with permission from the Association for Computing Machinery.

[1] https://bugzilla.mozilla.org/show_bug.cgi?id=533573

Example 2: Performance issue: TextArea very slow when accessibility API turned on[2]

While the first bug report describe a missing textual label in a graphical component, making it not accessible for blind users, the second bug report is related to a performance issue. Despite containing the keyword accessibility, this bug is not related to the accessibility of the software, but to a performance regression detected when integrating the accessibility library, through its API, to the system. These examples show that we cannot rely on the keyword accessibility to identify accessibility related bug reports, as the first example (accessibility bug report) did not contain the keyword *accessibility*, while the second example (non-accessibility bug report) did.

To support software developers with the correction of accessibility errors in their systems, we propose a classification-based approach for the automatic detection of accessibility bug reports. However, the detection of such reports is challenging, besides the inherited ambiguity of distinguishing meanings, in any natural language text, the above example show how the keyword *accessibility* can be misleading, which hardens the reliance on that keyword alone. To cope with these challenges, we design our study to harvest a potential terminology that can be used to describe accessibility errors and faults.

Our approach relies on Natural Language Processing (NLP) techniques to distill from a training sample (set of accessibility bug reports) the proper *features*, i.e., phrases that tend to specifically describe accessibility related faults in code. We performed our study on seven open-source systems hosted in two popular issue tracking systems Bugzilla [79] and Monorail [213] repositories. We mine all the bug reports for the selected projects to identify accessibility and non-accessibility bug reports based on their tags (manual inspection). To the best of our knowledge, this is the first study that builds classification models to classify bug reports and identify accessibility issues. Specifically, we address the following research questions:

RQ1: *Can we accurately detect accessibility-related bug reports?*

[2]https://bugs.chromium.org/p/chromium/issues/detail?id=868830

Our aim is to design an approach that can automatically identify accessibility-related bug reports. Therefore, we put under test, various classifiers, such as neural networks, decision trees, and SVM, known to be efficient and widely used for binary classification problems. Answering to this research question would reveal the best performing model that we should deploy for our current problem, along with showing how much we can advance the state-of-the-art of detecting accessibility-related bug reports.

RQ2: *What is the size of the training dataset needed for the classification to effectively identify accessibility bug reports?*

After evaluating the accuracy of our model, we analyze the number of bug reports needed for training in order to achieve our optimal model classification accuracy. We anticipate our model to be easily exported and extended if it can achieve an acceptable performance using a relatively small set of training data. Otherwise, if the model requires a large number of bug reports, for training, then we report a need for a considerable time and effort for labeling.

To summaries, the paper makes the following contributions:

- We present an automatic accessibility identification on seven open-source systems to identify accessibility-related bug reports by using machine learning algorithms. To the best of our knowledge, this is the first accessibility classification study to date on the bug reports dataset.
- An experimental study on a real world dataset. Our key findings show that our model accurately identifies accessibility-related bug reports with an average F-Score of 85% to 90%. Furthermore, we infer which features, *i.e.,* keywords, are relevant for the detection of such type of bug reports
- We also publicly provide our dataset that served us as the *ground-truth*, for replication and extension purposes[3].

[3]https://smilevo.github.io/access/

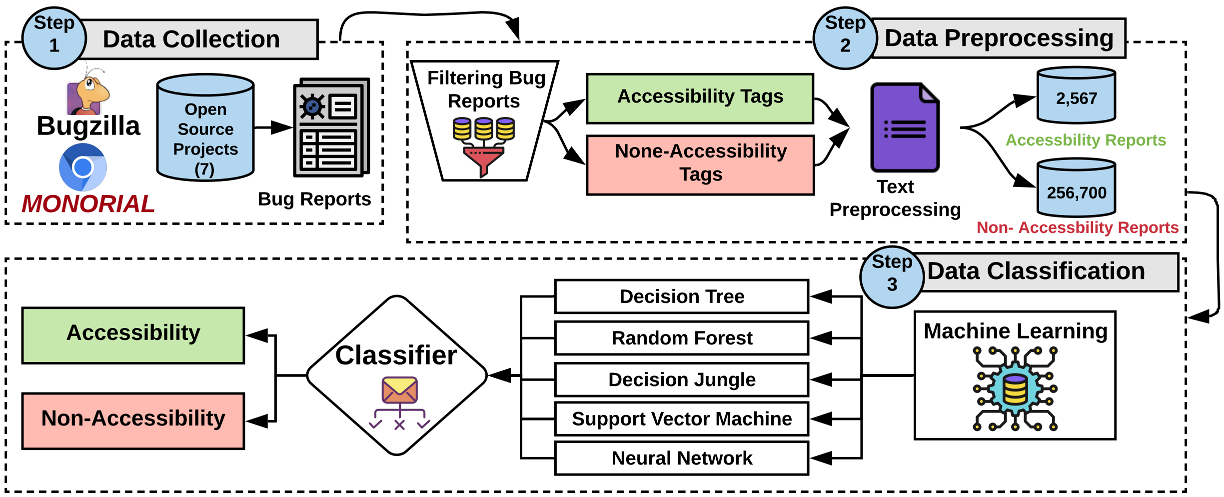

FIGURE 10.1. Overview approach of our study.

10.2. Methodology

The following section explains our methodology and how we obtained and analyzed the data for classifying accessibility bug reports to answer the research questions of our study. Figure 10.1 presents an overview about our study which consists of the following main steps:

TABLE 10.1. Statistics of the datasets.

System	Platform	#Non-Bug Reports	#Accessibility Bug Reports	Start Date	End Date
Firefox	Firefox	25,000	250	29-09-2000	06-04-2020
	Core	59,900	599	08-04-1997	05-04-2020
Chromium	Mac	30,700	307	23-09-2016	05-03-2020
	Windows	44,200	442	28-09-2016	05-03-2020
	Chrome	41,200	412	08-05-2017	05-03-2020
	Android	34,700	347	10-12-2012	05-03-2020
Apache	NetBeans	21,000	210	14-07-2000	02-01-2018
Total		**256,700**	**2,567**		

(A) **Data Collection (Step 1):** As an initial step of our study, we need to collect our experimental dataset which consists of a set of real world bug reports from open source projects. To do so, we mine the bug reports archive of seven selected open-source systems. We have implemented a parser that takes every bug report in the tracker as an input, then verifies whether it was tagged as an accessibility reports. If so, its corresponding information will be copied over to our database. We keep track of the project containing the bug report along with all the its metadata. It is important for us to keep as much information as possible about each bug report, so that the manual analysis that would be coming later would be easier for the authors.

(B) **Data Preprocessing (Step 2):** After the data collection step, we need to pre-process the text and only keep important textual information, which can be used to train a model afterwards [40]. The results of this setp put the report's text into a format that the classification algorithms can easily transform. This way, the noise will be removed, allowing for informative featurization. Note that we only pre-process the textual description of the reports, and we do not alter any meta-data information.

(C) **Data Classification (Step 3):** In the final step, we apply machine learning techniques to build a classification model. In particular, a binary classifier is used to classify accessibility bug reports on five widely-used algorithms. We only used bug report description to identify accessibility bug reports.

10.2.1. Step 1: Data Collection

Data collection is the first step in our study methodology. Our goal is to analyze bug tracking systems of various open-source software projects where their reports are publicly available. Our study uses two of the large open-source bug report repositories, Bugzilla [79], and Monorail [213]. We chose various project system domains that range from web browsers, mobile platforms, and desktop applications. We have also chosen these projects because they contain accessibility frameworks, integrated as libraries, and heavily used in their systems, to make their content and services accessible. We collected more than 15 projects to be analyzed, as our focus was only on bug reports identified as ***defect*** type. In order to select a project repository to be studied, we selected projects that support the type of bug report in their repositories, and we eliminated the projects that do not support the information of bug report types. From all the 15 projects, there are only seven projects that supported the bug report types. The projects that used Bugzilla are Firefox-Platform [4], Firefox Core, Apache NetBeans, and projects that use Monorail are Google Chromium platforms[5] (Android, Windows, Chrome, and Macintosh).

After collecting all the bug reports, we discarded bug reports that were reported in a different language than English, and bug reports that were flagged as invalid, or not relevant. We provided some examples in Table 10.2.

TABLE 10.2. Examples of invalid bug reports.

Type	**Description**
Non-English	Girdiğim eğitim sitesi güvenlik hatası veriyorhttps://bugs.chromium.org/p/chromium/issues/detail?id=1024836
Testing	testing a bughttps://bugzilla.mozilla.org/show˙bug.cgi?id=599707
Non-Meaningful	afdfsadsdsadhttps://bugzilla.mozilla.org/show˙bug.cgi?id=668458
Thanking	Thank youhttps://bugs.chromium.org/p/chromium/issues/detail?id=910827

As our accessibility bug reports were gathered based on their accessibility tags by the developers reporting them, and validated using the keywords that exist in the BBC guidelines [63, 311], we followed the process of [178] to further verify the collected data, which are referred to as accessibility bug reports. We randomly selected a 12% sample of bug reports, i.e., 334 out of the 2,567 bug reports. This quantity is equal to a sample size with a confidence level[6] of 95% and a confidence interval of 10. Two of the authors performed the labeling process separately. Both authors were given the same set of bug reports to label to either accessibility related or not. The chosen reports were not previously exposed to the authors. The analysis process took seven days to prevent exhaustion. The authors had the ability to search online for any unknown references in the reports. We cross-check results of the manual labeling to calculate the ratio of agreement and disagreement between the authors. For all cases of disagreement, a third author is requested to re-label the instance and break the tie. We present an example of an agreed on and disagreed on bug report. For the example of the disagreed on bug report, the third author has considered this to be a

[4]https://bugzilla.mozilla.org/home

[5]https://bugs.chromium.org/p/chromium/issues/list

[6]https://www.surveysystem.com/sscalc.htm#one

non-accessibility bug report, as it describes an error with handlers of copying accessibility.

> **Agreed on Example:** Panning incorrect with Fullscreen Magnifier Accessibility feature enabled while display set in a non- tablet rotatio[7]
>
> **Disagreed on Example:** Copy for accessibility permission incorrect for some PDFs with revision 2 security handlers[8]

We adopted Cohen's Kappa coefficient [94] to assess the inter-rater agreement level for the categorical classes. We obtained an agreement level of 0.83. According to Fleiss et al. [121], these agreement values are considered to have an almost *perfect agreement* (*i.e.*, 0.61–0.80).

To summarize, we only considered the bug report that is typed as a defect or bug. We discarded the bug report that typed as enhancement, task, feature, or patch. After finalizing our target projects, we collected all bug reports archived in each of the selected project systems. The total number of **A**ccessibility **B**ug **R**eports (ABR) are 2,567 while the total number of non-accessibility bug reports are 256,700. Note that after we gathered all the defect bug reports in each project, we randomly selected non-accessibility bug reports. Table 10.1 illustrates the details of the collected data in the study. Table 8.1 also showcases the keywords we encountered during our manual analysis, and how they related to various types of accessibility guidelines.

10.2.2. Data Preprocessing

We text preprocessing (TP) the textual information in each bug report in the *description* field. The bug report description d can be mixed with words and different characters, for example, comma, apostrophe, etc. In the text preprocessing, we clean up the documents by removing the unhelpful elements of special characters and stopping words such as a, the, etc. Then, we use Natural Language Processing[9] (NLP) for identifying the basis of each word. Words can be written in different grammar styles, but the meaning is similar.

[7]https://bugs.chromium.org/p/chromium/issues/detail?id=1009329

[8]https://bugs.chromium.org/p/chromium/issues/detail?id=989408

[9]https://nlp.stanford.edu/software/

Text Preprocessing (DPP)

$$\hat{d} = DPP(d) \tag{10.1}$$

For example, d is a bug report description, then $\widehat{d}$ is generated, using DPP. The DPP process is explained as below: **input (d)**: 'Print dialog too large for screen when accessibility features being used; needs to be resizable'[a]

1- Tokenization: In this step, we transform the textual information "words" into a tokens list as each single token will be processed separately.

['Print', 'dialog', 'too', 'large', 'for', 'screen', 'when', 'accessibility', 'features', 'being', 'used', ';', 'needs', 'to', 'be', 'resizable']

2- Numerical & Special Characters Removal: In this part, all the numbers and special characters (punctuation) will be eliminated, tags for instance ';' from the tokens list.

['Print', 'dialog', 'too', 'large', 'for', 'screen', 'when', 'accessibility', 'features', 'being', 'used', 'needs', 'to', 'be', 'resizable']

3- Stop-Word Removal: is the process of deleting all the common English words as well as reserved words[b] such as "too", "for", "be", "to", "when", "need".

['Print', 'dialog', 'large', 'screen', 'accessibility', 'features', 'used', 'needs', 'resizable']

4- Lemmatization: In this step, we minimize the words' derivationally to the root of the words, which helps remove the inflection. For example *prints, printing, printed,* $\Rightarrow$ *print, features,* $\Rightarrow$ *feature.*

['Print', 'dialog', 'large', 'screen', 'access', 'feature', 'use', 'size']

5- Ouput($\widehat{d}$): This is the final step where all the characters converted into lowercase and then merge all the tokens to a single string.

'Print dialog large screen access feature use size'

[a] https://bugzilla.mozilla.org/show˙bug.cgi?id=327939

[b] http://www.textfixer.com/resources/common-english-words.tx

10.2.3. Data Transformation

Given machine learning, all the algorithms used in machine learning are trained using *feature vectors*. The feature vector is a numerical vector with data in numerical form [210]. The collected data that we used in this study does not come in vector form. Therefore, those data have to transform into feature vectors using feature extraction before using the data to train the machine learning algorithms. To transform the data to feature vector, we used the feature hashing technique known as the Hashing trick. Feature hashing is a popular and powerful technique in machine learning for handling spares and high dimensional features [314]. It is applied to reduce the dimensionality of the analyzed data [274, 323]. The hashing function does this transformation in the feature hashing technique. For example, if we input a bug report *description* using the feature hashing technique, it will output a fixed-length size of a hash value. Figure 10.2 shows an example of how feature hashing works.

There are several popular schemes for feature encoding or extraction like Bag-of-words, TF-IDF, etc. However, there are issues in the mentioned techniques, such as the curse of dimensionality as well as semantic selection. Prior studies have shown that feature hashing is a robust approach to achieve fast similarity search [303, 150, 156]. If we use a feature selection technique other than the feature hashing technique in this study, it will select a subset from the original features and reduce the parameter vector's size, but still, we need to map from string to integers. Nevertheless, if we apply feature hashing, it will automatically do the mapping into a hash function; thus, there is no need for mapping strings to integer. Performing text hashing increases efficiency and scalability in the classification of big data analytics.

10.2.4. Data Classification

The primary objective of the classification step is to train a binary classifier that classifies whether a bug report is accessibility or non-accessibility after learning from the bug report that we have identified as an accessibility bug reports of all the different projects. Data obtained from Bugzilla and Monorail archives for accessibility bug reports was highly imbalanced; precisely, the number of non-accessibility of bug reports is much higher than the accessibility of bug reports. Imbalanced data restricts the standard deviation in the majority of the classification approaches from operating well with the lower classes (i.e., the class containing significantly lower data). There are different machine learning methods designed to solve this issue [133]—for instance, data resampling techniques, gradient boosting techniques (tree-based models), and ensemble techniques. Similar costing methods become computationally costly to sample approaches, such as the Synthetic Minority Over-sampling (SMOTE) [87]. Therefore, the main reason for selecting the ensemble learning techniques in our study is that we can combine several different classification methods to overcome imbalanced data.

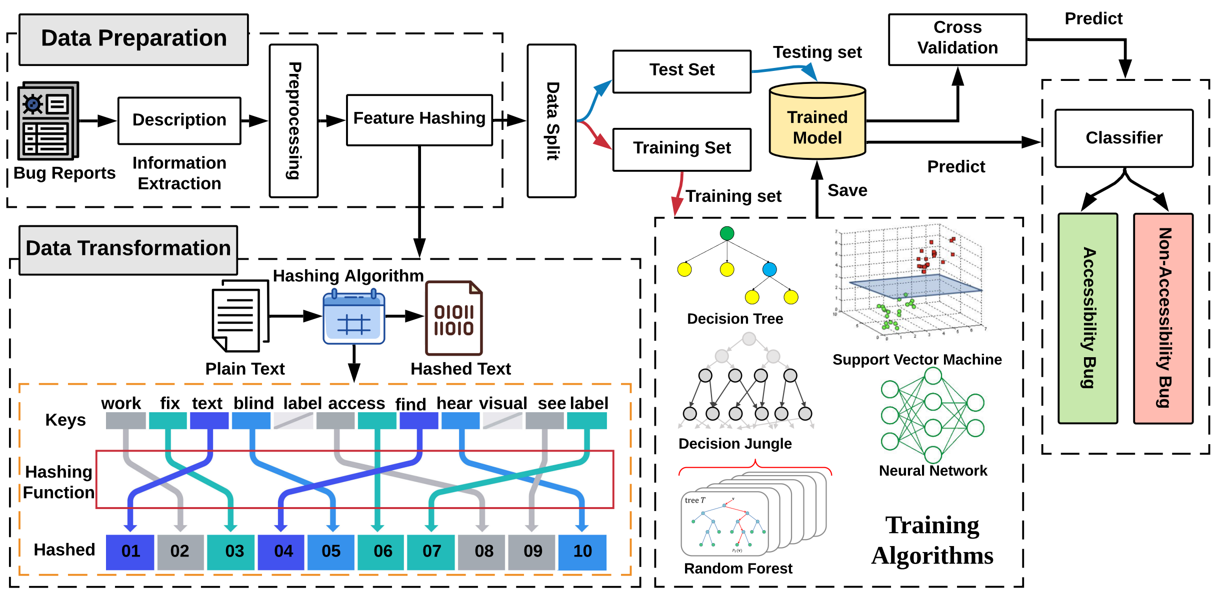

FIGURE 10.2. Overview approach of data preparation and data transformation.

10.2.5. Machine Learning Algorithms

Choosing a suitable classifier that can provide an optimal identification for our study purpose is not a straightforward task [120]. Our study addresses a binary classification (two-class) problems, as our dataset is being classified into two classes, accessibility bug reports, and non-accessibility bug reports. Since we have a dataset already labeled as two classes, our methodology depends on supervised machine learning algorithms to allocate each bug report into one of the defined classes. We evaluated five different machine learning algorithms to observe which one offers the most successful outcomes for the classification of accessibility and bug reports. Specifically, Decision Tree (DT), Random Forest (RF), Decision Jungle (DJ), Support Vector Machine (SVM), and Neural Network (NN). We selected these algorithms because they are widely used in the literature of software defect classification [234, 184, 132, 300], also they are stated to work well with the imbalance datasets, and NLP in literature [125, 167]. To enable replication of our findings, we present the chosen key parameters for the selected machine learning algorithms techniques, as described in the Table 10.3.

TABLE 10.3. Summary of the hyperparameter in machine learning algorithm.

Algorithm	**Hyperparameter**	**Default**	**Description**
	max˙n˙leaf	20	The maximum number of leaves per tree
Decision Tree	min˙samples˙leaf	10	The minimum number of samples per leaf node
	learning˙rate	0.2	Learning rate
	n˙tree	100	The number of trees constructed
	n˙estimators	8	The number of decision trees
Decision Forest	max˙depth	32	The maximum depth of the decision trees
	n˙samples˙leaf	125	The number of random splits per node
	min˙samples˙split	1	The minimum number of samples per leaf node
	n˙estimators	8	The number of decision directed acyclic graphs
Decision Jungle	max˙depth	32	The maximum depth of the decision directed acyclic graphs
	max˙width	128	The maximum of the decision directed acyclic graphs
	n˙optimiz	2048	The number of optimization steps per decision directed acyclic graphs layer
	n˙iter	1	The number of iterations
SVM	Lambda	0.001	The Lambda
	n˙nodes	100	The number of hidden nodes
Neural Network	learning˙rate	0.1	Learning rate
	n˙learning˙rate	100	The Number of learning iterations
	learning˙rate˙weights	0.1	The initial learning weights diameter
	momentum	0	The momentum

10.2.6. Evaluation Metrics

This study used a 10-fold cross-validation method to train the model to assess the variability and reliability. For individual models, we distributed our dataset in 10 folds of the same size bug reports. Afterward, we performed 10 tests with separate data sets, during which 9 folds were utilized in each assessment as training sets and the remaining fold used as test sets. Then, we evaluate these machine learning models' performance in terms of accuracy, precision, recall, and F_1 score. These evaluation parameters are mostly used in binary classification problems, as in our case [132]. For each evaluation metric, the score rank is between 0.0 and 1.0, where 0.0 represents the classifier's lowest performance, while the 1.0 score represents the classifier's highest performance.

Accuracy

provides the score, which shows that how much a classifier is accurate. It can be defined as the total number of correct predictions divided by the total number of predictions. It works well on a balanced dataset.

$$Accuracy = \frac{total\ number\ of\ correct\ predictions}{total\ number\ of\ predictions} \tag{10.2}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{10.3}$$

- **TP** is the classification made by a classifier as "Yes" against an example, and the actual label of the example is also "Yes".
- **TN** is the classification made by a classifier as "No" against an example, and the actual label of the example is also "No".
- **FP** is the classification made by a classifier as "Yes" against an example, but the actual label of the example was "No".
- **FN** is the classification made by a classifier as "No" against an example, but the actual label of the example was "Yes".

Precision

is also known as a positive predictive value, which is the fraction of relevant examples among the retrieved examples. It tells us the number of correct positive classifications from the classifier's total number of positive classifications. It can be calculated as:

Table 10.4. Distribution of the number of non-accessibility bug reports dataset divided in ten iterations.

Platforms	#ABR	#Non *ABR X2*	#Non *ABR X10*	#Non *ABR X20*	#Non *ABR X30*	#Non *ABR X40*	#Non *ABR X50*	#Non *ABR X60*	#Non *ABR X70*	#Non *ABR X80*	#Non *ABR X90*	#Non *ABR X100*
Firefox	250	500	2,500	5,000	7,500	10,000	12,500	15,000	17,500	2,000	22,500	25,000
Core	599	1,198	5,990	11,980	17,970	23,960	29,950	35,940	41,930	47,920	53,910	59,900
Mac	307	614	3,070	6,140	9,210	12,280	15,350	18,420	21,490	24,560	27,630	30,700
Windows	442	884	4,420	8,840	13,260	17,680	22,100	26,520	30,940	35,360	39,780	44,200
Chrome	412	824	4,120	8,240	12,360	16,480	20,600	24,720	28,840	32,960	37,080	41,200
Android	347	694	3,470	6,940	10,410	13,880	17,350	20,820	24,290	27,760	31,230	34,700
NetBeans	210	420	2,100	4,200	6,300	8,400	10,500	12,600	14,700	16,800	18,900	21,000
Total	2,567	5,134	25,670	51,340	77,010	102,680	128,350	154,020	179,690	205,360	231,030	256,700

Recall is also known as sensitivity. It tells us how many correct positive predictions are made by a classifier from the total number of actual positive predictions. It can be calculated as:

$$Recall = \frac{TP}{TP\ +\ FN} \tag{10.4}$$

F-score

is also known as the F_1 score or F measure. It is a harmonic mean of precision and recall. It can be calculated as:

$$F - score = 2\ *\ \frac{Precision\ *\ Recall}{Precision\ +\ Recall} \tag{10.5}$$

AUC stands for the area under the curve, AUC is a performance measurement for classification problems. It tells us about the successful classification rate of a classifier.

10.3. Study Results

RQ1: *What is the accuracy of different models in detecting bug reports?*

Approach. In this research question, we double the number of the accessibility bug report for each project to run the experiment of RQ1. For instance, the Firefox platform contains 250 accessibility bug reports (ABR), so we double the number of non-accessibility bug reports (Non-ABR X2) to become 500 bug reports as shown in the Table 10.4. Then, we conducted a 10-folds cross-validation [164] procedure to split our data into training data and evaluation data on the five machine learning models. We use cross-validation because the dataset we used in this study is an imbalanced dataset. Therefore cross-validation is a more appropriate approach as compared to the conventional train test split approach. To evaluate our result in RQ1, we used our performance evaluation accuracy, precision, recall, F score, and AUC, which are described in detail in Section 10.2.6.

Results. The result of all machine learning models show in Table 10.5. The model performs differently in different scenarios, such as when we apply the machine learning model for the Firefox project bug report classification. Neural networks outperform all other models in accuracy, precision, recall, F-score, and AUC by achieving the 0.91, 0.94, 0.88, 0.90, and 0.97. Decision Tree is the only model with the same accuracy, recall, and F score as a Neural Networks. However, Neural Network achieves high precision and AUC score than all other models; thus, Neural Networks is significant in the case of the Firefox project.

Decision Tree outperforms all other models in terms of all evaluation parameters in the Core project bug report classification. The Decision Tree achieves 0.92 accuracy, precision, recall, F score, and 0.96 ACU score, while SVM performs poorly in this case with a 0.87 accuracy score. Decision Tree also performs well in the Windows project and achieves the 0.89 accuracy score, same as in Core project bug reports classification, and SVM is the worst performer than all other Windows project models.

In Netbeans and Android bug reports classification, Random Forest and Neural Networks perform significantly than other models and achieve equal accuracy. In terms of the AUC neural network, lead the table with a 0.92 score. Mac project bug reports classification is the only case where Decision Jungle achieves high accuracy. In this case, Random Forest also achieves the same accuracy as Decision Jungle, so both share their Mac case's significant performance.

Discussion. According to the result, all tree-based ensemble models such as Decision Tree, Random Forest, and Decision Jungle perform better than the linear model SVM, except the case on the Chrome project. The reason for the better performance of the tree-based ensemble model is that when the number of base learners work on a single problem, it performs better than an individual learning model. Random Forest and Decision Jungle are ensemble models that make a final prediction based on their numbers of decision tree predictions using voting criteria. Random Forest is better than Decision Jungle in some cases where data is more imbalanced because Random Forest controls the over-fitting problem on imbalanced data more efficiently [277]. After all, each tree in Random Forest is constructed on a bag, and each bag is a uniform random sample from the original dataset with the replacement of samples, that the reason tree in Random Forest is biased in the same direction and magnitude (on average) by class imbalance.

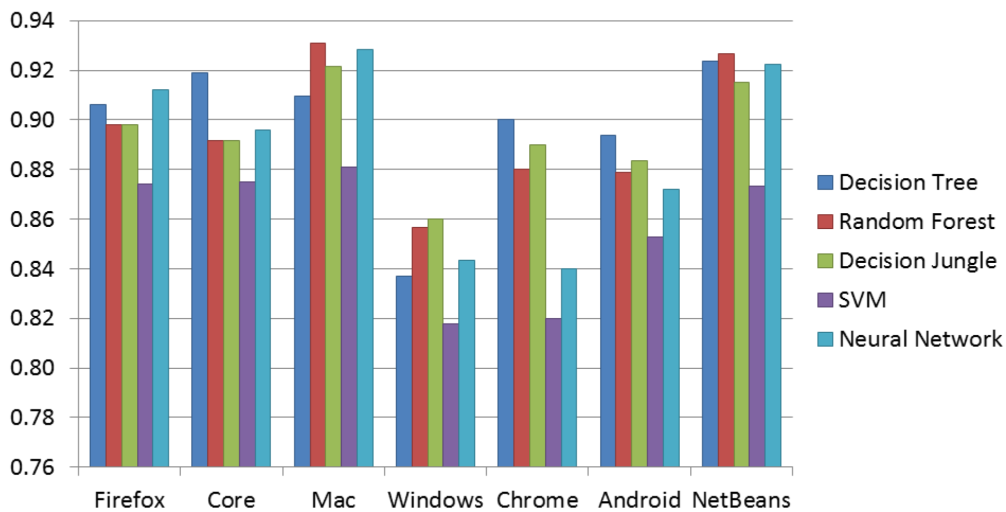

FIGURE 10.3. Distribution of classification accuracy metric in all classifiers.

On the other linear model, SVC shows poor performance in all cases except chrome project in term of the accuracy, as shown in Figure 10.3 because its kernel trick is not to consider more suitable to boost the performance on the small and imbalanced dataset as compare to a tree-based model that can perform better also on small data size. Neural Network is also performed better in all cases and beats the SVM, where it reaches 0.95 accuracy in the Chrome project. There is no significant difference in the tree-based model and Neural Network model performance. To compare all classifiers results, Random Forest and Decision tree are a more fitting model as compare to others. In the case of Chrome, it achieves the highest accuracy of all this study 0.97.

TABLE 10.5. The results of the classifiers.

Project	Classifier	Mean				
		Accuracy	Precision	Recall	F-Score	AUC
Firefox	Decsion Tree	0.91	0.92	0.88	0.90	0.95
	Random Forest	0.90	0.94	0.85	0.89	0.96
	Decsion Jungle	0.90	0.93	0.86	0.87	0.93
	SVM	0.87	0.88	0.86	0.87	0.93
	Neural Network	0.91	0.94	0.88	0.90	0.97
Core	Decsion Tree	0.92	0.92	0.92	0.92	0.96
	Random Forest	0.89	0.90	0.88	0.89	0.96
	Decsion Jungle	0.89	0.91	0.87	0.89	0.94
	SVM	0.87	0.87	0.88	0.87	0.94
	Neural Network	0.90	0.90	0.89	0.89	0.95
Mac	Decsion Tree	0.84	0.83	0.85	0.84	0.92
	Random Forest	0.86	0.88	0.82	0.85	0.91
	Decsion Jungle	0.86	0.92	0.79	0.85	0.91
	SVM	0.82	0.84	0.80	0.81	0.89
	Neural Network	0.84	0.86	0.83	0.84	0.90
Chrome	Decsion Tree	0.90	0.90	0.90	0.90	0.95
	Random Forest	0.80	0.89	0.87	0.88	0.94
	Decsion Jungle	0.89	0.93	0.84	0.88	0.93
	SVM	0.82	0.84	0.80	0.82	0.90
	Neural Network	0.84	0.85	0.83	0.84	0.92
Windows	Decsion Tree	0.89	0.89	0.89	0.89	0.95
	Random Forest	0.88	0.90	0.85	0.87	0.94
	Decsion Jungle	0.88	0.92	0.84	0.88	0.94
	SVM	0.85	0.86	0.85	0.85	0.92
	Neural Network	0.87	0.88	0.87	0.87	0.94
Android	Decsion Tree	0.92	0.92	0.93	0.92	0.96
	Random Forest	0.93	0.92	0.93	0.93	0.96
	Decsion Jungle	0.92	0.93	0.89	0.91	0.95
	SVM	0.87	0.88	0.87	0.87	0.93
	Neural Network	0.92	0.92	0.93	0.92	0.96
NetBeans	Decsion Tree	0.91	0.92	0.91	0.91	0.96
	Random Forest	0.93	0.96	0.90	0.93	0.97
	Decsion Jungle	0.92	0.94	0.90	0.92	0.96
	SVM	0.88	0.91	0.86	0.88	0.96
	Neural Network	0.93	0.94	0.92	0.93	0.98

*RQ*1 Summary

We find that tree-based and Neural Networks classifiers perform better than linear model (SVM) classifier when classifying accessibility bug reports. However, *Decision Tree*'s performance significantly outperforms all other classifiers in terms of evaluation parameters. In terms of projects, NetBeans and Android bug reports are more correctly classified in comparison with other projects.

RQ2: *What is the size of the training dataset needed for the classification to effectively identify accessibility bug reports?*

Approach. This question aims to investigate the size of the dataset needed for the classifiers to classify the accessibility bug reports. To examine this, we performed the RQ2 by incrementally increase the dataset size step by step. We apply this approach to ten iterations. For the first iteration, we randomly selected 10 accessibility bug reports, 100 non-accessibility bug reports. Then we used the Random Forest classifier to examine the outputs of the study experiment. We performed the same approach in the second iteration, but we increased the dataset (double size) as the first iteration. We randomly selected 20 accessibility bug reports and 200 non-accessibility bug reports. We apply this method until we reach the ten iterations with 100 accessibility bug reports and 1000 non-accessibility bug reports. We separately examined each project to find out if different projects needed less or more dataset to classify. For the evaluation parameters of RQ2, we used F1-Score, since accuracy is not considered the best parameters because we have an imbalanced data issue by incrementally increasing each iteration by adding the non-ABR reports.

To assess this RQ2, we performed 10-folds cross-validation techniques. We collected all the results of the F1-Score for all the ten iterations, as shown in Figure 10.4. When the F1-Scores present stability in the works, we consider the number of accessibility bug reports needed for classification to classify the accessibility bug reports.

Results. The machine learning model performance depends on the size of data and the feature correlation with the target class. In this study, the experiment performs on different project dataset using machine learning algorithms to analyze the impact of the dataset size on model performance. The Figure 10.4 show the performance random forest on the different project data. As in Figure 10.4 on the X-axis number show the iterations, we increase the size of data for each class after each iteration. In the first iteration, when we train random forest, the first project dataset contains ten records for the ABR and 100 for non-ABR, and on the second iteration, there are 20 records for ABR and 200 records for non-ABR, and this procedure applied to all of the ten iterations. If we analyze random forest performance in each project, we can see that it is more consistent as we increase the dataset size. Random forest performance evaluates using the F1 score because the ratio of target classes (ABR & non-ABR) is unequal in the dataset. After all, the F1 score can better interpret the machine learning model performance.

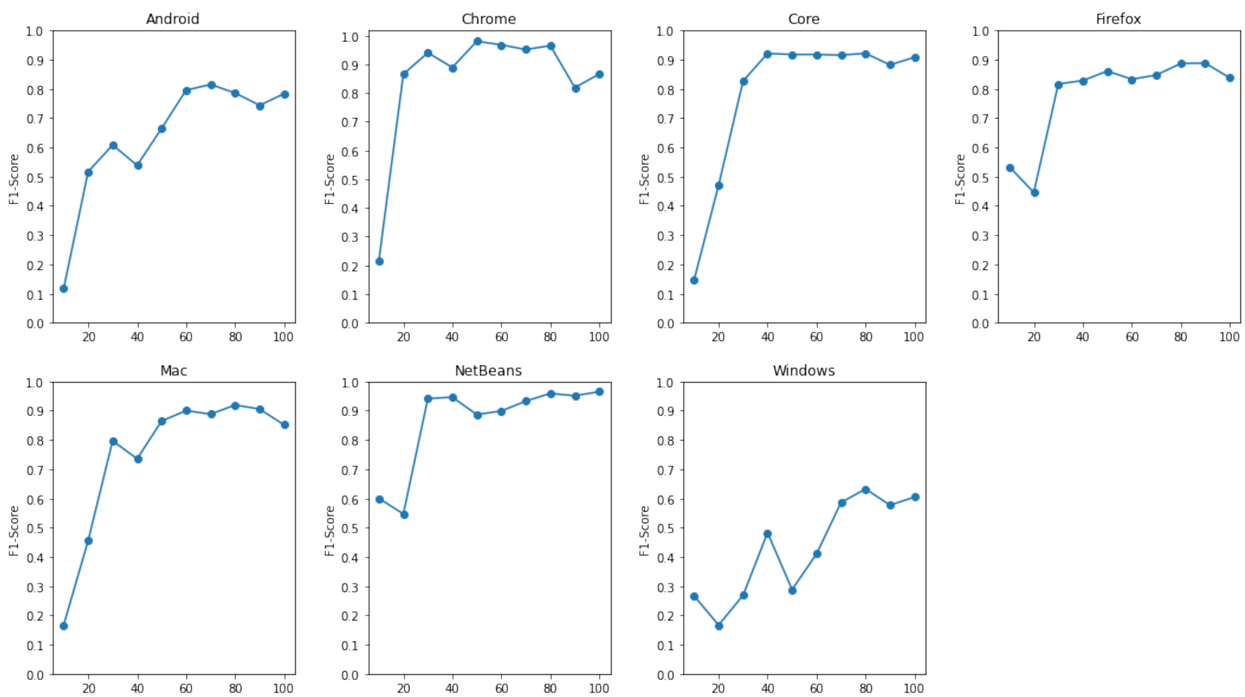

FIGURE 10.4. Distribution of classification F1-score in random classifier when incrementally increase accessibility bug reports in ten iterations.

The Android project model performs very poorly in the first iteration when there are only ten records in the dataset, but as the dataset size increases model performs gradually, and after six iterations, it becomes more consistent, as shown in Figure 10.4. Random forest performance on the Chrome project is different as compared to Android. Radom forest performs extraordinary only after one iteration and achieves the 0.87% score, which is the highest score on the second iteration compared to the other models. Random forest perform more accurately in chrome second iteration even on a small dataset, and its reason can be a good correlation between the features and the target class in the chrome dataset. There is little fluctuation in the ninth and 10th iteration in the F1 score on the chrome dataset, as illustrated in Figure 10.4. Random forest becomes consistent in performance after the fourth iteration in Core and Mac dataset and maintains its consistency until the tenth iteration.

NetBeans and Firefox datasets also contain more better-correlated features for target classes because random forest performs very well on these two projects only after three iterations and becomes a consistent performer after the third iteration. The performance of random forest is different on Windows project data as compare to the others. The Window project dataset model performed very poorly and achieved the highest 0.63% F1 score from all ten iterations, but the model becomes consistent in the score after the seventh iteration, which shows that the model is more accurate when dataset size becomes large.

*RQ*2 Summary

We find that to achieve a performance equivalent to 93% of the high F-measure score, only one fold of bug reports is required for the training of the binary classifier. In terms of projects, NetBeans and Android bug reports seem to contain the highest number of discriminative keywords, yielding in better accuracy of the classification, in comparison with other projects.

10.4. Conclusion

In this chapter, we tackled the detection of accessibility bug reports as a binary classification problem. We challenged various classifiers using a large set of reports, exported from multiple open-source projects. Our experiments show that the *Decision Tree*'s performance significantly outperforms all other classifiers in terms of evaluation parameters.

In the future, we plan to study the applicability of our approach to other projects developed in different programming languages, and to other domains. Another potential research direction is to use the current findings to build a model that handles the class imbalance problem, in the context where the number of accessibility bug reports becomes a minority class, which hinders the learning of its discriminative features.

10.5. Chapter Summary

: This chapter proposed a classification-based approach for the automatic detection of accessibility bug reports to support software developers with the correction of accessibility errors in their systems.

In the next chapter, we go into additional detail about where our study fits in the spectrum of previously conducted studies and how it has implications for the field of contemporary research and practice.

CHAPTER 11

RESEARCH IMPLICATION

This section further discusses positions our work in the spectrum of existing studies and how it implicates current research and practice.

11.1. Goal 1: *To identify the accessibility problems and challenges faced by students.*

11.1.1. Implications for Practitioners and Researchers

Implication 1: Improve collaboration and partnership. It has been clear that all stakeholders should be involved in improving deaf education. The proposed solutions indicate the important role played by the government, teachers, parents, and specialists in improving education outcomes for deaf students. Using the example of Saudi Arabia [193], governments can play a crucial role to help in creating a conducive environment for deaf education. Furthermore, in Italy, Tomasuolo et al. [297] explain the crucial role of stakeholder lobbying by deaf organizations such as the World Federation of deaf (WFD), the Italian National Deaf Association, among others. It is noted that collaboration between deaf community members, deaf organizations, scholars, and activists in many countries around the world has led to greater access to education, improved use of captions, greater use of Text apps, broadcasting of content that considers the deaf community, utilization of clear masks, among others [285, 236]. Therefore, such collaborations and partnerships provide important opportunities for improving the quality of deaf education in the current pandemic.

Implication 2: Simplify the LMS systems. Our study has shown that the mere availability of the LMS systems does not guarantee quality online education for deaf students. Indeed, the switch to online learning has been abrupt due to COVID-19, and most deaf students faced tremendous challenges in accessing the content on LMS platforms [193]. It has also been observed that there were predominant challenges in ensuring an uninterrupted-learning environment via video conferencing, for example, whether Zoom could adequately display LMS-located content or not [191]. Such systems need to be simplified and customized to improve their usability features and look and feel for deaf students. LMS systems are extremely important for remote access to materials and learning for deaf students. The suggestion for their simplification is a crucial takeaway that should be taken into account so that deaf and hard-of-hearing students can fully take advantage of such platforms.

Implication 3: Simplify LMS systems. Our findings showed that most students were facing problems navigating through the LMS systems, blackboard in particular. For instance, they did not know how to change languages, switch between content, obtain course materials, among others. The problem is worse for deaf students, who cannot follow audio directions on the systems. Such technical issues were also identified by Alsadoon and Turkestani [46] as significant barriers to e-learning for deaf students. It will be important for software engineers to investigate how LMS systems can be simplified for deaf students.

Implication 4: E-learning limitation. We found that the inadequacy of tools with limited cameras that can be visible to teachers and students posed great challenges to deaf education. The tools do not provide subtitles, and for Zoom, they provide a caption for the stream class, without supporting languages, such as Arabic. Such a finding greatly affected deaf students' learning because they cannot hear what is being said but depend on what they see on the screens. The importance of visual media in education is also indicated by Fernandes et al. [119] in Indonesia, who found that the effective use of videos greatly promoted education. Improvement of such aspects could greatly help in improving deaf education. Future researchers can compare the effectiveness of various e-learning tools to suggest which are more appropriate for deaf students.

11.1.2. Implications for Educators

Implication 1: Lack of experience. It was clear that teachers and students are not trained on the tools or do not even have good documentation to follow. Without such training, there were problems in how both students and teachers used the technology, leading to challenges in education. For deaf students, in particular, have not been trained to use the special tools needed to facilitate their education. Such findings corroborate a similar study by Krishnan et al. [168] in Malaysia that demonstrated issues in e-learning due to a lack of familiarity with technology. Other scholars can investigate the range of training programs, methods, and guidelines that would be useful in enlightening the population on how to undertake online education effectively.

Implication 2: Provide necessary equipment and technology. We have established that a lack of equipment such as hearing aids and inaccessibility to the internet are major obstacles impeding deaf education in the COVID-19 pandemic. The problem is worse in rural areas and those with high levels of poverty [193]. As further indicated by Paatsch and Toe [237], global research has shown that many deaf students attend mainstream classes that do not have adequate support for the difficulties that such students face. It has also been demonstrated that deaf students face challenges when using Zoom platforms, especially given that the platform has a steep learning curve and its features are not easily understood by all students [295]. One of the technologies lacking for many deaf students is Remote microphone (RM) hearing assistive technology (HAT), which should be customized to the needs of every student [152]. It is important to address such issues in order to promote remote deaf education during the current pandemic.

Implication 3: Improve accessibility and usage of learning materials. We have noted that many institutions have digitized their content, however it is still inaccessible due to lack of captioning and unclear audio, among other issues. Such a finding is consistent with Fernandes et al. [119], who found that learning materials for deaf students should meet the validity and effectiveness so that they can be of help to deaf students. However, it is not translated even when such content is accessed, and there are no speech-to-text services. Furthermore, deaf students find it hard to follow the teacher during virtual classes, when several faces are appearing on the screen simultaneously, or when captions' speed is fast [169]. The lack of self-explanatory images, presence of background music, and inclusion of unnecessary decorative details also make the accessibility of learning materials difficult [171]. It is important to provide visual materials and techniques that will help deaf students learn more effectively [46]. Another accessibility challenge during the COVID-19 pandemic is that the use of face masks by teachers on online platforms makes it hard for deaf students to read lips, which is a major challenge in their learning that should be overcome by using clear masks [289]. The provision of accessible learning materials will be very important in improving deaf education.

Implication 4: Cater for the mental health needs of deaf and impaired students. We have found that some students developed mental health issues during the pandemic, while others already had them prior. As explained by Krishnan et al. [168], such a situation has been brought about by the social distancing and related protocols during the COVID-19 pandemic, which has added to their isolation and lack of social interactions. Swanwick et al. [290] indicates that deaf students faced social exclusion even before the pandemic, but the current situation has exposed and deepened the issue. The pandemic has also led to negative emotional responses from deaf students because the pandemic has led to the school closing, fear of illness, social distancing, among other family problems [284]. It has been noted that deaf students are psychologically resistant to the effects of the pandemic but show less mental resilience compared to normal hearing students [321]. Providing counseling and psychological services is crucial.

11.2. Goal 2: *To provide developers with insights on how to ensure software accessibility.*

11.2.1. Implications for Practitioners and Researchers

Implication 1: App reviews are rich source of information that can be mined to identify specific accessibility problems with the mobile app. There are so many accessibility guidelines that developers and designers can find it difficult to test for all of these guidelines. Additionally, adhering to these guidelines does not necessarily guarantee the accessibility of the said app. Also, usability testing with different groups of people with disabilities, e.g., blind or deaf, can be infeasible especially for medium and small-scale companies. One way to discover accessibility problems which prior testing did not reveal is to listen to the users and learn from the reviews they wrote. Our approach can aid technology professionals to quickly spot accessibility problems with their app.

Implication 2: Accessibility as part of mobile apps maintenance and evolution. There exist accessibility testing tools and methods that are designed to support the implementation and testing phases of the software. However, there are no tools, to the best of our knowledge, that supports software accessibility in the maintenance phase. With changes made to an app, either for adding a feature or fixing a bug, accessibility can be at risk. Also, with updates made to the phone's operating system or the installed assistive technology, the accessibility of an app may deteriorate. We call for innovative methods that can support technology professionals in maintaining the accessibility of their app after its release. Our approach in analyzing app reviews offers an opportunity for developers and designers in detecting accessibility pitfalls based on their users' written feedback. However, with the tremendous number of reviews developers receive on a daily basis, it becomes impractical to manually read through them and identify potential issues related to their new release. Adding our model to the pipeline, will alleviate the manual overhead of looking up accessibility related reviews, and so developers can quickly locate their corresponding issues, and add them to their maintenance pipeline.

Implication 3: Understanding users' language in expressing their accessibility concerns. When we compared our BDTs-model to the keyword-based detector, we found that some accessibility reviews did not contain the accessibility keywords that were driven from accessibility guidelines [112]. This indicates that users voice their accessibility feedback using "user taxonomy" which may or may not echo the technical and professional terms used in accessibility standards. Further research is needed to understand how users describe mobile accessibility issues. By learning the accessibility "user taxonomy", we can improve our BDTs-model, which will lead to enhanced discovery of accessibility reviews.

Implication 4: The interplay between developers and designers, accessibility experts, and users. Accessibility experts establish guidelines and design methods in support of creating accessible software. Technology professionals often are not able to digest all these guidelines and often find existing resources lacking. This situation yielded to the existence of software products that are inaccessible to people with disabilities. The effective involvement of people with disabilities in this process can help bridging the communication gap between accessibility experts and developers and designers. By giving users the opportunity to lead the prioritization of accessibility issues based on their usage experience, mobile apps accessibility can be improved in a more meaningful way for people with disabilities. Analyzing app reviews is one way to give users the lead in determining which accessibility issue should be fixed in the next release. Analyzing app reviews can also offer insights to accessibility experts on users' accessibility needs right from the field, which will be more realistic than results collected from controlled lab studies.

Implication 5: Direct and immediate apps filtering benefit for end users. People find online reviews helpful in making purchase decisions [57]. Peer comments help users become aware of the limitations of reviewed products [217]. Currently, on mobile applications stores, e.g., App Store and Google Play, users can read all reviews, sort them by most helpful or most recent. However, mobile application stores provide no means to filtering reviews based on relevance to specific quality metrics, e.g., accessibility. This lack of filtering pushes users to download the app first and then experience its accessibility, leaving no room for benefiting from peer comments. Sometimes, apps suffer from accessibility regression giving users an unpleasant surprise with an updated app that is less accessible than its former version [299]. We call on mobile application stores to take action and allow users to filter reviews based on relevance to accessibility.

Implication 6: Pushing the boundaries of Accessibility testing. Current accessibility testing strategies are human intensive, and therefore become expensive and impractical, as most developers struggle to find the appropriate testers who can evaluate the compliance of their apps to accessibility guidelines. Existing accessibility scanners are tailored for the web, and they cannot be applied to the mobile environment. In this context, online user reviews, offer a rich source of scenarios, which can be coupled with the app's current version, to create test cases of practically captured anomalies. Relying on this set of reviews, as a shared knowledge, developers can quickly identify potential test cases that they need to perform, in case they are incorporating a given accessibility tool in their app. Furthermore, as the mobile environment is extremely dynamic, recent user reviews can quickly reveal any appearing anomalies in the newer app releases.

11.3. Chapter Summary

: In this chapter, we elaborated on how our study fits into the continuum of previously completed research and its significance for the area of contemporary research and practice.

In the next chapter, we identify potential threats to the validity of our approaches and experiments.

CHAPTER 12

THREATS TO VALIDITY

In this section, we identify potential threats to the validity of our approach and our experiments.

12.1. Internal Validity

The first limitation of the survey is the scope and appropriate selection of digital libraries. Therefore, we selected nine diverse electronic data sources. The next step was to ensure that the relevant literature publications were identified and included. We reasoned, though, that there might be other sources relevant within our domain search. Regardless, we attempted to mitigate this limitation as follows. We seeded a domain search with a set of search queries. If sufficient domain expertise is available, the search queries can be created manually; otherwise, a snowballing technique can be used [317, 316] in which a small number of initial search terms are used to retrieve a set of results, and then commonly occurring domain-specific phrases are identified and used to seed further search queries. We also employed an iterative strategy for our term-list construction. Different research communities might likely refer to the same concept or term differently. Hence, the iterative strategy ensured that adequate terms were used in the search process.

The second limitation is the validity of the constructed taxonomy. We reason whether the taxonomy has sufficient breadth and depth to ensure that accurate classification and systematic analysis are achieved within the deaf and hearing disability domain's scope. To mitigate this limitation, we employed a well-known content analysis method. In this case, the taxonomy was continuously filtered and evolved to account for every essential component of the paper included. This iterative process boosted our confidence that the taxonomy incurred substantially good coverage for the methods and types of disabilities that were included and examined throughout this literature review.

The third limitation refers to the objectiveness of the study. Typically this reflects on possible biases or flaws in the results. To mitigate this limitation, we have examined each reviewer's bias by cross-checking the papers. What that means is that no paper received only one reviewer. Thus multiple reviewers were involved in the process. Furthermore, we have also obtained the summary of the conclusions according to a collection of categorized papers, rather than following only individual reviewers' interpretations or views with one goal only to avoid bias.

12.2. Construct Validity

Threats to the validity relate to the appropriateness of our dataset and accuracy of the previous work [112]. A potential threat is related to creating a training dataset or the manual classification. Developing a training dataset is typically a tedious job, also subject to reader bias. We mitigated this risk by choosing a dataset of accessibility reviews as our training data that were previously identified and validated [112]. Additionally, we used all of the identified reviews as training input rather than choosing a sample set of reviews. A total of 2,663 reviews were previously identified as accessibility reviews from 214,053 app reviews through manual inspections and validations.

Another potential threat relates to the keywords used for the identification of accessibility reviews through a string-matching approach. The string-matching approach relied on 213 keywords derived from 54 accessibility recommendations by BBC. The keywords and phrases users use in their reviews do not necessarily match the keywords available in the guidelines and recommendations. This mismatch includes but not limited to situations when keywords would be spelled incorrectly by reviewers. A related concern is whether the set of keywords is inclusive of all possible keywords that users use to express their accessibility concerns. To mitigate this threat, we used keywords defined by [112] in which the authors adopted variants for these keywords to ensure they would not miss any relevant review during their manual validation. This raised our confidence to use the dataset that has these keywords as a representative sample of accessibility reviews.

12.3. External Validity

Threats to the validity relate to the generalizability of our findings for this evaluation. We evaluated and tested our findings on a dataset collected by previous researchers [112]. The dataset was collected only from Android open-source applications. Therefore, the dataset did not represent the entire mobile apps on the App stores such as Apple store applications. Also, we only study mobile application reviews of open-source applications. Our results may not generalize to commercially developed projects or to other reviews that are written in other languages than English.

12.4. Chapter Summary

: In this chapter, we identified potential threats to the validity of our approaches and experiments.

In the next chapter, we provide a summary as well as recommendations for potential future study paths.

CHAPTER 13

CONCLUSION

In this thesis, we propose an approach that automates the classification of app reviews and bug reports as accessibility-related or not so developers can easily detect accessibility issues with their products and improve them to more accessible and inclusive apps utilizing the users' input. As Hayes pointed out:In Action Research, the goal is ultimately to create sustainable change. That is to say, once the research facilitators leave, the community partners should be able to maintain the positive changes that have been made [139]. Our goal is to create a sustainable change, by including a model in developer's software maintenance pipeline, and raising awareness of existing errors that hinders the accessibility of mobile apps, which is a pressing need [244]. As we develop our model, we conducted an evaluation of various different classifiers using an existing dataset of manually validated accessibility reviews. The results indicate that our approach outperforms the two state-of-the-art approaches with the F1-measure of 90.7%.

In addition, we presented an automated approach for classifying accessibility app reviews in four categories, i.e., Principles, Audio/Video, Design, and Focus, for helping the developers detect app issues and performance improvement by considering user reviews. We also automated sentiment analysis-based approach for the classification of accessibility-related app reviews to help the developers detect these issues and improve their app's performance in light of user's reviews. An existing dataset that comprises manually validated accessibility app reviews has been employed in our work. We employed six classification models, namely Extra Tree Classifier, Random Forest, Support Vector Classification, Decision Tree, K-Nearest Neighbors, and Logistic Regression. To evaluate their performance, we used four classification metrics, i.e., Accuracy, Precision, Recall, and F1-Score for measuring their performance. Evaluation results have shown that KNN exhibits the least accuracy while the ETC model outperformed other models in overall accuracy with TF-IDF features. In the future, we intend to increase the keywords and sample size to improve the selection and analysis process of accessibility reviews and provide a mechanism to check whether the developers have addressed the users' concerns in the subsequent releases by implementing the required features.

Furthermore, understanding the challenges that deaf students faced during the COVID-19 period is of paramount importance to the deaf community. In this thesis, we also investigate the e-learning experience of 65 deaf students by focusing on the Technical and Vocational Training Corporation (TVTC) in Saudi Arabia. Due to the closure of physical classes, online learning using several devices in synchronous (live) and asynchronous (pre-recorded) environments become an alternative learning method. However, this alternative learning method becomes challenging to the deaf students due to the limited resources and accessibility to online learning. We found that: (1) Blackboard as well as the course material are not easily accessible to the deaf students, (2) deaf students find that learning is extremely stressful during the pandemic, (3) Google Meet is the preferable e-learning tool, (4) communication between deaf students and teachers is not effective which impacts the learning outcomes, (5) lack of support in terms of the provided interpreters hinder the learning process, and (6) technology is not always enhanced for people who are deaf or hard of hearing. This early contribution of the present work opens an opportunity for the research community and the educational sector to address these needs broadly and globally with similar interest and care. Additionally, our work directly contributes to the literature by providing a detailed analysis of online learning challenges for deaf and hard-of-hearing students. Most critically, it brings forward attention to recommending educational systems to be more accessible during pandemic crises and leverage teaching strategies that can be easily incorporated even in the face of environmental crisis.

Recommendations for Future Research

- Gaps between developer and user perception. The preliminary indication from this thesis is that developers and users have different perceptions as to the impact of accessibility issues on the usability of apps. Bridging this gap would help developers prioritize fixes for accessibility issues that are most critical for users first. However, in order to do a meaningful comparison and have a better understating of user perspective, a more extensive user study, involving disabled users, would be needed. Given that property conducting such a study would require, among others, access to users with different types of disability (e.g., visual, hearing, mobility impairment).

- Future researchers may investigate the techniques of refining LMS systems to improve their accessibility. Such a proposition is made because this study has established that many deaf students are unable to fully take advantage of LMS systems [21]. Potentially, scholars may look at improving the functionality of such systems, customizing them to meet the needs of individual students, and simplifying their navigation.
- Scholars can investigate how mental health issues among deaf students can be mitigated. It is apparent that deaf students are facing a hard time during the pandemic, and the inability to cope can lead to stress. Researchers can investigate the possible ways of addressing the educational and socioeconomic factors that should be addressed so that such students have peace of mind and better mental health outcomes.
- It would also be important to explore how stakeholder engagement can be improved to harness their efforts to help deaf students. It has been established in this study that the roles of various stakeholders are very important in ensuring quality education for the deaf. Other scholars may utilize stakeholder engagement models and frameworks to explain how such stakeholders can work with each other collaboratively so that the learning outcomes of deaf students can be achieved.
- Since we have established a challenge relating to learning materials, subsequent studies could investigate the factors that lead to their inadequacy. For example, it would be important to establish whether institutions get enough funding from the government to purchase materials and other resources that are needed to educate deaf students. In addition, the maintenance of such materials, ensuring efficient and equitable use, as well as their administration, should be evaluated.

- Future researchers can investigate the issues facing other sections of the deaf population, such as immigrants, or across different age groups. It has been established that there are wide disparities in learning and education outcomes between such groups and the rest of the population. Given that being deaf also comes with unique challenges, it would be important to understand how the intersectionality between social disadvantage and deafness affects deaf students. For instance, it would be prudent to explore the challenges that deaf students from poor backgrounds face during the pandemic.

REFERENCES

[1] *Accessibility dataset.*, https://doi.org/10.5281/zenodo.5540624, 2021.

[2] *Survey questions.*, https://ufile.io/n41g2rn8, 2021.

[3] *Replication dataset.*, https://doi.org/10.5281/zenodo.6678309, 2022.

[4] Muhammad Adeel Abid, Saleem Ullah, Muhammad Abubakar Siddique, Muhammad Faheem Mushtaq, Wajdi Aljedaani, and Furqan Rustam, *Spam sms filtering based on text features and supervised machine learning techniques*, Multimedia Tools and Applications 81 (2022), no. 28, 39853–39871.

[5] Gulmira M Abildinova, Aitugan K Alzhanov, Nazira N Ospanova, Zhymatay Taybaldieva, Dametken S Baigojanova, and Nikita O Pashovkin, *Developing a mobile application" educational process remote management system" on the android operating system.*, International Journal of Environmental and Science Education 11 (2016), no. 12, 5128–5145.

[6] ADA, *Americans with disabilities act (ada).*, https://www.ada.gov/ada_title_III.htm, February 2022.

[7] Chuck Adams, Alastair Campbell, Rachael Montgomery, Michael Cooper, and Andrew Kirkpatrick, *Web content accessibility guidelines (wcag) 2.2*, WWW Consortium (W3C)(2020) (2020).

[8] Olufemi Timothy Adigun, *The experiences of emergency-remote teaching via zoom: The case of natural-science teachers handling of deaf/hard-of-hearing learners in south africa*, International Journal of Learning, Teaching and Educational Research 21 (2022), no. 2.

[9] Gaurav Agrawal, Devendra Kumar, Mayank Singh, and Diksha Dani, *Evaluating accessibility and usability of airline websites*, International Conference on Advances in Computing and Data Sciences, Springer, 2019, pp. 392–402.

[10] Maha Al-shammari, Asma Ashankyty, Najmah Al-Mowina, Nadia Al-Mutairy, Lulwah

Al-shammari, Susan Amin, et al., *Social-emotional perceptions of deaf students in hail, saudi arabia*, American Journal of Educational Research 2 (2014), no. 5, 304–315.

[11] Ahmad M Al-Shomar, Muhammad Al-Qurish, and Wajdi Aljedaani, *A novel framework for remote management of social media big data analytics*, Social Network Analysis and Mining 12 (2022), no. 1, 172.

[12] Afnan A Al-Subaihin, Atheer S Al-Khalifa, and Hend S Al-Khalifa, *Accessibility of mobile web apps by screen readers of touch-based mobile phones*, International Conference on Mobile Web and Information Systems, Springer, 2013, pp. 35–43.

[13] Ayshah Alahmari, *The state of distance education in saudi arabia*, Quarterly Review of Distance Education 18 (2017), no. 2, 91–98.

[14] Khadijah Alaidarous and Abeer Ahmed Madini, *Exploring efl students' perception in blended learning environment in saudi technical education context*, International Journal of Educational Investigations 3 (2016), no. 6, 69–81.

[15] Victor John Levi L Alcazar, Abdullah Nasser M Maulana, Raymon O Mortega, and Mary Jane C Samonte, *Speech-to-visual approach e-learning system for the deaf*, 2016 11th International Conference on Computer Science & Education (ICCSE), IEEE, 2016, pp. 239–243.

[16] Abdulaziz Aldiab, Harun Chowdhury, Alex Kootsookos, Firoz Alam, and Hamed Allhibi, *Utilization of learning management systems (lmss) in higher education system: A case review for saudi arabia*, Energy Procedia 160 (2019), 731–737.

[17] R Alebaikan, *A blended learning framework for saudi higher education*, A paper Presented at the Second International Conference of E-Learning and Distance Learning, Riyadh: National Center for E-Learning and Distance Learning, 2011.

[18] Abdulsalam K Alhazmi, Athar Imtiaz, Fatima Al-Hammadi, and Ezzadeen Kaed, *Success and failure aspects of lms in e-learning systems.*, International Journal of Interactive Mobile Technologies 15 (2021), no. 11.

[19] Jehan AlHumaid, Saqib Ali, and Imran Farooq, *The psychological effects of the covid-*

19 pandemic and coping with them in saudi arabia., Psychological Trauma: Theory, Research, Practice, and Policy 12 (2020), no. 5, 505.

[20] Jamal Kaid Mohammed Ali, *Blackboard as a motivator for saudi efl students: A psycholinguistic study*, International Journal of English Linguistics 7 (2017), no. 5, 144–151.

[21] Wajdi Aljedaani, Mona Aljedaani, Eman Abdullah AlOmar, Mohamed Wiem Mkaouer, Stephanie Ludi, and Yousef Bani Khalaf, *I cannot see you—the perspectives of deaf students to online learning during covid-19 pandemic: Saudi arabia case study*, Education Sciences 11 (2021), no. 11, 712.

[22] Wajdi Aljedaani, Mona Aljedaani, Mohamed Wiem Mkaouer, and Stephanie Ludi, *Teachers perspectives on transition to online teaching deaf and hard-of-hearing students during the covid-19 pandemic: A case study*, 16th Innovations in Software Engineering Conference (formerly known as India Software Engineering Conference), 2023.

[23] Wajdi Aljedaani and Yasir Javed, *Bug reports evolution in open source systems*, 5th International Symposium on Data Mining Applications, Springer, 2018, pp. 63–73.

[24] ———, *Empirical study of software test suite evolution*, 2020 6th Conference on Data Science and Machine Learning Applications (CDMA), IEEE, 2020, pp. 87–93.

[25] Wajdi Aljedaani, Yasir Javed, and Mamdouh Alenezi, *Lda categorization of security bug reports in chromium projects*, Proceedings of the 2020 European Symposium on Software Engineering, 2020, pp. 154–161.

[26] ———, *Open source systems bug reports: meta-analysis*, Proceedings of the 2020 The 3rd International Conference on Big Data and Education, 2020, pp. 43–49.

[27] Wajdi Aljedaani, Rrezarta Krasniqi, Sanaa Aljedaani, Mohamed Wiem Mkaouer, Stephanie Ludi, and Khaled Al-Raddah, *If online learning works for you, what about deaf students? emerging challenges of online learning for deaf and hearing-impaired students during covid-19: a literature review*, Universal access in the information society (2022), 1–20.

[28] Wajdi Aljedaani, Mohamed Wiem Mkaouer, Stephanie Ludi, and Yasir Javed, *Auto-*

matic classification of accessibility user reviews in android apps, 2022 7th International Conference on Data Science and Machine Learning Applications (CDMA), IEEE, 2022, pp. 133–138.

[29] Wajdi Aljedaani, Mohamed Wiem Mkaouer, Stephanie Ludi, Ali Ouni, and Ilyes Jenhani, *On the identification of accessibility bug reports in open source systems*, Proceedings of the 19th International Web for All Conference, 2022, pp. 1–11.

[30] Wajdi Aljedaani, Mohamed Wiem Mkaouer, Anthony Peruma, and Stephanie Ludi, *Do the test smells assertion roulette and eager test impact students' troubleshooting and debugging capabilities?*, Proceedings of the ACM/IEEE 45nd International Conference on Software Engineering: Software Engineering Education and Training, 2023.

[31] Wajdi Aljedaani, Meiyappan Nagappan, Bram Adams, and Michael Godfrey, *A comparison of bugs across the ios and android platforms of two open source cross platform browser apps*, 2019 IEEE/ACM 6th International Conference on Mobile Software Engineering and Systems (MOBILESoft), IEEE, 2019, pp. 76–86.

[32] Wajdi Aljedaani, Anthony Peruma, Ahmed Aljohani, Mazen Alotaibi, Mohamed Wiem Mkaouer, Ali Ouni, Christian D Newman, Abdullatif Ghallab, and Stephanie Ludi, *Test smell detection tools: A systematic mapping study*, Evaluation and Assessment in Software Engineering (2021), 170–180.

[33] Wajdi Aljedaani, Furqan Rustam, Stephanie Ludi, Ali Ouni, and Mohamed Wiem Mkaouer, *Learning sentiment analysis for accessibility user reviews*, 2021 36th IEEE/ACM International Conference on Automated Software Engineering Workshops (ASEW), IEEE, 2021, pp. 239–246.

[34] Wajdi Aljedaani, Furqan Rustam, Mohamed Wiem Mkaouer, Abdullatif Ghallab, Vaibhav Rupapara, Patrick Bernard Washington, Ernesto Lee, and Imran Ashraf, *Sentiment analysis on twitter data integrating textblob and deep learning models: The case of us airline industry*, Knowledge-Based Systems 255 (2022), 109780.

[35] Wajdi Aljedaani, Eysha Saad, Furqan Rustam, Isabel de la Torre Díez, and Imran

Ashraf, *Role of artificial intelligence for analysis of covid-19 vaccination-related tweets: Opportunities, challenges, and future trends*, Mathematics 10 (2022), no. 17, 3199.

[36] Ayman N Alkhaldi and Abdallah M Abualkishik, *The mobile blackboard system in higher education: Discovering benefits and challenges facing students*, International Journal of Advanced and Applied Sciences 6 (2019), no. 6, 6–14.

[37] Bader Alkhazi, Andrew DiStasi, Wajdi Aljedaani, Hussein Alrubaye, Xin Ye, and Mohamed Wiem Mkaouer, *Learning to rank developers for bug report assignment*, Applied Soft Computing 95 (2020), 106667.

[38] Sultan Saleh Ahmed Almekhlafy, *Online learning of english language courses via blackboard at saudi universities in the era of covid-19: perception and use*, PSU Research Review (2020).

[39] Abdullah S Alofi, M Diane Clark, Amber E Marchut, et al., *Life stories of saudi deaf individuals*, Psychology 10 (2019), no. 11, 1506.

[40] Eman Abdullah AlOmar, Wajdi Aljedaani, Murtaza Tamjeed, Mohamed Wiem Mkaouer, and Yasmine N El-Glaly, *Finding the needle in a haystack: On the automatic identification of accessibility user reviews*, Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, 2021, pp. 1–15.

[41] Eman Abdullah AlOmar, Mohamed Wiem Mkaouer, and Ali Ouni, *Can refactoring be self-affirmed? an exploratory study on how developers document their refactoring activities in commit messages*, 2019 IEEE/ACM 3rd International Workshop on Refactoring (IWoR), 2019, pp. 51–58.

[42] ———, *Toward the automatic classification of self-affirmed refactoring*, Journal of Systems and Software 171 (2020), 110821.

[43] Eman Abdullah AlOmar, Anthony Peruma, Mohamed Wiem Mkaouer, Christian Newman, Ali Ouni, and Marouane Kessentini, *How we refactor and how we document it? on the use of supervised machine learning algorithms to classify refactoring documentation*, Expert Systems with Applications (2020), 114176.

[44] Faisl M Alqraini and Khalid N Alasim, *Distance education for d/deaf and hard of*

hearing students during the covid-19 pandemic in saudi arabia: Challenges and support, Research in Developmental Disabilities 117 (2021), 104059.

[45] Abdulmalik Alqurshi, *Investigating the impact of covid-19 lockdown on pharmaceutical education in saudi arabia–a call for a remote teaching contingency strategy*, Saudi Pharmaceutical Journal 28 (2020), no. 9, 1075–1083.

[46] Elham Alsadoon and Maryam Turkestani, *Virtual classrooms for hearing-impaired students during the covid-19 pandemic.*, Romanian Journal for Multidimensional Education/Revista Romaneasca pentru Educatie Multidimensionala 12 (2020).

[47] Abdallah A Alshawabkeh, M Lynn Woolsey, and Faten F Kharbat, *Using online information technology for deaf students during covid-19: A closer look from experience*, Heliyon 7 (2021), no. 5, e06915.

[48] Abdulaziz Alshayban, Iftekhar Ahmed, and Sam Malek, *Accessibility issues in android apps: state of affairs, sentiments, and ways forward*, 2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE), IEEE, 2020, pp. 1323–1334.

[49] Yasser Ali Alshehri, Najwa Mordhah, Sameer Alsibiani, Samir Alsobhi, and Noha Alnazzawi, *How the regular teaching converted to fully online teaching in saudi arabia during the coronavirus covid-19*, Creative Education 11 (2020), no. 7, 985–996.

[50] Dalal Alsindi et al., *Optimizing online learning experiences and outcomes for hearing-impaired art and design students*, International Journal of Learning, Teaching and Educational Research 20 (2021), no. 7.

[51] Uthman T Alturki, Ahmed Aldraiweesh, et al., *Evaluating the usability and accessibility of lms "blackboard" at king saud university*, Contemporary Issues in Education Research (CIER) 9 (2016), no. 1, 33–44.

[52] Aashir Amaar, Wajdi Aljedaani, Furqan Rustam, Saleem Ullah, Vaibhav Rupapara, and Stephanie Ludi, *Detection of fake job postings by utilizing machine learning and natural language processing approaches*, Neural Processing Letters (2022), 1–29.

[53] Galen Andrew and Jianfeng Gao, *Scalable training of l 1-regularized log-linear models*, Proceedings of the 24th international conference on Machine learning, 2007, pp. 33–40.

[54] Samina Ashraf, Musarrat Jahan, and Muhammad Saad, *Educating students with hearing impairment during covid-19 pandemic: A case of inclusive and special schools*, Review of Applied Management and Social Sciences 4 (2021), no. 4, 783–794.

[55] Comfort Atanga, Beth A Jones, Lacy E Krueger, and Shulan Lu, *Teachers of students with learning disabilities: Assistive technology knowledge, perceptions, interests, and barriers*, Journal of Special Education Technology 35 (2020), no. 4, 236–248.

[56] Microsoft Azure, *Azure machine learning*, 2020, Library Catalog: azure.microsoft.com.

[57] Hyunmi Baek, JoongHo Ahn, and Youngseok Choi, *Helpfulness of online consumer reviews: Readers' objectives and review cues*, International Journal of Electronic Commerce 17 (2012), no. 2, 99–126.

[58] Elsa Bakiu and Emitza Guzman, *Which feature is unusable? detecting usability and user experience issues from user reviews*, 2017 IEEE 25th International Requirements Engineering Conference Workshops (REW), IEEE, 2017, pp. 182–187.

[59] Mars Ballantyne, Archit Jha, Anna Jacobsen, J Scott Hawker, and Yasmine N El-Glaly, *Study of accessibility guidelines of mobile applications*, Proceedings of the 17th international conference on mobile and ubiquitous multimedia, 2018, pp. 305–315.

[60] Concha Batanero, Luis de Marcos, Jaana Holvikivi, José Ramón Hilera, and Salvador Otón, *Effects of new supportive technologies for blind and deaf engineering students in online learning*, IEEE Transactions on Education 62 (2019), no. 4, 270–277.

[61] Concepción Batanero-Ochaíta, Luis De-Marcos, Luis Felipe Rivera, Jaana Holvikivi, José Ramón Hilera, and Salvador Otón Tortosa, *Improving accessibility in online education: Comparative analysis of attitudes of blind and deaf students toward an adapted learning platform*, IEEE Access 9 (2021), 99968–99982.

[62] BBC, *The BBC Standards and Guidelines for Mobile Accessibility*, 2017.

[63] ———, *The bbc standards and guidelines for mobile accessibility*, https://www.bbc.co.uk/guidelines/futuremedia/accessibility/mobile, June 2020.

[64] Izak Benbasat, David K Goldstein, and Melissa Mead, *The case research strategy in studies of information systems*, MIS quarterly (1987), 369–386.

[65] Kristin P Bennett and Colin Campbell, *Support vector machines: hype or hallelujah?*, ACM SIGKDD explorations newsletter 2 (2000), no. 2, 1–13.

[66] Larwan Berke, Matthew Seita, and Matt Huenerfauth, *Deaf and hard-of-hearing users' prioritization of genres of online video content requiring accurate captions*, Proceedings of the 17th International Web for All Conference, 2020, pp. 1–12.

[67] Serenella Besio, Elena Laudanna, Francesca Potenza, Lucia Ferlino, and Federico Occhionero, *Accessibility of educational software: from evaluation to design guidelines*, International Conference on Computers for Handicapped Persons, Springer, 2008, pp. 518–525.

[68] Nigel Bevan, James Carter, and Susan Harker, *Iso 9241-11 revised: What have we learnt about usability since 1998?*, International Conference on Human-Computer Interaction, Springer, 2015, pp. 143–151.

[69] Pamela Bhattacharya, Liudmila Ulanova, Iulian Neamtiu, and Sai Charan Koduru, *An empirical analysis of bug reports and bug fixing in open source android apps*, 2013 17th European Conference on Software Maintenance and Reengineering, IEEE, 2013, pp. 133–143.

[70] Blackboard, *Blackboard learn: An advanced lms.*, https://www.blackboard.com/en-eu/teaching-learning/learning-management/blackboard-learn, October 2021.

[71] Dan German Blazer, *Hearing loss and psychiatric disorders*, 2020.

[72] Felipe F Bocca and Luiz Henrique Antunes Rodrigues, *The effect of tuning, feature engineering, and feature selection in data mining applied to rainfed sugarcane yield modelling*, Computers and electronics in agriculture 128 (2016), 67–76.

[73] Venkateswarlu Bonta and Nandhini Kumaresh2and N Janardhan, *A comprehensive study on lexicon based approaches for sentiment analysis*, Asian Journal of Computer Science and Technology 8 (2019), no. S2, 1–6.

[74] Richard E Boyatzis, *Transforming qualitative information: Thematic analysis and code development*, sage, 1998.

[75] Virginia Braun and Victoria Clarke, *Using thematic analysis in psychology*, Qualitative research in psychology 3 (2006), no. 2, 77–101.

[76] Mary E Brenner, *Interviewing in educational research*, Handbook of complementary methods in education research 2 (2006).

[77] Mr Brijain, R Patel, MR Kushik, and K Rana, *A survey on decision tree algorithm for classification*, (2014).

[78] Peter Brunet, Barry Alan Feigenbaum, Kip Harris, Catherine Laws, R Schwerdtfeger, and Lawrence Weiss, *Accessibility requirements for systems design to accommodate users with vision impairments*, IBM Systems Journal 44 (2005), no. 3, 445–466.

[79] Bugzilla, *Bugzilla issue tracker)*, https://www.bugzilla.org/, June 2020.

[80] Sheryl Burgstahler, *Designing software that is accessible to individuals with disabilities*, Retrieved March 24 (2008), 2009.

[81] ———, *Universal design: Implications for computing education*, ACM Transactions on Computing Education (TOCE) 11 (2011), no. 3, 1–17.

[82] ———, *Opening doors or slamming them shut? online learning practices and students with disabilities*, Social Inclusion 3 (2015), no. 6, 69–79.

[83] Laura V. Galvis Carreno and Kristina Winbladh, *Analysis of user comments: An approach for software requirements evolution*, 2013 35th International Conference on Software Engineering (ICSE) (San Francisco, CA, USA), IEEE, May 2013, pp. 582–591 (en).

[84] Helena Carla Castro, Lins Ramos AS, Gildete Amorim, and Norman Arthur Ratcliffe, *Covid-19: don't forget deaf people.*, Nature 579 (2020), no. 7799, 343–343.

[85] Jackjun Caupayan and Angeline Pogoy, *Unheard stories of deaf students in online learning: A phenomenological study*, Available at SSRN 3856136 (2021).

[86] Darren Chadwick and Caroline Wesson, *Digital inclusion and disability*, Applied cyberpsychology, Springer, 2016, pp. 1–23.

[87] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer,

Smote: synthetic minority over-sampling technique, Journal of artificial intelligence research 16 (2002), 321–357.

[88] Sarah Chiti and Barbara Leporini, *Accessibility of android-based mobile devices: a prototype to investigate interaction with blind users*, International Conference on Computers for Handicapped Persons, Springer, 2012, pp. 607–614.

[89] Witold Chmielarz, *The usage of smartphone and mobile applications from the point of view of customers in poland*, Information 11 (2020), no. 4, 220.

[90] Gobinda G Chowdhury, *Natural language processing*, Annual review of information science and technology 37 (2003), no. 1, 51–89.

[91] Adelina Ciurumelea, Andreas Schaufelbuhl, Sebastiano Panichella, and Harald C. Gall, *Analyzing reviews and code of mobile apps for better release planning*, 2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER) (Klagenfurt, Austria), IEEE, February 2017, pp. 91–102 (en).

[92] Adelina Ciurumelea, Andreas Schaufelbühl, Sebastiano Panichella, and Harald C Gall, *Analyzing reviews and code of mobile apps for better release planning*, 2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER), IEEE, 2017, pp. 91–102.

[93] Jordan Cleminson, *A thematic analysis of a photo elicitation investigating 'what does it mean to a person to be deaf or hard of hearing?'*, Journal of Applied Psychology and Social Science 5 (2019), no. 1, 1–30.

[94] Jacob Cohen, *A coefficient of agreement for nominal scales*, Educational and psychological measurement 20 (1960), no. 1, 37–46.

[95] Michael Collins, *Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms*, Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10, Association for Computational Linguistics, 2002, pp. 1–8.

[96] Claudiu Coman, Laurențiu Gabriel Țîru, Luiza Meseșan-Schmitz, Carmen Stanciu, and

Maria Cristina Bularca, *Online teaching and learning in higher education during the coronavirus pandemic: students' perspective*, Sustainability 12 (2020), no. 24, 10367.

[97] Lynne M Connelly, *Pilot studies*, Medsurg Nursing 17 (2008), no. 6, 411.

[98] Steve Cook, Duncan Watson, and Dimitrios Vougas, *Solving the quantitative skills gap: a flexible learning call to arms!*, Higher Education Pedagogies 4 (2019), no. 1, 17–31.

[99] Elisabeth A Counselman Carpenter, Ariel Meltzer, and Matthea Marquart, *Best practices for inclusivity of deaf/deaf/hard of hearing students in the synchronous online classroom.*, World Journal of Education 10 (2020), no. 4, 26–34.

[100] Michael Crabb, Michael Heron, Rhianne Jones, Mike Armstrong, Hayley Reid, and Amy Wilson, *Developing accessible services: Understanding current knowledge and areas for future support*, Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (New York, NY, USA), CHI '19, Association for Computing Machinery, 2019, p. 1–12.

[101] Cláudia Ferreira da Silva, Simone B Leal Ferreira, and Carolina Sacramento, *Mobile application accessibility in the context of visually impaired users*, Proceedings of the 17th Brazilian Symposium on Human Factors in Computing Systems, 2018, pp. 1–10.

[102] Talal Daghriri, Furqan Rustam, Wajdi Aljedaani, Abdullateef H Bashiri, and Imran Ashraf, *Electroencephalogram signals for detecting confused students in online education platforms with probability-based features*, Electronics 11 (2022), no. 18, 2855.

[103] Niely Silva de Souza, Alessandra Marcone Tavares Alves de Figueirêdo, Carlos Alberto da Silva, Júlia Maria Soares Ferraz Júnior, and Márcio Jean Fernandes Tavares, *Inclusive teaching in organic chemistry: A visual approach in the time of covid-19 for deaf students*, International Journal for Innovation Education and Research (2022), 290–306.

[104] Zhenyun Deng, Xiaoshu Zhu, Debo Cheng, Ming Zong, and Shichao Zhang, *Efficient knn classification algorithm for big data*, Neurocomputing 195 (2016), 143–148.

[105] Andrea Di Sorbo, Sebastiano Panichella, Carol V. Alexandru, Corrado A. Visaggio, and Gerardo Canfora, *SURF: Summarizer of User Reviews Feedback*, 2017 IEEE/ACM

39th International Conference on Software Engineering Companion (ICSE-C) (Buenos Aires), IEEE, May 2017, pp. 55–58 (en).

[106] José-Manuel Díaz-Bossini and Lourdes Moreno, *Accessibility to mobile interfaces for older people*, Procedia Computer Science 27 (2014), 57–66.

[107] Trinidad Domínguez Vila, Elisa Alén González, and Simon Darcy, *Website accessibility in the tourism industry: an analysis of official national tourism organization websites around the world*, Disability and rehabilitation 40 (2018), no. 24, 2895–2906.

[108] Li Duan and Gang Zhu, *Psychological interventions for people affected by the covid-19 epidemic*, The Lancet Psychiatry 7 (2020), no. 4, 300–302.

[109] Horatiu Dumitru, Marek Gibiec, Negar Hariri, Jane Cleland-Huang, Bamshad Mobasher, Carlos Castro-Herrera, and Mehdi Mirakhorli, *On-demand feature recommendations derived from mining public product descriptions*, Proceeding of the 33rd international conference on Software engineering - ICSE '11 (Waikiki, Honolulu, HI, USA), ACM Press, 2011, p. 181 (en).

[110] Shahul H Ebrahim and Ziad A Memish, *Saudi arabia's drastic measures to curb the covid-19 outbreak: temporary suspension of the umrah pilgrimage*, Journal of Travel Medicine 27 (2020), no. 3, taaa029.

[111] Yasmine N El-Glaly, Anthony Peruma, Daniel E Krutz, and J Scott Hawker, *Apps for everyone: mobile accessibility learning modules*, ACM Inroads 9 (2018), no. 2, 30–33.

[112] Marcelo Medeiros Eler, Leandro Orlandin, and Alberto Dumont Alves Oliveira, *Do Android app users care about accessibility?: an analysis of user reviews on the Google play store*, Proceedings of the 18th Brazilian Symposium on Human Factors in Computing Systems (Vitória Espírito Santo Brazil), ACM, October 2019, pp. 1–11 (en).

[113] ———, *Do android app users care about accessibility? an analysis of user reviews on the google play store*, Proceedings of the 18th Brazilian Symposium on Human Factors in Computing Systems, 2019, pp. 1–11.

[114] Marcelo Medeiros Eler, Jose Miguel Rojas, Yan Ge, and Gordon Fraser, *Automated Accessibility Testing of Mobile Apps*, 2018 IEEE 11th International Conference on

Software Testing, Verification and Validation (ICST) (Vasteras), IEEE, April 2018, pp. 116–126 (en).

[115] Menna Elkhateeb, Abdulaziz Shehab, and Hazem El-Bakry, *Mobile learning system for egyptian higher education using agile-based approach*, Education Research International 2019 (2019).

[116] Shahnoor C Eshan and Mohammad S Hasan, *An application of machine learning to detect abusive bengali text*, 2017 20th International Conference of Computer and Information Technology (ICCIT), IEEE, 2017, pp. 1–6.

[117] Fan Fang, John Wu, Yanyan Li, Xin Ye, Wajdi Aljedaani, and Mohamed Wiem Mkaouer, *On the classification of bug reports to improve bug localization*, Soft Computing 25 (2021), no. 11, 7307–7323.

[118] Tarah H Fatani, *Student satisfaction with videoconferencing teaching quality during the covid-19 pandemic*, BMC Medical Education 20 (2020), no. 1, 1–8.

[119] Reno Fernandes, Nora Susilawati, Rila Muspita, Eka Vidya Putra, Emizal Amri, Amin Akbar, and Aprizon Putra, *Voter education for the deaf during the covid–19 pandemic*, PalArch's Journal of Archaeology of Egypt/Egyptology 17 (2020), no. 6, 10518–10528.

[120] Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim, *Do we need hundreds of classifiers to solve real world classification problems?*, The journal of machine learning research 15 (2014), no. 1, 3133–3181.

[121] Joseph L Fleiss, Bruce Levin, Myunghee Cho Paik, et al., *The measurement of interrater agreement*, Statistical methods for rates and proportions 2 (1981), no. 212-236, 22–23.

[122] Tracey Forman, *Clear communication helps with transition to online learning.*, Disability Compliance for Higher Education 25 (2020), no. 11, 2–2.

[123] Jerome H Friedman, *Greedy function approximation: a gradient boosting machine*, Annals of statistics (2001), 1189–1232.

[124] Bharat Gaind, Varun Syal, and Sneha Padgalwar, *Emotion detection and analysis on social media*, arXiv preprint arXiv:1901.08458 (2019).

[125] Mikel Galar, Alberto Fernandez, Edurne Barrenechea, Humberto Bustince, and Francisco Herrera, *A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches*, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42 (2011), no. 4, 463–484.

[126] Abdullatif Ghallab, Ali Almuzaiqer, Abdullah Al-Hashedi, Abdulqader Mohsen, Kamal Bechkoum, and Wajdi Aljedaani, *Factors affecting intention to adopt open source erp systems by smes in yemen*, 2021 International Conference on Intelligent Technology, System and Service for Internet of Everything (ITSS-IoE), IEEE, 2021, pp. 1–7.

[127] Anwar Ghammam, Thiago Ferreira, Wajdi Aljedaani, Marouane Kessentini, and Ali Husain, *Dynamic software containers workload balancing via many-objective search*, IEEE Transactions on Services Computing (2023).

[128] Cole Gleason, Stephanie Valencia, Lynn Kirabo, Jason Wu, Anhong Guo, Elizabeth Jeanne Carter, Jeffrey Bigham, Cynthia Bennett, and Amy Pavel, *Disability and the covid-19 pandemic: Using twitter to understand accessibility during rapid societal transition*, The 22nd International ACM SIGACCESS Conference on Computers and Accessibility, 2020, pp. 1–14.

[129] Ramiro Gonçalves, Frederico Branco, António Pereira, Manuel Au-Yong-Oliveira, José Martins, et al., *Accessible software development: a conceptual model proposal*, Universal Access in the Information Society 18 (2019), no. 3, 703–716.

[130] Alice Goodenough and Sue Waite, *Real world research: a resource for users of social research methods in applied settings*, 2012.

[131] Google, *Google play store*, https://play.google.com/store?hl=en_US&gl=US, October 2021.

[132] Katerina Goseva-Popstojanova and Jacob Tyo, *Identification of security related bug reports via text mining using supervised and unsupervised classification*, 2018 IEEE International Conference on Software Quality, Reliability and Security (QRS), IEEE, 2018, pp. 344–355.

[133] Guo Haixiang, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Gong

Bing, *Learning from class-imbalanced data: Review of methods and applications*, Expert Systems with Applications 73 (2017), 220–239.

[134] Astri Hanjarwati, Jamil Suprihatiningrum, et al., *Is online learning accessible during covid-19 pandemic? voices and experiences of uin sunan kalijaga students with disabilities*, Nadwa 14 (2020), no. 1, 1–38.

[135] Lars Kai Hansen and Peter Salamon, *Neural network ensembles*, IEEE Transactions on Pattern Analysis & Machine Intelligence (1990), no. 10, 993–1001.

[136] Negar Hariri, Carlos Castro-Herrera, Mehdi Mirakhorli, Jane Cleland-Huang, and Bamshad Mobasher, *Supporting Domain Analysis through Mining and Recommending Features from Online Product Listings*, IEEE Transactions on Software Engineering 39 (2013), no. 12, 1736–1752 (en).

[137] M. Harman, Yue Jia, and Yuanyuan Zhang, *App store mining and analysis: MSR for app stores*, 2012 9th IEEE Working Conference on Mining Software Repositories (MSR) (Zurich), IEEE, June 2012, pp. 108–111 (en).

[138] Tonmoy Hasan and Abdul Matin, *Extract sentiment from customer reviews: A better approach of tf-idf and bow-based text classification using n-gram technique*, Proceedings of International Joint Conference on Advances in Computational Intelligence, Springer, 2021, pp. 231–244.

[139] Gillian R Hayes, *The relationship of action research to human-computer interaction*, ACM Transactions on Computer-Human Interaction (TOCHI) 18 (2011), no. 3, 1–20.

[140] Jeff Heaton, *An empirical analysis of feature engineering for predictive modeling*, SoutheastCon 2016, IEEE, 2016, pp. 1–6.

[141] J Hemabala and ESM Suresh, *The frame work design of mobile learning management system*, International Journal of Computer and Information Technology 1 (2012), no. 2, 179–184.

[142] Stefan Henß, Martin Monperrus, and Mira Mezini, *Semi-automatically extracting faqs to improve accessibility of software development knowledge*, 2012 34th International Conference on Software Engineering (ICSE), IEEE, 2012, pp. 793–803.

[143] Ralf Herbrich, Thore Graepel, and Colin Campbell, *Bayes point machines*, Journal of Machine Learning Research 1 (2001), no. Aug, 245–279.

[144] Michael Heron, Vicki L Hanson, and Ian Ricketts, *Open source and accessibility: advantages and limitations*, Journal of interaction Science 1 (2013), no. 1, 1–10.

[145] Xiao Hu, J Stephen Downie, and Andreas F Ehmann, *Lyric text mining in music mood classification*, American music 183 (2009), no. 5,049, 2–209.

[146] Ashatu Hussein, *The use of triangulation in social sciences research: Can qualitative and quantitative methods be combined*, Journal of comparative social work 1 (2009), no. 8, 1–12.

[147] Clayton Hutto and Eric Gilbert, *Vader: A parsimonious rule-based model for sentiment analysis of social media text*, Proceedings of the International AAAI Conference on Web and Social Media, vol. 8, 2014.

[148] Claudia Iacob and Rachel Harrison, *Retrieving and analyzing mobile apps feature requests from online reviews*, 2013 10th Working Conference on Mining Software Repositories (MSR) (San Francisco, CA, USA), IEEE, May 2013, pp. 41–44 (en).

[149] Venkata N Inukollu, Divya D Keshamoni, Taeghyun Kang, and Manikanta Inukollu, *Factors influencing quality of mobile apps: Role of mobile app development life cycle*, arXiv preprint arXiv:1410.4537 (2014).

[150] Jiyong Jang, David Brumley, and Shobha Venkataraman, *Bitshred: feature hashing malware for scalable triage and semantic analysis*, Proceedings of the 18th ACM conference on Computer and communications security, 2011, pp. 309–320.

[151] Nishant Jha and Anas Mahmoud, *Mining non-functional requirements from app store reviews*, Empirical Software Engineering 24 (2019), no. 6, 3659–3695.

[152] Cheryl DeConde Johnson, *Remote learning for children with auditory access needs: What we have learned during covid-19*, Seminars in hearing, vol. 41, Thieme Medical Publishers, Inc., 2020, pp. 302–308.

[153] Cijo Jose, Prasoon Goyal, Parv Aggrwal, and Manik Varma, *Local deep kernel learning*

for efficient non-linear svm prediction, International conference on machine learning, 2013, pp. 486–494.

[154] Konstantinos Karampidis, Athina Trigoni, Giorgos Papadourakis, Maria Christofaki, and Nuno Escudeiro, *Removing education barriers for deaf students at the era of covid-19*, 2021 30th Annual Conference of the European Association for Education in Electrical and Information Engineering (EAEEIE), IEEE, 2021, pp. 1–6.

[155] Nisal Karunaratne and Dilhara Karunaratne, *The implications of hearing loss on a medical student: A personal view and learning points for medical educators*, Medical Teacher (2020), 1–2.

[156] Riivo Kikas, Marlon Dumas, and Dietmar Pfahl, *Using dynamic and contextual features to predict issue lifetime in github projects*, 2016 IEEE/ACM 13th Working Conference on Mining Software Repositories (MSR), IEEE, 2016, pp. 291–302.

[157] Jennifer Renée Kilpatrick, Suzanne Ehrlich, and Michelle Bartlett, *Learning from covid-19: Universal design for learning implementation prior to and during a pandemic*, The Journal of Applied Instructional Design 10 (2021), no. 1.

[158] Royce Kimmons, *Open to all? nationwide evaluation of high-priority web accessibility considerations among higher education websites*, Journal of Computing in Higher Education 29 (2017), no. 3, 434–450.

[159] Shelley Kinash, Jeffrey Brand, and Trishita Mathew, *Challenging mobile learning discourse through research: Student perceptions of blackboard mobile learn and ipads*, Australasian journal of educational technology 28 (2012), no. 4.

[160] Margaret King-Sears, *Universal design for learning: Technology and pedagogy*, Learning Disability Quarterly 32 (2009), no. 4, 199–201.

[161] Ajla Kirlic and Zeynep Orhan, *Measuring human and vader performance on sentiment analysis*, Measuring human and Vader performance on sentiment analysis 1 (2017), 42–46.

[162] Barbara A Kitchenham and Shari Lawrence Pfleeger, *Principles of survey research part*

2: designing a survey, ACM SIGSOFT Software Engineering Notes 27 (2002), no. 1, 18–20.

[163] Eric Knauss, Daniela Damian, German Poo-Caamano, and Jane Cleland-Huang, *Detecting and classifying patterns of requirements clarifications*, 2012 20th IEEE International Requirements Engineering Conference (RE) (Chicago, IL, USA), IEEE, September 2012, pp. 251–260 (en).

[164] Ron Kohavi et al., *A study of cross-validation and bootstrap for accuracy estimation and model selection*, Ijcai, vol. 14, Montreal, Canada, 1995, pp. 1137–1145.

[165] Kowsari, Jafari Meimandi, Heidarysafa, Mendu, Barnes, and Brown, *Text Classification Algorithms: A Survey*, Information 10 (2019), no. 4, 150 (en).

[166] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown, *Text classification algorithms: A survey*, Information 10 (2019), no. 4, 150.

[167] Bartosz Krawczyk, *Learning from imbalanced data: open challenges and future directions*, Progress in Artificial Intelligence 5 (2016), no. 4, 221–232.

[168] Isai Amutan Krishnan, Geraldine De Mello, Shelen Aderina Kok, Saabdev Kumar Sabapathy, Saravanan Munian, Hee Sio Ching, Pushpa Kandasamy, Selvajothi Ramalingam, Shasthrika Baskaran, and Vasudevan Naidu Kanan, *Challenges faced by hearing impairment students during covid-19*, Malaysian Journal of Social Sciences and Humanities (MJSSH) 5 (2020), no. 8, 106–116.

[169] Karen L Kritzer and Chad E Smith, *Educating deaf and hard-of-hearing students during covid-19: What parents need to know*, The Hearing Journal 73 (2020), no. 8, 32.

[170] Rachmawan Adi Laksono, Kelly Rossa Sungkono, Riyanarto Sarno, and Cahyaningtyas Sekar Wahyuni, *Sentiment analysis of restaurant customer reviews on tripadvisor using naïve bayes*, 2019 12th International Conference on Information & Communication Technology and System (ICTS), IEEE, 2019, pp. 49–54.

[171] Marco Lazzari and Federica Baroni, *Remote teaching for deaf pupils during the covid-*

19 emergency, 14th International Conference on e-Learning 2020, IADIS Press, 2020, pp. 170–174.

[172] E learning Center, *Technical and vocational training corporation*, http://elearning.edu.sa/, April 2021.

[173] Ernesto Lee, Furqan Rustam, Wajdi Aljedaani, Abid Ishaq, Vaibhav Rupapara, and Imran Ashraf, *Research article predicting pulsars from imbalanced dataset with hybrid resampling approach*, (2021).

[174] Ernesto Lee, Furqan Rustam, Patrick Bernard Washington, Fatima El Barakaz, Wajdi Aljedaani, and Imran Ashraf, *Racism detection by analyzing differential opinions through sentiment analysis of tweets using stacked ensemble gcr-nn model*, IEEE Access 10 (2022), 9717–9728.

[175] Stefan Lessmann, Bart Baesens, Christophe Mues, and Swantje Pietsch, *Benchmarking classification models for software defect prediction: A proposed framework and novel findings*, IEEE Transactions on Software Engineering 34 (2008), no. 4, 485–496.

[176] Stanislav Levin and Amiram Yehudai, *Boosting Automatic Commit Classification Into Maintenance Activities By Utilizing Source Code Changes*, Proceedings of the 13th International Conference on Predictive Models and Data Analytics in Software Engineering - PROMISE (Toronto, Canada), ACM Press, 2017, pp. 97–106 (en).

[177] ———, *Boosting automatic commit classification into maintenance activities by utilizing source code changes*, Proceedings of the 13th International Conference on Predictive Models and Data Analytics in Software Engineering, 2017, pp. 97–106.

[178] ———, *Towards software analytics: Modeling maintenance activities*, arXiv preprint arXiv:1903.04909 (2019).

[179] ———, *Towards Software Analytics: Modeling Maintenance Activities*, arXiv:1903.04909 [cs] (2019) (en), arXiv: 1903.04909.

[180] Xiaoqing Li, *Students' acceptance of mobile learning: An empirical study based on blackboard mobile learn*, Mobile Devices in Education: Breakthroughs in Research and Practice, IGI Global, 2020, pp. 354–373.

[181] Xiaozhou Li, Zheying Zhang, and Kostas Stefanidis, *Mobile App Evolution Analysis based on User Reviews*, (2018), 14 (en).

[182] ———, *Mobile app evolution analysis based on user reviews*, New Trends in Intelligent Software Methodologies, Tools and Techniques, IOS Press, 2018, pp. 773–786.

[183] Anastasia Liasidou and Loizos Symeou, *Neoliberal versus social justice reforms in education policy and practice: Discourses, politics and disability rights in education*, Critical studies in education 59 (2018), no. 2, 149–166.

[184] Andy Liaw, Matthew Wiener, et al., *Classification and regression by randomforest*, R news 2 (2002), no. 3, 18–22.

[185] Eleanor Lisney, Jonathan P Bowen, Kirsten Hearn, and Maria Zedda, *Museums and technology: Being inclusive helps accessibility for all*, Curator: The Museum Journal 56 (2013), no. 3, 353–361.

[186] Yepang Liu, Chang Xu, and Shing-Chi Cheung, *Characterizing and detecting performance bugs for smartphone applications*, Proceedings of the 36th international conference on software engineering, 2014, pp. 1013–1024.

[187] Gary L Long, Carol Marchetti, and Richard Fasse, *The importance of interaction for academic success in online courses with hearing, deaf, and hard-of-hearing students*, International Review of Research in Open and Distributed Learning 12 (2011), no. 6, 1–19.

[188] Gary L Long, Karen Vignare, Raychel P Rappold, and Jim Mallory, *Access to communication for deaf, hard-of-hearing and esl students in blended learning courses*, International Review of Research in Open and Distributed Learning 8 (2007), no. 3, 1–13.

[189] Steven Loria, *textblob documentation*, Release 0.15 2 (2018), 269.

[190] Mengmeng Lu and Peng Liang, *Automatic classification of non-functional requirements from augmented app user reviews*, Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering, 2017, pp. 344–353.

[191] Matthew A Lynn, David C Templeton, Annemarie D Ross, Austin U Gehret, Morgan

Bida, Timothy J Sanger, and Todd Pagano, *Successes and challenges in teaching chemistry to deaf and hard-of-hearing students in the time of covid-19*, Journal of Chemical Education 97 (2020), no. 9, 3322–3326.

[192] Walid Maalej, Hans-Jörg Happel, and Asarnusch Rashid, *When users become collaborators: towards continuous and context-aware user input*, Proceedings of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications, 2009, pp. 981–990.

[193] Abdullah Madhesh, *Full exclusion during covid-19: Saudi deaf education is an example*, Heliyon (2021), e06536.

[194] Amiya Kumar Maji, Kangli Hao, Salmin Sultana, and Saurabh Bagchi, *Characterizing failures in mobile oses: A case study with android and symbian*, 2010 IEEE 21st International Symposium on Software Reliability Engineering, IEEE, 2010, pp. 249–258.

[195] Everton da Silva Maldonado, Emad Shihab, and Nikolaos Tsantalis, *Using Natural Language Processing to Automatically Detect Self-Admitted Technical Debt*, IEEE Transactions on Software Engineering 43 (2017), no. 11, 1044–1062 (en).

[196] Constantinos N Mantzikos and Christina S Lappa, *Difficulties and barriers in the education of deaf and hard of hearing individuals in the era of covid-19: The case of greece-a viewpoint article*, European Journal of Special Education Research 6 (2020), no. 3.

[197] Norbert Markus, Szabolcs Malik, Zoltan Juhasz, and András Arató, *Accessibility for the blind on an open-source mobile platform*, International Conference on Computers for Handicapped Persons, Springer, 2012, pp. 599–606.

[198] Juan Martınez-Miranda and Arantza Aldea, *Emotions in human and artificial intelligence*, Computers in Human Behavior 21 (2005), no. 2, 323–341.

[199] Universal Masking, *Unmasked: How the covid-19 pandemic exacerbates disparities for people with communication-based disabilities*, Journal of Hospital Medicine 16 (2021), no. 3, 185.

[200] Elizabeth Mathews, Patrick Cadwell, Shaun O'Boyle, and Senan Dunne, *Crisis in-*

terpreting and deaf community access in the covid-19 pandemic, Perspectives (2022), 1–19.

[201] Elizabeth S Mathews, *Signs of equity: Access to teacher education for deaf students in the republic of ireland*, Sign Language Studies 21 (2020), no. 1, 68–97.

[202] Diego Mayordomo-Martínez, Juan M Carrillo-de Gea, Ginés García-Mateos, José A García-Berná, José Luis Fernández-Alemán, Saúl Rosero-López, Salvador Parada-Sarabia, and Manuel García-Hernández, *Sustainable accessibility: a mobile app for helping people with disabilities to search accessible shops*, International journal of environmental research and public health 16 (2019), no. 4, 620.

[203] Stuart McIlroy, Nasir Ali, Hammad Khalid, and Ahmed E. Hassan, *Analyzing and automatically labelling the types of user issues that are raised in mobile app reviews*, Empirical Software Engineering 21 (2016), no. 3, 1067–1106 (en).

[204] Stuart McIlroy, Nasir Ali, Hammad Khalid, and Ahmed E Hassan, *Analyzing and automatically labelling the types of user issues that are raised in mobile app reviews*, Empirical Software Engineering 21 (2016), no. 3, 1067–1106.

[205] Michael McKee, Christa Moran, and Philip Zazove, *Overcoming additional barriers to care for deaf and hard of hearing patients during covid-19*, JAMA Otolaryngology–Head & Neck Surgery 146 (2020), no. 9, 781–782.

[206] Caitlin McKeown and Julia McKeown, *Accessibility in online courses: understanding the deaf learner*, TechTrends 63 (2019), no. 5, 506–513.

[207] André Luís Ferreira Meireles and Louisiana Carolina Ferreira de Meireles, *Impact of social isolation due to the covid-19 pandemic in patients with pediatric disorders: Rehabilitation perspectives from a developing country*, Physical Therapy 100 (2020), no. 11, 1910–1912.

[208] Zoë Meleo-Erwin, Betty Kollia, Joe Fera, Alyssa Jahren, and Corey Basch, *Online support information for students with disabilities in colleges and universities during the covid-19 pandemic*, Disability and Health Journal 14 (2021), no. 1, 101013.

[209] Montassar Ben Messaoud, Ilyes Jenhani, Nermine Ben Jemaa, and Mohamed Wiem

Mkaouer, *A multi-label active learning approach for mobile app user review classification*, International Conference on Knowledge Science, Engineering and Management, Springer, 2019, pp. 805–816.

[210] Tom Mitchell, *Introduction to machine learning*, Machine Learning 7 (1997), 2–5.

[211] Tom M Mitchell et al., *Machine learning*, (1997).

[212] Noor-ud-din Mohammed, *Deaf students' linguistic access in online education: The case of trinidad*, Deafness & Education International 23 (2021), no. 3, 217–233.

[213] Monorail, *Monorail issue tracker)*, https://bugs.chromium.org/p/chromium/issues/list, June 2020.

[214] Daniel Mont, *Combatting the costs of exclusion for children with disabilities and their families.*, UNICEF (2021).

[215] Kevin Moran, Richard Bonett, Carlos Bernal-Cárdenas, Brendan Otten, Daniel Park, and Denys Poshyvanyk, *On-device bug reporting for android applications*, 2017 IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems (MOBILESoft), IEEE, 2017, pp. 215–216.

[216] Zethembe Mseleku, *A literature review of e-learning and e-teaching in the era of covid-19 pandemic*, SAGE 57 (2020), no. 52, 6.

[217] Susan M Mudambi and David Schuff, *Research note: What makes a helpful online review? a study of customer reviews on amazon. com*, MIS quarterly (2010), 185–200.

[218] Debjyoti Mukherjee and Guenther Ruhe, *Analysis of compatibility in open source android mobile apps*, 2020 IEEE Seventh International Workshop on Artificial Intelligence for Requirements Engineering (AIRE), IEEE, 2020, pp. 70–78.

[219] Puja Munjal, Meenakshi Narula, Sandeep Kumar, and Hema Banati, *Twitter sentiments based suggestive framework to predict trends*, Journal of Statistics and Management Systems 21 (2018), no. 4, 685–693.

[220] Stanley Murairwa, *Voluntary sampling design*, International Journal of Advanced Research in Management and Social Sciences 4 (2015), no. 2, 185–200.

[221] Emerson Murphy-Hill, Chris Parnin, and Andrew P. Black, *How We Refactor, and*

How We Know It, IEEE Transactions on Software Engineering 38 (2012), no. 1, 5–18 (en).

[222] Rila Muspita, Achmad Hufad, Asep Bayu Dani Nandiyanto, Reno Fernandes, Amin Akbar, Tryastuti Irawati Belliny Manullang, Rafikah Trinalia, et al., *Developing a media to teach chemical technology to students with hearing impairments*, Journal of Engineering Education Transformations 34 (2020), 43–48.

[223] Millicent Malinda Musyoka and Zanthia Yvette Smith, *Mainstreamed deaf/hh students' online learning in k-12: Challenges, opportunities, and solutions*, Curriculum Development and Online Instruction for the 21st Century, IGI Global, 2021, pp. 69–89.

[224] Reem N. Alshenaifi and Jinjuan Heidi Feng, *Investigating the use of social media in supporting children with cognitive disabilities and their caregivers from saudi arabia*, The 22nd International ACM SIGACCESS Conference on Computers and Accessibility, 2020, pp. 1–4.

[225] Alexey Natekin and Alois Knoll, *Gradient boosting machines, a tutorial*, Frontiers in neurorobotics 7 (2013), 21.

[226] United Nations, *A disability-inclusive response to covid-19*, https://www.un.org/en/coronavirus/disability-inclusion, March 2020.

[227] Mariaclaudia Nicolai, Luca Pascarella, Fabio Palomba, and Alberto Bacchelli, *Healthcare android apps: a tale of the customers' perspective*, Proceedings of the 3rd ACM SIGSOFT International Workshop on App Market Analytics, 2019, pp. 33–39.

[228] Lorelli S Nowell, Jill M Norris, Deborah E White, and Nancy J Moules, *Thematic analysis: Striving to meet the trustworthiness criteria*, International journal of qualitative methods 16 (2017), no. 1, 1609406917733847.

[229] Obianuju Okafor, Wajdi Aljedaani, and Stephanie Ludi, *Comparative analysis of accessibility testing tools and their limitations in rias*, International Conference on Human-Computer Interaction, Springer, 2022, pp. 479–500.

[230] Alberto Oliveira, Paulo Sérgio dos Santos, Wilson Estécio Júnior, Wajdi Aljedaani, Danilo Eler, and Marcelo Eler, *Analyzing accessibility reviews associated with visual*

disabilities or eye conditionsility user reviews, Proceedings of the 2023 CHI conference on human factors in computing systems, 2023, pp. 1–15.

[231] Boutkhoum Omar, Furqan Rustam, Arif Mehmood, Gyu Sang Choi, et al., *Minimizing the overlapping degree to improve class-imbalanced learning under sparse feature selection: application to fraud detection*, IEEE Access 9 (2021), 28101–28110.

[232] Rachel O'Neill and Brian Shannan, *The views and experiences of deaf young people and their parents using assistive devices at home before and during the covid-19 pandemic*, Moray House School of Education and Sport, University of Edinburgh (2022).

[233] Luis Ortiz-Jiménez, Victoria Figueredo-Canosa, Macarena Castellary López, and María Carmen López Berlanga, *Teachers' perceptions of the use of icts in the educational response to students with disabilities*, Sustainability 12 (2020), no. 22, 9446.

[234] Moein Owhadi-Kareshk, Sarah Nadi, and Julia Rubin, *Predicting merge conflicts in collaborative software development*, 2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), IEEE, 2019, pp. 1–11.

[235] Osman Özokcu and Taskin Yildirim, *Determining the fears of student with special needs in inclusive environments.*, International Education Studies 11 (2018), no. 6, 174–182.

[236] Rachel O'Neill and Jill Duncan, *From policy to practice: Working globally and standing united to support deaf children's education*, 2021.

[237] Louise Paatsch and Dianne Toe, *The impact of pragmatic delays for deaf and hard of hearing students in mainstream classrooms*, Pediatrics 146 (2020), no. Supplement 3, S292–S297.

[238] Fabio Palomba, Mario Linares-Vasquez, Gabriele Bavota, Rocco Oliveto, Massimiliano Di Penta, Denys Poshyvanyk, and Andrea De Lucia, *User reviews matter! Tracking crowdsourced reviews to support evolution of successful apps*, 2015 IEEE International Conference on Software Maintenance and Evolution (ICSME) (Bremen, Germany), IEEE, September 2015, pp. 291–300 (en).

[239] Fabio Palomba, Mario Linares-Vásquez, Gabriele Bavota, Rocco Oliveto, Massimiliano Di Penta, Denys Poshyvanyk, and Andrea De Lucia, *User reviews matter! tracking*

crowdsourced reviews to support evolution of successful apps, 2015 IEEE international conference on software maintenance and evolution (ICSME), IEEE, 2015, pp. 291–300.

[240] Sebastiano Panichella, Andrea Di Sorbo, Emitza Guzman, Corrado A Visaggio, Gerardo Canfora, and Harald C Gall, *How can i improve my app? classifying user reviews for software maintenance and evolution*, 2015 IEEE international conference on software maintenance and evolution (ICSME), IEEE, 2015, pp. 281–290.

[241] Sebastiano Panichella, Andrea Di Sorbo, Emitza Guzman, Corrado A. Visaggio, Gerardo Canfora, and Harald C. Gall, *How can i improve my app? Classifying user reviews for software maintenance and evolution*, 2015 IEEE International Conference on Software Maintenance and Evolution (ICSME) (Bremen, Germany), IEEE, September 2015, pp. 281–290 (en).

[242] Kyudong Park, Taedong Goh, and Hyo-Jeong So, *Toward accessible mobile application design: Developing mobile application accessibility guidelines for people with visual impairment*, Proceedings of HCI Korea (Seoul, KOR), HCIK '15, Hanbit Media, Inc., 2014, p. 31–38.

[243] Luca Pascarella, Davide Spadini, Fabio Palomba, Magiel Bruntink, and Alberto Bacchelli, *Information needs in contemporary code review*, Proceedings of the ACM on Human-Computer Interaction 2 (2018), no. CSCW, 1–27.

[244] Rohan Patel, Pedro Breton, Catherine M Baker, Yasmine N El-Glaly, and Kristen Shinohara, *Why software is not accessible: Technology professionals' perspectives and challenges*, Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems, 2020, pp. 1–9.

[245] Lucas Pelloni, Giovanni Grano, Adelina Ciurumelea, Sebastiano Panichella, Fabio Palomba, and Harald C. Gall, *BECLoMA: Augmenting stack traces with user review information*, 2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER) (Campobasso), IEEE, March 2018, pp. 522–526 (en).

[246] Lucas Pelloni, Giovanni Grano, Adelina Ciurumelea, Sebastiano Panichella, Fabio Palomba, and Harald C Gall, *Becloma: Augmenting stack traces with user review in-*

formation, 2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER), IEEE, 2018, pp. 522–526.

[247] Christopher Power, André Freire, Helen Petrie, and David Swallow, *Guidelines are only half of the story: accessibility problems encountered by blind users on the web*, Proceedings of the SIGCHI conference on human factors in computing systems, 2012, pp. 433–442.

[248] Rizqi Fajar Pradipta, Mohammad Efendi, Abdul Huda, Dimas Arif Dewantoro, and Mohd Hanafi Mohd Yasin, *Comparative study: Use of ict media in learning for deaf students during the covid-19 pandemic in malaysia and indonesia*, 7th International Conference on Education and Technology (ICET 2021), Atlantis Press, 2021, pp. 182–188.

[249] Anita Prinzie and Dirk Van den Poel, *Random forests for multiclass classification: Random multinomial logit*, Expert systems with Applications 34 (2008), no. 3, 1721–1732.

[250] Cynthia Putnam, Kathryn Wozniak, Mary Jo Zefeldt, Jinghui Cheng, Morgan Caputo, and Carl Duffield, *How do professionals who create computing technologies consider accessibility?*, Proceedings of the 14th international ACM SIGACCESS conference on Computers and accessibility, 2012, pp. 87–94.

[251] Rajeev DS Raizada and Yune-Sang Lee, *Smoothness without smoothing: why gaussian naive bayes is not naive for multi-subject searchlight studies*, PloS one 8 (2013), no. 7, e69566.

[252] Mohammad H Rajab, Abdalla M Gazal, and Khaled Alkattan, *Challenges to online medical education during the covid-19 pandemic*, Cureus 12 (2020), no. 7.

[253] Jacek Ratzinger, Thomas Sigmund, and Harald C Gall, *On the relation of refactorings and software defect prediction*, Proceedings of the 2008 international working conference on Mining software repositories, 2008, pp. 35–38.

[254] Aijaz Ahmad Reshi, Furqan Rustam, Wajdi Aljedaani, Shabana Shafi, Abdulaziz Alhossan, Ziyad Alrabiah, Ajaz Ahmad, Hessa Alsuwailem, Thamer A Almangour,

Musaad A Alshammari, et al., *Covid-19 vaccination-related sentiments analysis: a case study using worldwide twitter dataset*, Healthcare, vol. 10, MDPI, 2022, p. 411.

[255] André Rodrigues, Hugo Nicolau, Kyle Montague, João Guerreiro, and Tiago Guerreiro, *Open challenges of blind people using smartphones*, International Journal of Human–Computer Interaction 36 (2020), no. 17, 1605–1622.

[256] Anne Spencer Ross, Xiaoyi Zhang, James Fogarty, and Jacob O. Wobbrock, *Epidemiology as a framework for large-scale mobile application accessibility assessment*, Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility (New York, NY, USA), ASSETS '17, Association for Computing Machinery, 2017, p. 2–11.

[257] ———, *Examining image-based button labeling for accessibility in android apps through large-scale analysis*, Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility (New York, NY, USA), ASSETS '18, Association for Computing Machinery, 2018, p. 119–130.

[258] Kathryn Roulston, *Data analysis and 'theorizing as ideology'*, Qualitative research 1 (2001), no. 3, 279–302.

[259] Per Runeson and Martin Höst, *Guidelines for conducting and reporting case study research in software engineering*, Empirical software engineering 14 (2009), no. 2, 131–164.

[260] Vaibhav Rupapara, Furqan Rustam, Wajdi Aljedaani, Hina Fatima Shahzad, Ernesto Lee, and Imran Ashraf, *Blood cancer prediction using leukemia microarray gene data and hybrid logistic vector trees model*, Scientific Reports 12 (2022), no. 1, 1–15.

[261] Shanna Russ and Foad Hamidi, *Online learning accessibility during the covid-19 pandemic*, Proceedings of the 18th International Web for All Conference, 2021, pp. 1–7.

[262] Furqan Rustam, Madiha Khalid, Waqar Aslam, Vaibhav Rupapara, Arif Mehmood, and Gyu Sang Choi, *A performance comparison of supervised machine learning models for covid-19 tweets sentiment analysis*, Plos one 16 (2021), no. 2, e0245909.

[263] Furqan Rustam, Aijaz Ahmad Reshi, Wajdi Aljedaani, Abdulaziz Alhossan, Abid

Ishaq, Shabana Shafi, Ernesto Lee, Ziyad Alrabiah, Hessa Alsuwailem, Ajaz Ahmad, et al., *Vector mosquito image classification using novel rifs feature selection and machine learning models for disease epidemiology*, Saudi Journal of Biological Sciences 29 (2022), no. 1, 583–594.

[264] Nasir Safdari, Hussein Alrubaye, Wajdi Aljedaani, Bladimir Baez Baez, Andrew DiStasi, and Mohamed Wiem Mkaouer, *Learning to rank faulty source files for dependent bug reports*, Big Data: Learning, Analytics, and Applications, vol. 10989, International Society for Optics and Photonics, 2019, p. 109890B.

[265] Melania Safirista, Sofiarti Murtadlo, and Endang Pudjisartinah, *A study accessibility of deaf students during the covid-19 pandemic*, Eighth Southeast Asia Design Research (SEA-DR) & the Second Science, Technology, Education, Arts, Culture, and Humanity (STEACH) International Conference (SEADR-STEACH 2021), Atlantis Press, 2022, pp. 79–82.

[266] Mehmet Şahin, *Blended learning environment in vocational education*, The 5th International Conference on Virtual Learning (ICVL) (2010), 244–254.

[267] Arthur L Samuel, *Some studies in machine learning using the game of checkers*, IBM Journal of research and development 3 (1959), no. 3, 210–229.

[268] Erin C Schafer, Benjamin Kirby, and Sharon Miller, *Remote microphone technology for children with hearing loss or auditory processing issues*, Seminars in Hearing, vol. 41, Thieme Medical Publishers, Inc., 2020, pp. 277–290.

[269] Scikit-learn.org, *Parameter estimation using grid search with scikit-learn. available online:*, https://scikit-learn.org/stable/modules/grid_search.html,, 2006, Accessed: 2020-04-01.

[270] Matthew Seita, Khaled Albusays, Sushant Kafle, Michael Stinson, and Matt Huenerfauth, *Behavioral changes in speakers who are automatically captioned in meetings with deaf or hard-of-hearing peers*, Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility, 2018, pp. 68–80.

[271] Norbert Seyff, Florian Graf, and Neil Maiden, *Using Mobile RE Tools to Give End-*

Users Their Own Voice, 2010 18th IEEE International Requirements Engineering Conference (Sydney, Australia), IEEE, September 2010, pp. 37–46 (en).

[272] ———, *Using mobile re tools to give end-users their own voice*, 2010 18th IEEE International Requirements Engineering Conference, IEEE, 2010, pp. 37–46.

[273] Colin Shanley, *Cracking accessibility on mobile devices: The definitive field guide to accessibility and digital inclusion for business managers and project teams*, 2016.

[274] Qinfeng Shi, James Petterson, Gideon Dror, John Langford, Alex Smola, and SVN Vishwanathan, *Hash kernels for structured data*, The Journal of Machine Learning Research 10 (2009), 2615–2637.

[275] Won Sug Shin and Minseok Kang, *The use of a mobile learning management system at an online university and its effect on learning satisfaction and achievement*, International Review of Research in Open and Distributed Learning 16 (2015), no. 3, 110–130.

[276] Jamie Shotton, Toby Sharp, Pushmeet Kohli, Sebastian Nowozin, John Winn, and Antonio Criminisi, *Decision jungles: Compact and rich models for classification*, Advances in Neural Information Processing Systems, 2013, pp. 234–242.

[277] R Shreyas, DM Akshata, BS Mahanand, B Shagun, and CM Abhishek, *Predicting popularity of online articles using random forest regression*, 2016 Second International Conference on Cognitive Computing and Information Processing (CCIP), IEEE, 2016, pp. 1–5.

[278] João Silva, Ramiro Gonçalves, José Martins, Frederico Branco, and António Pereira, *Accessibility in software engineering: pursuing the mainstream from a classroom*, International Conference on Learning and Collaboration Technologies, Springer, 2018, pp. 505–517.

[279] Samuel B Slike, Pamela D Berman, Travis Kline, Kathryn Rebilas, and Erin Bosch, *Providing online course opportunities for learners who are deaf, hard of hearing, or hearing*, American Annals of the deaf 153 (2008), no. 3, 304–308.

[280] Chad Smith and Sarah Colton, *Creating a youtube channel to equip parents and teach-*

ers of students who are deaf, Journal of Technology and Teacher Education 28 (2020), no. 2, 453–461.

[281] Clinton Smith, *Challenges and opportunities for teaching students with disabilities during the covid-19 pandemic*, International Journal of Multidisciplinary Perspectives in Higher Education 5 (2020), no. 1, 167–173.

[282] Edward Smith, Robert Loftin, Emerson Murphy-Hill, Christian Bird, and Thomas Zimmermann, *Improving developer participation rates in surveys*, 2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), IEEE, 2013, pp. 89–92.

[283] Erika E Smith, *3 things to consider when designing remote teaching.*, Mount Royal University (2020).

[284] Kylie Sommer, *The effect of covid-19 on deaf and hard of hearing college students*, 4 Kevin Ung, Director of McNair Scholar's Program (2020), 312.

[285] Kaitlin Stack Whitney and Kristoffer Whitney, *Inaccessible media during the covid-19 crisis intersects with the language deprivation crisis for young deaf children in the us*, Journal of Children and Media (2020), 1–4.

[286] Statista, *Number of mobile (cellular) subscriptions worldwide from 1993 to 2021*, https://www.statista.com/statistics/262950/global-mobile-subscriptions-since-1993/, December 2021.

[287] Konstantinos Stroggylos and Diomidis Spinellis, *Refactoring–Does It Improve Software Quality?*, Fifth International Workshop on Software Quality (WoSQ'07: ICSE Workshops 2007) (Minneapolis, MN, USA), IEEE, May 2007, pp. 10–10 (en).

[288] K Supriya, Chris Mead, Ariel D Anbar, Joshua L Caulkins, James P Collins, Katelyn M Cooper, Paul C LePore, Tiffany Lewis, Amy Pate, Rachel A Scott, et al., *Covid-19 and the abrupt shift to remote learning: Impact on grades and perceived learning for undergraduate biology students*, bioRxiv (2021).

[289] Halley Sutton, *Guide offers best practices for meeting the needs of deaf students during covid-19 pandemic*, Disability Compliance for Higher Education 26 (2020), no. 4, 9–9.

[290] Ruth Swanwick, Alexander M Oppong, Yaw N Offei, Daniel Fobi, Obed Appau, Joyce Fobi, and F Frempomaa Mantey, *The impact of the covid-19 pandemic on deaf adults, children and their families in ghana*, Journal of the British Academy 8 (2020), 141–165.

[291] Amy Szarkowski, Alys Young, Danielle Matthews, and Jareen Meinzen-Derr, *Pragmatics development in deaf and hard of hearing children: a call to action*, Pediatrics 146 (2020), no. Supplement 3, S310–S315.

[292] Chade-Meng Tan, Yuan-Fang Wang, and Chan-Do Lee, *The use of bigrams to enhance text categorization*, Information Processing & Management 38 (2002), no. 4, 529–546 (en).

[293] Jenifer Tidwell, *Designing interfaces: Patterns for effective interaction design*, " O'Reilly Media, Inc.", 2010.

[294] Garreth W Tigwell, David R Flatla, and Neil D Archibald, *Ace: a colour palette design tool for balancing aesthetics and accessibility*, ACM Transactions on Accessible Computing (TACCESS) 9 (2017), no. 2, 1–32.

[295] Garreth W Tigwell, Roshan L Peiris, Stacey Watson, Gerald M Garavuso, and Heather Miller, *Student and teacher perspectives of learning asl in an online setting*, The 22nd International ACM SIGACCESS Conference on Computers and Accessibility, 2020, pp. 1–6.

[296] Ahmed Tlili, Natalia Amelina, Daniel Burgos, Achraf Othman, Ronghuai Huang, Mohamed Jemni, Mirjana Lazor, Xiangling Zhang, and Ting-Wen Chang, *Remote special education during crisis: Covid-19 as a case study*, Radical Solutions for Education in a Crisis Context, Springer, 2020, pp. 69–83.

[297] Elena Tomasuolo, Tiziana Gulli, Virginia Volterra, and Sabina Fontana, *The italian deaf community at the time of coronavirus*, Frontiers in Sociology 5 (2021), 125.

[298] TVTC, *Rayat*, https://www.tvtc.gov.sa/rayat.html, 2021, (Accessed: 03/14/2021).

[299] Twitter, *The new skype is beyond frustrating*, 2018.

[300] Qasim Umer, Hui Liu, and Yasir Sultan, *Emotion based automated priority prediction for bug reports*, IEEE Access 6 (2018), 35743–35752.

[301] UNESCO, *Including learners with disabilities in covid-19 education responses*, https://en.unesco.org/news/including-learners-disabilities-covid-19-education-responses, February 2020.

[302] UNESCO-UNEVOC, *Technical and vocational training corporation*, https://unevoc.unesco.org/home/Explore+the+UNEVOC+Network/centre=300, December 2020.

[303] Solomon Ogbomon Uwagbole, William J Buchanan, and Lu Fan, *Applied machine learning predictive analytics to sql injection attack detection and prevention*, 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), IEEE, 2017, pp. 1087–1090.

[304] Mojtaba Vaismoradi, Hannele Turunen, and Terese Bondas, *Content analysis and thematic analysis: Implications for conducting a qualitative descriptive study*, Nursing & health sciences 15 (2013), no. 3, 398–405.

[305] Nisha Valvi, Sanjeev Sonawane, and Priti Jadhav, *Preparing inclusive class for the children with special needs during covid-19 crisis*, Educational Quest 11 (2020), no. 3, 183–187.

[306] Rajesh Vasa, Leonard Hoon, Kon Mouzakis, and Akihiro Noguchi, *A preliminary analysis of mobile app user reviews*, Proceedings of the 24th Australian computer-human interaction conference, 2012, pp. 241–244.

[307] Christopher Vendome, Diana Solano, Santiago Liñán, and Mario Linares-Vásquez, *Can everyone use my app? an empirical study on accessibility in android apps*, 2019 IEEE International Conference on Software Maintenance and Evolution (ICSME), IEEE, 2019, pp. 41–52.

[308] S Vijayarani, R Janani, et al., *Text mining: open source tokenization tools-an analysis*, Advanced Computational Intelligence: An International Journal (ACII) 3 (2016), no. 1, 37–47.

[309] Phong Minh Vu, Tam The Nguyen, Hung Viet Pham, and Tung Thanh Nguyen, *Mining User Opinions in Mobile App Reviews: A Keyword-based Approach*, arXiv:1505.04657 [cs] (2015) (en), arXiv: 1505.04657.

[310] ———, *Mining user opinions in mobile app reviews: A keyword-based approach*, arXiv preprint arXiv:1505.04657 (2015).

[311] W3C, *Web content accessibility guidelines (wcag) 2.1*, https://www.w3.org/TR/WCAG21/, June 2021.

[312] Eman Walabe, *E-learning delivery in saudi arabian universities*, Ph.D. thesis, Université d'Ottawa/University of Ottawa, 2020.

[313] Jiahui Wang and Chen Li, *Research on user satisfaction of video education application based on reviews*, 2020 IEEE 10th International Conference on Electronics Information and Emergency Communication (ICEIEC), IEEE, 2020, pp. 340–343.

[314] Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg, *Feature hashing for large scale multitask learning*, Proceedings of the 26th annual international conference on machine learning, 2009, pp. 1113–1120.

[315] Brian Wentz, Dung Pham, Erin Feaser, Dylan Smith, James Smith, and Allison Wilson, *Documenting the accessibility of 100 us bank and finance websites*, Universal Access in the Information Society 18 (2019), no. 4, 871–880.

[316] Krzysztof Wnuk and Thrinay Garrepalli, *Knowledge management in software testing: A systematic snowball literature review*, e-Informatica 12 (2018), no. 1, 51–78.

[317] Claes Wohlin, *Guidelines for snowballing in systematic literature studies and a replication in software engineering*, Proceedings of the 18th international conference on evaluation and assessment in software engineering, 2014, pp. 1–10.

[318] Xindong Wu, Vipin Kumar, J Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J McLachlan, Angus Ng, Bing Liu, S Yu Philip, et al., *Top 10 algorithms in data mining*, Knowledge and information systems 14 (2008), no. 1, 1–37.

[319] Shunguo Yan and P. G. Ramachandran, *The Current Status of Accessibility in Mobile Apps*, ACM Transactions on Accessible Computing 12 (2019), no. 1, 1–31 (en).

[320] Shunguo Yan and PG Ramachandran, *The current status of accessibility in mobile apps*, ACM Transactions on Accessible Computing (TACCESS) 12 (2019), no. 1, 1–31.

[321] Ying Yang, Yanan Xiao, Yulu Liu, Qiong Li, Changshuo Shan, Shulin Chang, and

Philip H-S Jen, *Mental health and psychological impact on students with or without hearing loss during the recurrence of the covid-19 pandemic in china*, International Journal of Environmental Research and Public Health 18 (2021), no. 4, 1421.

[322] Xin Ye, Yongjie Zheng, Wajdi Aljedaani, and Mohamed Wiem Mkaouer, *Recommending pull request reviewers based on code changes*, Soft Computing 25 (2021), no. 7, 5619–5632.

[323] William S Yerazunis, *Sparse binary polynomial hashing and the crm114 discriminator*, 2003 Cambridge Spam Conference Proceedings, vol. 1, 2003.

[324] Saber Yezli and Anas Khan, *Covid-19 social distancing in the kingdom of saudi arabia: Bold measures in the face of political, economic, social and religious challenges*, Travel Medicine and Infectious Disease (2020), 101692.

[325] Joong-O Yoon and Minjeong Kim, *The effects of captions on deaf students' content comprehension, cognitive load, and motivation in online learning*, American annals of the deaf 156 (2011), no. 3, 283–289.

[326] Bei Yu, *An evaluation of text classification methods for literary study*, Literary and Linguistic Computing 23 (2008), no. 3, 327–343.

[327] Nor Shahida Mohamad Yusop, John Grundy, and Rajesh Vasa, *Reporting usability defects: do reporters report what software developers need?*, Proceedings of the 20th international conference on evaluation and assessment in software engineering, 2016, pp. 1–10.

[328] Zaidah Zainal, *Case study as a research method*, Jurnal kemanusiaan 5 (2007), no. 1.

[329] Han Zhang, Paula Nurius, Yasaman Sefidgar, Margaret Morris, Sreenithi Balasubramanian, Jennifer Brown, Anind K Dey, Kevin Kuehn, Eve Riskin, Xuhai Xu, et al., *How does covid-19 impact students with disabilities/health concerns?*, arXiv preprint arXiv:2005.05438 (2020).