Future Generation Computer Systems 158 (2024) 167–182

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Prioritizing user requirements for digital products using explainable artificial intelligence: A data-driven analysis on video conferencing apps

Shizhen Bai [a], Songlin Shi [a], Chunjia Han [b,*], Mu Yang [b], Brij B. Gupta [c,d,e,f,**], Varsha Arya [g,h]

[a] School of Management, Harbin University of Commerce, Harbin, China
[b] Department of Management, Birkbeck, University of London, UK
[c] Department of Computer Science and Information Engineering, Asia University, Taichung 413, Taiwan, China
[d] Kyung Hee University, 26 Kyungheedae-ro, Dongdaemun-gu, Seoul 02447, Korea
[e] Symbiosis Centre for Information Technology (SCIT), Symbiosis International University, Pune, India
[f] Center for Interdisciplinary Research, University of Petroleum and Energy Studies (UPES), Dehradun, India
[g] Department of Business Administration, Asia University, Taiwan, China
[h] Department of Electrical and Computer Engineering, Lebanese American University, Beirut, 1102, Lebanon

ARTICLE INFO

Keywords:
User requirements
Requirements prioritization
Video conferencing apps
App reviews
Text analysis
Explainable artificial intelligence

ABSTRACT

The advent of Industry 5.0 has brought a wealth of digital information to mobile app stores. With the help of emerging technologies such as machine learning and explainable artificial intelligence (XAI), these large amounts of user-generated data can be efficiently captured and analyzed. In this study, we propose an app store analysis framework and demonstrate the utility of the framework by mining and prioritizing user requirements in three popular video conferencing apps. We used the Sentistrength sentiment analysis tool, structural topic modeling, the Gephi web analysis tool, machine learning, and XAI techniques to conduct an in-depth analysis of user requirements in Microsoft Teams, ZOOM Cloud Meetings, and Google Meet. The findings indicated that Steal data, Audio and video quality, Customer service, Hacker issues, Meeting and account passwords, Mute and unmute, Features, and Office platform were the web conferencing system's key areas for improvement. The study demonstrated the usability of app store analysis frameworks and the great potential of XAI to provide insights about requirements prioritization by interpreting machine learning models. Additionally, it offered valuable suggestions for app developers on using the massive data in app stores to improve their apps.

1. Introduction

Mobile devices' increasing portability and intelligence have propelled the app market's growth, positioning it as one of the most alluring and swiftly developing sectors [1]. Software distribution channels like Google Play and the Apple App Store host millions of apps, bringing in billions of dollars for Apple, Google, and so on [2]. Recent statistics show that in the third quarter of 2022, there are approximately 3.55 million apps to choose from on the Android platform and 1.6 million on the ios platform [3]. For app developers to derive value from the app store, they must be able to promptly and accurately comprehend user-posted content and other pertinent information and subsequently transform this data into valuable insights [4]. The app store allows developers to keep an eye on the market environment (such as customers and rivals) in real-time and spot opportunities and possible application challenges [5]. Mobile app developers can also integrate insights from app stores to switch strategies based on facts or gain a deeper understanding of their products and services [6].

App developers must be aware of the current competitive environment and be thoroughly aware of user groups' preferences and requirements to succeed in the app industry [7]. In addition to lowering the risk of failure, close user interaction is essential for boosting market competition and promoting innovation. The app store provides a sustainable channel for developers to obtain app feedback, which contains a wealth of valuable app-related data, such as user reviews, ratings, and app rankings. According to recent research, these data contain both technical and non-technical information that app developers may employ [8]. App developers may directly benefit from app store research findings, which are also frequently utilized in fields like requirements engineering, release planning, software design, security, and testing

* Corresponding author. Department of Management, Birkbeck, University of London, Malet Street, Bloomsbury, London WC1E 7HX, UK.
** Corresponding author. Department of Computer Science and Information Engineering, Asia University, Taichung 413, Taiwan, China
E-mail addresses: chunjia.han@bbk.ac.uk (C. Han), bbgupta@asia.edu.tw (B.B. Gupta).

https://doi.org/10.1016/j.future.2024.04.037
Received 2 December 2023; Received in revised form 11 April 2024; Accepted 16 April 2024
Available online 18 April 2024

methodologies [5,9]. Among them, requirements engineering has garnered extensive attention, which is not surprising because developers must meet the changing requirements of users to survive in the competition [10]. Apps are getting more and more research in requirements engineering [11]. However, the potential value of big data in app stores for requirement prioritization research and practice has not been well explored [8]. Previous research has usually focused on traditional data sources (such as questionnaires) and analytical methods (e.g., Analytic Hierarchy Process (AHP) and MoSCoW). As the requirement for information increases, app developers have struggled to process such large amounts of data manually using traditional methods and may run into computational complexity issues [12]. Additionally, rapidly growing app store data requires app developers to employ automated analytics wherever possible to gain insights about their apps and competitors [13].

The primary goal of this study is to convert app store data (structured and unstructured data) into supporting data to enhance app functionality, design, and service as well as to speed up app evolution. The difficulty of processing app store data fully manually makes this a barrier for app developers [14,15]. This study concerns web conferencing systems (WCS), one of the fastest growing, most significant, and fastest expanding markets in recent years, directly affecting people's everyday lives [16]. Some studies have explored the use of WCS, and the problems discovered in the outcomes might assist app developers in further optimizing and strengthening their products [17,18]. However, according to the literature survey, most of the literature is too general to be applied by developers of WCS, as research focuses on WCS usage trends [19] and its impact [20,21], while research on user requirements is still in its infancy. In contrast, this study is more specific, and the results obtained are more practical. Our research question is as follows:

How should WCS developers mine and prioritize users' pressing requirements from app store data to improve apps?

To address this issue, this study devised a framework for analyzing app store data to explore the current use of various approaches and techniques and seek to gain potential insights into app stores in the following areas: (i) highlighting users' most urgent requirements, (ii) identifying linkages between important requirements and (iii) prioritizing these requirements based on interpretable results from the model. The analysis selected the top three video conferencing software in the Apple App Store. Our contributions to this work are four-fold. Firstly, this study contributes to the software engineering literature by designing an analytical framework for WCS developers, which may help them to use data from the app store in the development, operation, improvement, and update of their apps. Secondly, the proposed requirements prioritization technique does not require significant human intervention and domain knowledge. In particular, the methodology employs the latest Explainable Artificial Intelligence (XAI) techniques, and the results obtained may be easier to understand and interpret compared to existing studies. Thirdly, the proposed approach can easily be extended to apps in various domains if the appropriate data is collected. Fourth, the findings provide insights for developers and operators on the use of app stores to support WCS in developing marketing and promotional strategies and improving product and service quality, which can improve mobile app companies' understanding of the business value of app store data and big data analytics. To our knowledge, this is the first study to use XAI to guide the prioritization of WCS user requirements, and the study may contribute to the advancement of research into big data analytics techniques.

1.1. App store review analysis

An application (app) is a type of software designed to run on a specific development platform [22]. The development platform can be a web browser such as Microsoft Edge [23], an app marketplace (e.g., Apple App Store) [24], or a social media platform (e.g., Facebook) [25]. App reviews are unstructured texts that contain a wealth of useful information. This information may relate to the app's functionality, quality, problem reports and/or new feature requirements. A prevalent issue in app review analysis research is the categorization of app reviews into different groups, with the objective of utilizing classification techniques to convert the numerous app reviews into practical insights [26]. For example, to categorize the app ratings for the Uber app, Sharma et al. [27] manually created a taxonomy using machine learning methods (such as decision trees and random forests).

Similarly, Bhatia et al. [28] developed an optimized app review classification method using supervised machine learning techniques (polynomial plain Bayes, etc.), automatically classifying app reviews into bug reports, feature requests, and drawbacks and enhancement requests related to requirements engineering. In a recent study, Malgaonkar et al. [29] developed a new method for automatically generating dynamic taxonomy for automated user comment categorization using natural language processing, feature engineering, and lexical disambiguation. The method was used on My Tracks' app review set to validate its feasibility. While these works classify reviews into certain interest areas (or groups), they still fail to address the question: "In what order should the pressing requirements of users present in app reviews be implemented?" To the best of our understanding, there is a dearth of research pertaining to prioritization techniques utilized in the application domain to convert user reviews into practical knowledge.

1.2. User requirements prioritization

Requirements discovery is essential for product positioning and strategic market development [30]. To determine user opinion on a particular feature, Dąbrowski et al. [31] developed a method to find salient issues expressed by users based on user sentiment associated with app reviews and the frequency of feature requests in those reviews. Malgaonkar et al. [32] utilized metrics such as entropy, frequency, TF-IDF, and sentiment analysis as variables for heuristic functions and compared the performance with another regression-based prioritization technique developed. Compared with previous work, this study results in 4 % and 185 % improvement in accuracy and time, respectively. Due to the proliferation of useful reviews, many studies have developed semi-automatic and automated techniques to assist application developers in using online user reviews. Here, we will briefly describe some recent studies. Chen et al. [33] developed a domain-dependent user requirements mining framework to facilitate mobile app quality upgrades, and the study demonstrated the framework's effectiveness for product quality improvement through 265 version update cases of 15 popular mobile apps. Consensus algorithms in distributed systems have also been used in user demand prioritization studies. For example, Etaiwi et al. [15] used consensus algorithms to help developers prioritize user reviews, and the study compared ranking results with reviews manually ranked by application developers and found a strong correlation between the two.

In another study, Yang et al. [34] proposed a novel user requirement prioritization method that extracts requirement phrases for app functionality from user reviews, calculates features such as frequency of occurrence ratings, and then automatically predicts higher-priority requirement phrases based on these features. Similarly, Kifetew et al. [13] proposed a method, ReFeed, which computes a set of relevant user-feedback for each requirement and extracts quantifiable attributes from such user-feedback that are relevant for prioritizing requirements to compute a ranking for each requirement. However, this method relies on domain knowledge in the form of ontologies. In a recent study, Dąbrowski et al. [11] evaluated and compared three existing tools to support requirements prioritization and found that the effectiveness of these methods in the new dataset was lower than the originally reported results.

1.3. XAI study in software engineering

Most ML and DL algorithms are labeled "black boxes" by academia because their underlying patterns are complicated and difficult for humans to describe and verify [35,36]. This opacity creates the need for XAI algorithms, which emphasize developing and utilizing tools and techniques to dismantle the black box by generating human-understandable and transparent explanations of AI decisions. XAI methods have been widely used in healthcare, industry, transportation, software engineering, and other fields [37]. Among them, the success of software engineering projects relies heavily on complex decisions (such as which of the user requirements should be implemented by app developers first) [38]. While various automated development tools based on machine learning have been able to help app developers extract useful insights from large amounts of information to support decision-making, they do not understand the reasons behind it, which often leads to distrust in the tools [39,40]. Existing literature has initially investigated how to use the interpretation of predictions provided by XAI to support software engineering tasks. For example, Chazette and Schneider [41] surveyed 107 end-users to assess the relationship between interpretation and transparency and to analyze its possible effects on software quality. Tantithamthavorn and Jiarpakdee [42] presented three successful cases of using XAI in software engineering to solve the problem of software defect prediction models, and the study confirmed that XAI can enhance the practicality, interpretability, and operability of software analysis. Systems need to collect and process user-generated information to provide better decision support, but this data collection means that the realm of user privacy is increasingly under threat. By investigating app reviews about privacy issues, Brunotte et al. [16] used XAI to reveal informational interpretations about the system and its behavior as a means to increase user privacy awareness. Jamasb et al. [43] proposed an interpretable software requirements classifier (the core is the LIME algorithm) to study the applicability of XAI in software requirements classification. Experiments show that XAI can be used to help software developers better understand the prediction mechanism of classifiers.

While previous work has proposed different methods for prioritizing user requirements, they do not completely address app developers' difficulties with prioritizing due to the volume of app reviews. These studies are more likely to use traditional statistical methods or machine learning to prioritize user requirements, which may be somewhat outdated, and the opacity of machine learning methods may make it difficult to understand how the models work. Furthermore, it has been observed that certain research studies consider positive and negative reviews to be equivalent in terms of prioritizing requirements improvement. However, other studies have demonstrated substantial differences between the two in terms of customer responsiveness [8]. Similar to our work, Dalpiaz et al. [44] used more general language features (such as dependency types) to build an interpretable ML classifier for requirements engineering, which outperforms classifiers using high-dimensional feature sets on validation datasets and is more interpretable, but they did not prioritize the categorized requirements. Our work aims to address this problem. In this study, we targeted negative reviews and considered user-reported bugs, feature requests, and feature enhancements as user requirements, where user requirements obtained based on unsupervised machine learning (topic modeling) will be used to explore the relationship with ratings and then prioritize user requirements by using more advanced XAI.

1.4. Proposed framework

For some well-known apps, hundreds or thousands of reviews are attached to each release. How to efficiently capture user requirements from the rapidly growing number of app reviews has become a major challenge for information system developers and designers [34]. Therefore, developers need a framework that uses app reviews as input to quickly identify and prioritize requirements, enabling them to make better informed strategic judgments. In light of this, our study suggests such a framework. It integrates multiple methods (e.g., topic modeling, sentiment analysis, network analysis, and XAI) to perform data-intensive analysis of app reviews. The suggested framework is illustrated in this section and depicted in Fig. 1. Three primary stages make up the suggested structure. Data gathering and pre-processing make up the first step. Analyzing and visualizing data is the second step. The creation of a prediction model and model interpretation are the last steps.

The first step is obtaining app store resources (e.g., app descriptions, changelogs, reviews, downloads, rankings, and ratings) to use big data analytics in app development and design. For example, developers may leverage customer input from app store reviews, which are quick and free, to make their apps betters [45]. Data sources from various operating systems (e.g., iOS and Android) need to be considered in the data collection process. This is because there may be differences in the functionality and interface design of the same app in the app stores of different operating systems [46,47]. App data for android users are available on various app stores, such as Google play, while app data for ios users are only available on the Apple App Store. This data can be obtained manually, for example, through APIs, and collecting data from various app stores may be facilitated by using certain data mining tools, such as the BeautifulSoup (BS) web scraper [48]. The data collected may contain structured, semi-structured, and unstructured data, which must be pre-processed in order to be loaded into the relevant data platform [49].

The second step is examining the data using various techniques, including statistical analysis, sentiment analysis, text mining, and network analysis. These analysis methods present findings in a form easily understood by app designers and have received a great deal of usage and discussion in the field of information systems [50,51]. Surveys and reports analyzing various data may provide fact-based decision assistance for important corporate activities, such as customer service and payment plans [52].

In the final step, different predictive models and some model interpretation tools can be used to explore the relationship between various variables, such as version updates and downloads [53] and user sentiments and ratings [54]. Researchers across disciplines have been very interested in machine learning as a tool to extract information from data systematically [55]. Utilizing machine learning to extract valuable yet non-obvious or unobservable patterns and knowledge from large datasets aids organizations in their management and decision-making processes. In contrast to traditional statistical modeling, which requires a thorough grasp of the app domain, machine learning takes advantage of the trade-off between data availability and domain expertise. Flexible models may choose relevant factors and exclude uninformative ones using a huge quantity of data, but this progress comes at a cost. Complex models function imperceptibly and may not perform optimally according to researchers' expectations [56]. For example, the algorithm behind the Apple credit card was found to be gender prejudiced [57]. Therefore, model interpretation can help researchers gain insight into model-based predictions to better and more quickly understand the data being analyzed. Based on this framework, app designers can integrate app store data into their development, innovation, and maintenance processes.

The rest of the paper is organized as follows: Section 2 details the methodology used in each step of the app store analysis framework, using the example of user reviews of video conferencing apps. Section 3 presents the specific results of each analysis step. Section 4 provides an in-depth discussion of the findings and validity threats, provides conclusions, and defines future research directions.

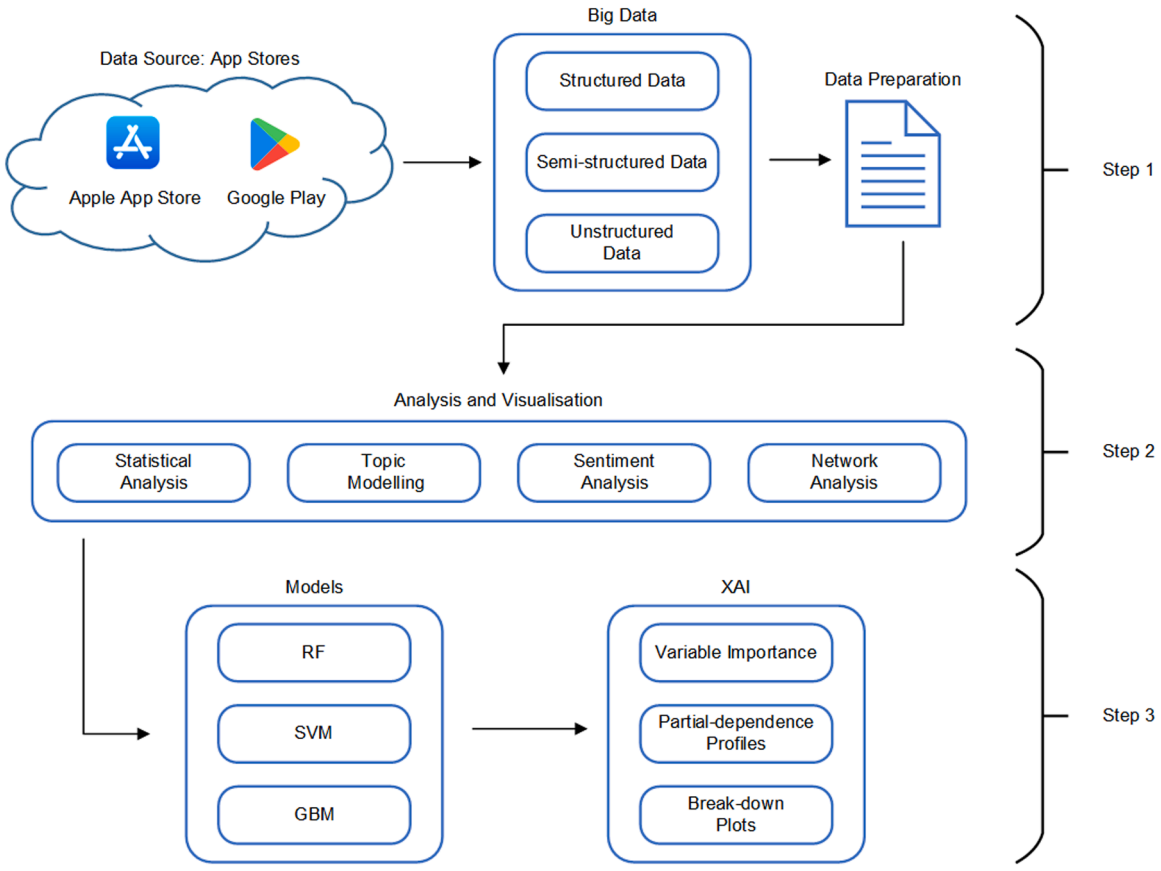


Fig. 1. An app store analysis framework.

2. Materials and methods

2.1. Step one: app reviews collection and pre-processing

In this study, we will demonstrate the proposed analysis framework. Specifically, we will identify the key requirements of the users and their priorities based on the collected user reviews of the WCS. Due to the explosion of COVID-19 and the ensuing ban on home, more and more people are adopting video conferencing as a means of communicating or attending business meetings [17]. Moreover, WCS has begun to be utilized to assist with work, community, family, and friend-related daily tasks [18]. This study selected three video conferencing apps from the "business" category of the Apple App Store, namely Microsoft Teams, ZOOM Cloud Meetings, and Google Meet. We selected the three video conferencing apps mentioned above for our study because they are the most popular in the current market [58]. In addition, these three video conferencing apps have also been analyzed and compared in many studies, and they have become typical representatives of research on video conferencing apps [59–61].

We collected data from the Apple App Store (such as dates, reviews, and ratings) for the above three apps from the US, covering the period from 1st of January 2019 to 31st of June 2022. The data collection process was conducted by using a programmable code written in Python. Apple App Store data was used for market size reasons and is commonly used in related studies [62,63]. Specifically, we used China's Qimai data platform (www.qimai.cn) to collect data. Qimai Data is a professional mobile app data analysis platform based in China that offers multidimensional data on apps available in various countries' iOS and Android app markets [64]. The execution process of the crawler code we wrote is: 1) use the requests library in the Python language to launch HTTP requests to the Qimai data website site, 2) use the lxml package to parse the obtained web content, and 3) save the parsed data as a csv file. A total of three separate datasets were collected, each containing data relating to one app. Next, we put the collected data sets together. After that, the combined data were examined and cleaned up, and any duplicate reviews were eliminated. There were 38,669 reviews in all that were selected and qualified for this study. We assume that the quality of collected user reviews can meet the research needs well. Fig. 2 shows an example of the collected Apple App Store app reviews. We then filtered out non-English reviews, cleaned up the text by eliminating numbers and punctuation, and converted all text to lowercase. After dividing the texts into separate words, we tokenized the terms. After that, stop words were eliminated, and data was stemmed to reduce the data's diversity and increase the findings' accuracy. After pre-processing the obtained user reviews, they are next analyzed for sentiment analysis to explore the different sentiments in the user reviews.

2.2. Step two: sentiment analysis, topic modeling and network analysis

Sentiment analysis is a crucial technique for assessing the efficiency of app operations and service management and is already widely used in the processing of app reviews [65]. Most of the urgent requirements expressed by the users focused on in this study were present in the

Fig. 2. An example of app reviews in the Apple App Store.

negative reviews, so sentiment analysis was employed to obtain the negative reviews from reviews. SentiStrength was applied to identify and pick out negative reviews, which is a very well-liked method for analyzing the range of emotions in each online review from different users [66]. SentiStrength separates a text into sections and then assigns positive and negative values to each section based on the words or phrases and other linguistic information, such as knowledge of grammatical structures, because psychological research indicates that individuals may feel both positively and negatively about the same content [67,68]. The values within this range are as follows: −1 for negative values and 1 to 5 for positive ones. SentiStrength searches its vocabulary for each word or phrase to determine its sentiment, using the corresponding sentiments (positive and negative) if found; if not, a value of zero is used (no sentiment) [69–71]. At the end of the analysis, reviews will be marked as any of the following three emotions: positive, negative, or neutral. In this study, the reviews that are eventually marked as negative are the key messages needed for this study and will be singled out for subsequent analysis.

To measure the accuracy of SentiStrength for sentiment analysis on the dataset of this paper, we randomly selected 100 reviews from the obtained reviews, manually determined the sentiment of these reviews, and labeled them as positive, neutral, or negative. The 100 reviews were then sentiment analyzed using SentiStrength to obtain the corresponding sentiment category for each review and to obtain the corresponding confusion matrix (Table 1). According to Table 1, the number of comments correctly classified by SentiStrength is 86, among which the are correctly classified into positive, neutral, and negative categories are 32, 29, and 25, respectively. Therefore, the accuracy of SentiStrength's sentiment classification for 100 comments is 86 %, which also shows that this tool can meet the sentiment analysis requirements of this study to a higher degree [72].

Next, we will perform text mining on unstructured forms of negative reviews obtained using sentiment analysis. Text mining seeks to turn unstructured user-generated data into understandable and useful information by extracting insights from text documents. It has been used extensively to analyze online review data [73]. To conduct text mining, a structural topic model (STM) was performed for topic modeling [74]. First, STM enables researchers to introduce document-level covariates (e.g., whether a review is positive or negative) into the topic prevalence parameter that affects the proportion of document topics. Therefore, researchers can easily use STM to determine how the proportion of document topics varies with different levels of covariates. Second, STM also enables researchers to add covariates at the document level to topic content factors that influence topic-word distributions. Third, STM is an expansion of the correlated topic model, which allows for the correlation of topics, allowing us to assess the connections between topics quickly. In the STM model, K the number of topics is as shown. D, the number of documents. Topics proportions, θ_d, can be correlated and the topical prevalence can be impacted by covariates, X, through a regression-type model $\theta_d \sim LogisticNormal(X_\gamma, \sum)$. For each word, w, the topic, $z_{d,n}$, is drawn from a response-specific distribution. Conditioned on the topic, a word is chosen from a multinomial distribution over words with parameters, β_k, where $k = z_{d,n}$. The topical content covariate allows word use within each topic to vary according to content. Readers can refer to probabilistic topic models [75] for further comprehension of the method.

We used the *stm* package [76] in R to build the model where the content of the app reviews was used as document input. App review date was used as a variable in the prevalence function. Next, the number of topics K, a crucial STM parameter that aids in a thorough evaluation of the modeling findings, was identified [77]. We started with several statistical diagnostic tools by assessing changes in semantic coherence and other measures of goodness-of-fit estimated by STM with the number of topics ranging from 3 to 15. We used the R packages *stm* and *furrr* [78] to assess the STM. It is important to note that semantic coherence reveals the internal consistency of the subject by determining if the most probable terms in each topic tend to appear together in the text [79]. Semantic coherence often decreases as the number of topics rises.

An increase in STM goodness-of-fit is associated with a higher held-out likelihood, higher lower bound, and lower model residuals [76,80]. Graphs of these statistical measures are shown in Fig. 3 for a wide range of topics ranging from 3 to 15. A comparison between indicators $K = 10$ was selected for the dataset. With this $K = 10$, we have semantic coherence locally maximized at – 133.06, the held-out likelihood becomes flat at −5.47, and residuals at a relatively low value at 3.15.

Then, a network analysis was conducted to confirm the relationships between topics derived from the topic modeling. Social network analysis is a method for gathering and examining data from online social networks like Facebook. Based on graph theory and mathematical modeling, social media research often uses it to recognize and clarify social network patterns, model connections, and follow the evolution and dynamics of activity on social networking sites [81,82]. It has been used in a variety of fields such as social, travel, information systems, innovation, and more, enabling managers to keep track of online conversations about important information and knowledge [83–85]. A graph is created to investigate a particular link and note-based social network structure in social network analysis. Nodes represent individual elements of a network, and links between nodes relate to social communication, connections, and interconnections between the components of a social media network [86]. In this study, the open-source graphical and network analysis program Gephi, created by the French academic institutions Sciences Po and Linkfluence and extensively utilized in the domains of social network analysis, biology, genetics, and other disciplines, was employed [87,88]. A keyword list was created using the top 20 keywords under each topic to visualize the links between topics and imported into Gephi to create an undirected network. Here, we assume that the keywords under each topic are representative of that topic and that the links between keywords reflect the links

Table 1
Confusion matrix.

Confusion matrix		Prediction		
		positive	negative	neutral
Reference	positive	32	3	2
	negative	2	25	1
	neutral	4	2	29

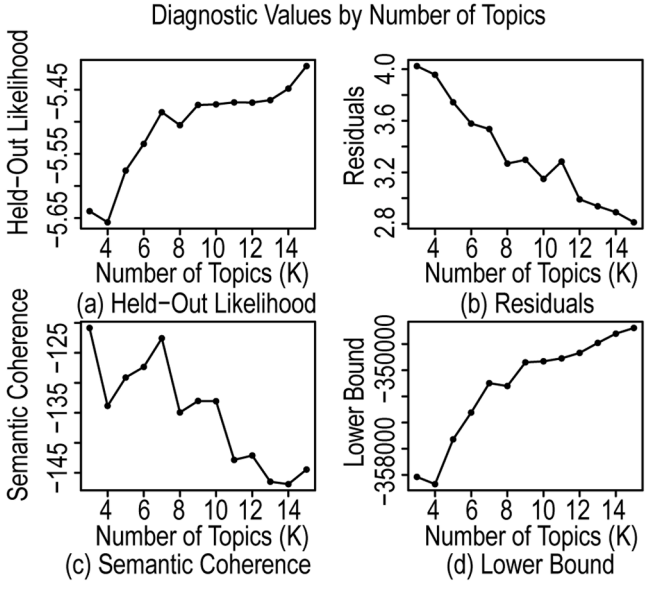


Fig. 3. Diagnostic values by the number of topics.

between topics. The network was visually mapped through a computational tool, displaying the relationship's organizational structure, tight integration, and meaningful keyword linkages [89–91]. Following network analysis, we will further explore negative reviews using machine learning and XAI approaches.

2.3. Step three: model exploration analysis

2.3.1. Modeling algorithm

We used machine learning techniques to analyze the impact of changes in the potential predictor variable (topic proportion) on the continuous dependent variable (ratings). The aim is to construct a regression model that predicts the star rating of a review based on the distribution of topics in the review content. This is because there is a correlation between ratings and reviews, and this correlation is the focus of our attention. Here, we assume that the samples in the dataset are independent of each other and come from the same distribution. The dataset input to the machine learning model consists of two parts, the independent variable (x_i) and the dependent variable (y), where the independent variable is the probability distribution of topics in each review obtained after topic modeling of user reviews, for example, if 8 topics are identified, then at the same time the probability distributions of these 8 topics in each review, i.e., the 8 independent variables, are also available. For the dependent variable the star rating of each review is taken as the dependent variable, which takes values in the range of 1–5. Table 2 shows an example of the data in the dataset.

Prior to model construction, we split the dataset into a training (80 % of observations) and a test (20 %) set. Only the training set was used in the model construction process. Subsequently, we trained six different machine learning algorithms: classification and regression tree (CART), lasso and elastic-net regularized generalized linear models (GLMNET), bootstrap aggregating (BAGGING), gradient boosting machine (GBM), k nearest neighbors (KNN), support vector machines (SVM). All of these algorithms were selected for implementation in regression analysis. All predictor variables were centered and scaled prior to analysis in order to reduce skewness and stabilize variance. We used the caret::train() function to train each model, and a 10-fold cross-validation with three repeats was set to improve the model's generalization [92]. All predictor variables were centered and scaled during model training to stabilize variance and reduce skewness. Because the regression problem was selected, we have chosen root mean squared error (RMSE) and the coefficient of determination (R^2) as the test metric. RMSE measures the average deviation of predicted values from actual values, calculated as the square root of the mean squared error (MSE). R-squared is a statistical measure representing the proportion of the variance in the dependent variable that is predictable from the independent variables. The equations for each are as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2},$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2},$$

where n is the number of observations, y_i is the actual value of the target variable for observation i, $\hat{y}_i$ is the predicted value of the target variable for observation i, $\bar{y}$ is the mean of the actual values of the target variable.

The next step in model interpretation is to select the best-performing model, as the model's predictive accuracy is critical to the prioritization of requirements. Models that perform poorly on the test set do not fit the regression relationship between topics and ratings well, and this can lead to bias in the results of the next model interpretation, which ultimately affects requirement prioritization. For brevity, all algorithms use default (automatically estimated) algorithm parameters and do not emphasize hyperparameter tuning. Nevertheless, we believe that the trained algorithms will give meaningful results when interpreting the model.

Table 2
Examples of datasets used for machine learning models.

Topic 1 (x_1)	Topic 2 (x_2)	Topic 3 (x_3)	…	Topic n (x_n)	Rating (y)
0.0025	0.0146	0.6258	…	0.2675	4

2.3.2. Model interpretability

Although sophisticated, machine learning models do not reveal how they arrived at their results [93]. Due to this, several algorithms have been created that try to explain how black box models structure, and these algorithms have applications in a number of different domains [94,95]. We used the *explain()* and the *model_parts()* functions of the DALEX package to gain further insight into the model [96]. DALEX is a consistent collection of interpreters for prediction models, and the method presented is model-independent, which means that it can be used for any prediction model or collection of models [97]. Here, we assume that the importance of features in the explained model can be inferred from the behavior of the explained model. We calculate variable importance using the dropout loss of RMSE - the predictive RMSE of the model increases when specific variables in the dataset are perturbed. The significance of the variable in accurately forecasting the result increases with the RMSE increment. A higher dropout loss (an increase in RMSE) indicates that a particular variable is more important to the correct prediction. In addition, we employed partial dependency graphs, a widely utilized technique for examining black-box models at the dataset level, which utilizes the average profile of all observations and assumes constant values for all remaining predictor variables to illustrate how model predictions differ across particular variables [98]. These charts, therefore, provide a broad view of how model predictions vary depending on certain characteristics, displaying the dynamics of the machine learning response function dependent on the values of one or two interested input variables while averaging the other input variables [99]. These representations are simple to understand, quantify the model's sensitivity to the variables, and are generally intuitive even when they do not precisely reflect the effects of capture. All the above analyses were performed in R version 4.1.3 (R Core Team, 2022).

3. Results

3.1. Results of sentiment analysis

Managers must identify which areas of the offered product or service are vulnerable to criticism and customer discontent in order to enhance product quality and service management. Negative reviews have proven to be an effective source for assessing information about service delivery, quality, and customer requirements [100]. This may aid developers in understanding consumers' concerns regarding subpar features and uncovering novel demands [101,102]. For instance, software engineers can analyze the frequency of negative reviews and how the frequency of occurrence fluctuates over time [103,104]. Thus, in this study, we will identify and prioritize user requirements based on negative user reviews. The sentiment analysis revealed that a significant proportion of the 38,669 user reviews exhibited a neutral or positive stance. While there were relatively few negative sentiment (23.6 %, 9107) reviews. The negative sentiment evaluations derived from the comprehensive sentiment analysis were deliberately chosen for further examination in order to obtain a deeper understanding of the pressing concerns articulated by users in these complaints and disillusionments.

3.2. Topic modeling for reviews with negative sentiment

3.2.1. Topic summary and labeling

Through the analysis, 10 topics were identified applying the STM technique. Next, we chose frequency-exclusivity (FREX) terms and

words with high probability (Highest Prob) from the STM results to create the labels. FREX quantifies the importance of words based on their total frequency and the degree of exclusivity to topics, which might result in semantically more comprehensible topic representations. Thus, the words generated by FREX statistics are mainly used to label topics, but given that FREX statistics sometimes place uncommon words high on the list [105], we also referred to the Highest Prob words. The name of each topic was marked manually through group discussion. To ensure the appropriateness and reliability of the topic names, we also checked the first 10 representative reviews under each topic. Using STM techniques, 10 topics from this research were retrieved. The outcomes of the topic modeling are shown in Table 3, along with the labels given to each topic.

Of the 10 topics generated by the topic modeling, we decided to exclude topics 4 and 7 because they were relevant to the educational context, and most of the reviews under the topics were students' complaints and negative experiences with online classes, such as hatred and boredom. The interesting point is that the star ratings for these reviews are generally low as if students would take out their dissatisfaction with the online classes on the tools they use, even if there is nothing wrong with those tools. However, these were not part of the user requirements and were excluded. Of the remaining topics, the top three with the highest proportions were audio and video quality (13.6 %), meeting and account passwords (13.2 %), and features (11.6 %), accounting for 38.4 % of all negative reviews.

Table 3
Topic modeling results.

Topic No.	Topic Label	Top Words	Topic Prop. (%)
1	Features	Highest Prob: screen, chat, update, option, feature, call, background FREX: background, option, chat, view, notif, screen, participate	11.6 %
2	Hacker issues	Highest Prob: issu, access, hack, secur, fix, privaci, requir FREX: hack, secur, network, hacker, issu, microphon, access	5.2 %
3	Customer service	Highest Prob: worst, time, servic, support, call, month, hour FREX: custom, servic, cancel, support, worst, upgrad, money	7.9 %
4*	Complaints about online school	Highest Prob: hate, school, meet, class, onlin, kick, time FREX: ruin, hate, onlin, life, class, kick, bore	13.3 %
5	Audio and video quality	Highest Prob: terribl, horribl, audio, video, qualiti, connect, hear FREX: horribl, qualiti, terribl, audio, sound, glitchi, laggi	13.6 %
6	Steal data	Highest Prob: steal, camera, data, share, facebook, info, comput FREX: steal, info, facebook, data, consent, camera, sell	9.8 %
7*	Experiences regarding Home-schooling	Highest Prob: work, teacher, annoy, student, star, delet, talk FREX: student, assign, rate, talk, star, everyday, work	11.8 %
8	Meeting and account passwords	Highest Prob: meet, join, sign, time, password, account, email FREX: password, sign, error, code, invit, link, wrong	13.2 %
9	Office platform	Highest Prob: easi, crash, platform, commun, complet, load, compani, FREX: platform, crap, commun, offic, batteri, easi, tool	8.5 %
10	Mute and unmute	Highest Prob: peopl, call, mute, reason, time, kid, unmut FREX: reason, kid, mute, unmut, child, stop, peopl	5.0 %

Note: "Topic Prop." Represents the estimated proportion of each topic; Numbers marked with an * indicate excluded topics.

3.2.2. Topic trend analysis

This study conducted a trend analysis of the eight topics obtained after deletion. Fig. 4 shows the estimated mean proportion of change for selected topics. As shown in Fig. 4(a) and 4 (g), The trend in the proportion of topics in the features and office platforms is similar, both gradually decreasing from September 2019 to the lowest level in April 2020. This finding may indicate that with the high frequency of updates to the three videoconferencing apps during this period (8.1 times/month, compared to an average of 6.9 times/month during the observation time frame), most of the feature issues encountered by users in the office are gradually resolved and user concerns decrease progressively. In March and April 2020, Zoom was heavily criticized for its various privacy and security practices [106]. During this period, users' concerns about privacy issues rapidly increased to their highest level (i.e., almost 25 % of all negative reviews), which also reduced the level of concern about other topics such as meetings and account passwords (see Fig. 4(f)) during the same period. Still, Zoom's rapid response to this crisis allowed public trust to be restored quickly and the level of user discussion to drop rapidly. During the period from September 2019 to January 2020, there was a limited amount of discussion among users regarding the topic. Upon examining the original reviews from that time frame, it was discovered that the majority of users expressed concerns about the issues they faced while at the office. This makes sense, as WCS was initially adopted primarily by businesses to facilitate business-to-business interaction and distributed teamwork [18,107].

The proportions of hacker issues, customer service, and mute and unmute have been relatively flat over time, but this seems to reveal that these issues are not being better addressed with the new version. Among them, one of the main threats to video conferencing apps is the issue of hacking [108]. Audio and video quality, as the topics that account for the highest proportion of all negative reviews, the degree of user attention changed significantly before May 2020, and then the proportion of this topic showed a downward trend. This may reveal that higher-quality audio and video experiences are available to people as information delivery technologies evolve [109].

3.3. The results of network analysis

Next, we used Gephi to automatically analyze the reviews and visualize the results to understand the relationship between key requirements. The ForceAltas2 layout technique was used to create network diagrams because it offers greater measurement quality than other layout algorithms [88]. Fig. 5 shows the network diagram generated by the software. The clusters represent the topics obtained by topic modeling, and the nodes of each cluster are composed of the top 20 keywords under the topic. Fig. 5 shows that central nodes (e.g., "people", "computer", "privacy", "call", "time", "account", etc.) appear more often on several topics rather than just on one. Interestingly, most of the topics are connected in pairs through central nodes, such as mute and unmute with hacker issues and customer service with meeting and account passwords, revealing that potential pairwise correlations between requirements.

"Time" is the most interconnected central node among these, connecting four topics (Customer service, Meeting and account password, Audio and video quality, Mute and unmute). This underscores the criticality of time in WCS. Time wasted due to system problems and untimely customer service can easily lead to negative user sentiment [110]. The keyword "frustrating" indicates user frustration with the features and meeting and account password clusters. The topic with the highest number of central nodes is mute and unmute, which reflects the fact that this topic is a core factor in user dissatisfaction in WCS. In contrast, the office platform is not closely related to other topics because the

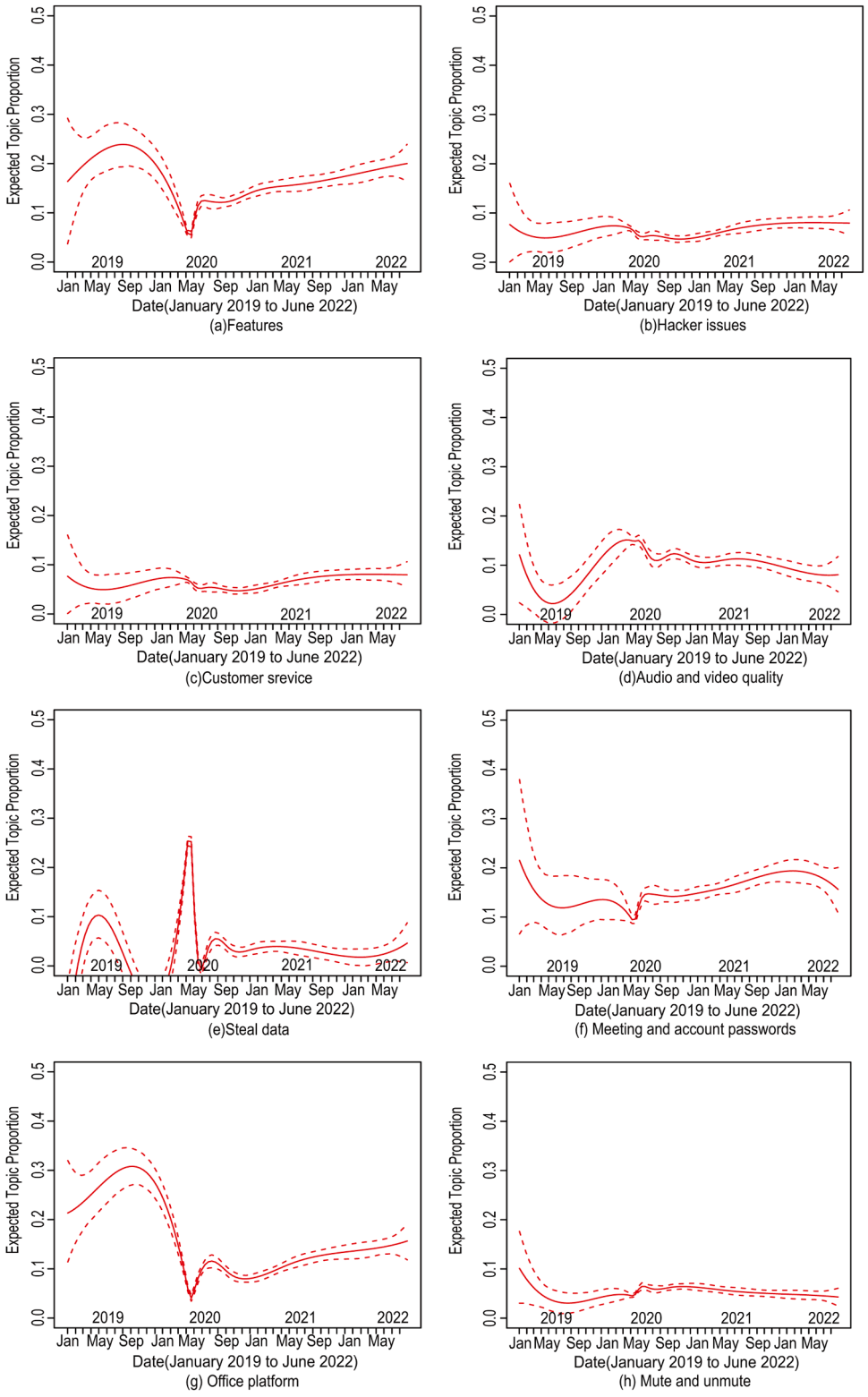


Fig. 4. Trends over time for the different topics estimated by the STM model (the popularity of each topic is plotted as a smoothed function of the time the review was posted, with the dashed line indicating a 95 % confidence interval).

keywords underneath are unique, unlike steal data and meeting and account passwords, which share more keywords with other topics. It is worth noting that the network has a modularity index of 0.774 but also has a large mean path length (4.726), whereas a shorter mean path length in a network allows for fast information transfer and low cost [111], suggesting that although clusters within the network are relatively concentrated, inter-cluster communication is loose and inter-group correlation is low, as the image demonstrates as well, the majority of remaining keywords do not connect to the topics that have been detected, which may be due to the fact that many user-written negative reviews address various facets of the videoconferencing program.

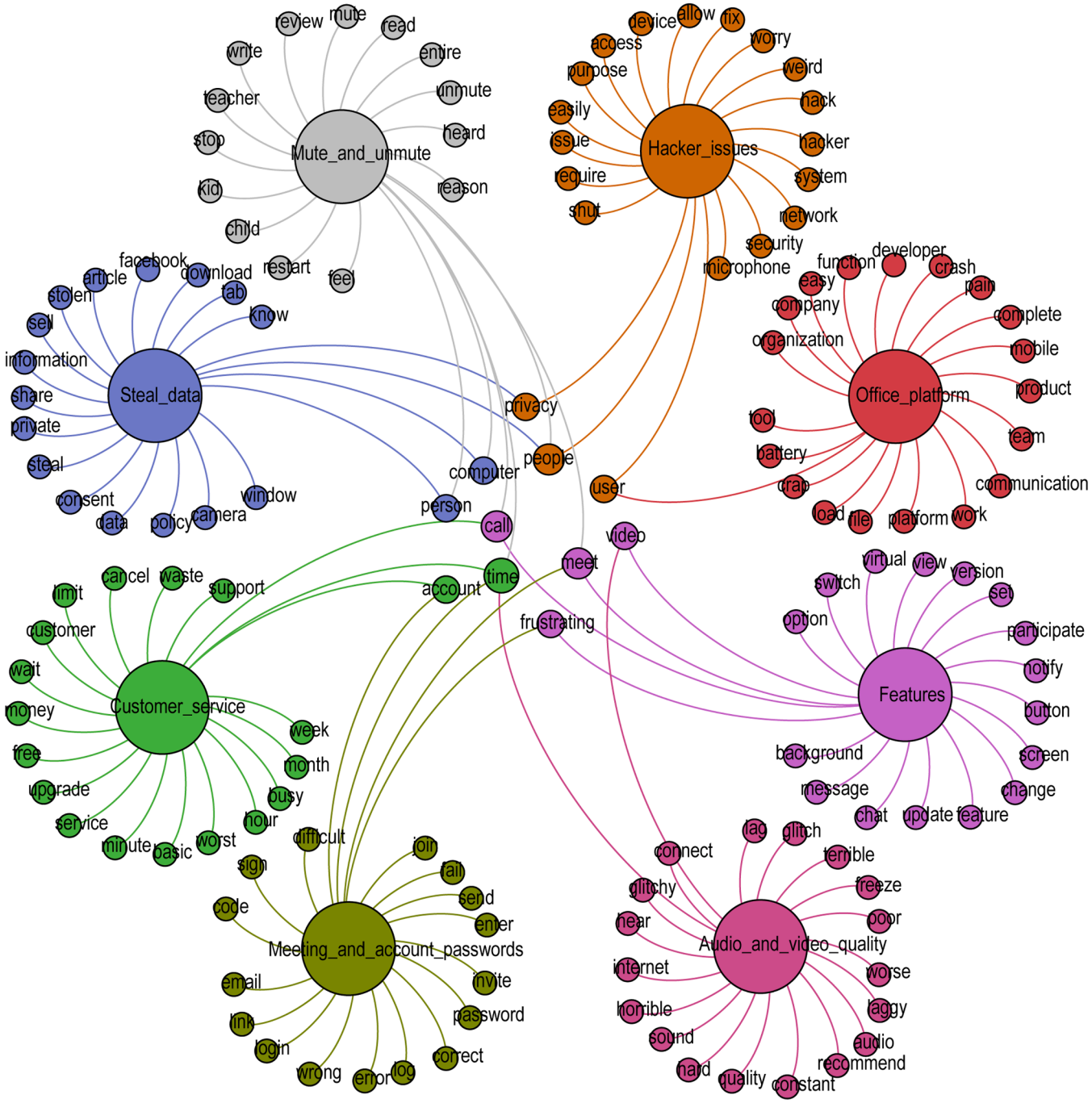


Fig. 5. Topic network for negative reviews.

3.4. Results of model exploration

3.4.1. Performances of prediction models

To reasonably evaluate the performance of each model, we use RMSE and R2 as measures of model performance [112]; a lower RMSE value is associated with higher model accuracy, with a value of 0 indicating a perfect fit, while a higher R2 value is associated with a better fit, with a value of 1 indicating a best-fit model. The performance of each test model on the training set is shown in Fig. 6. Both RMSE (1.24) and R2 (0.09) values show that the gradient boosting machine (GBM) outperforms the other models.

3.4.2. Variable importance and partial dependence

As described above, the experiments determined that GBM is the best-performing model on this dataset. Fig. 7 illustrates the significance of the explanatory variables incorporated in the GBM model. The RMSE value of the GBM model is denoted by the vertical dashed line on the left. The bars for each explanatory variable start with the RMSE value of the GBM model and end with the (average) RMSE value calculated using data with permuted values of the variable, the length of the bar representing the importance measure for each explanatory variable [98]. The office platform is the explanatory variable with the greatest impact on model prediction. This suggests that the model believes that a change in the proportion of this topic will result in the greatest change in rating. The next two important explanatory variables are features and steal data. It is worth noting that the latter is far less important than the former. The explanatory variables (Audio and video quality, Mute and unmute and Meeting and account passwords) ranked in the middle have

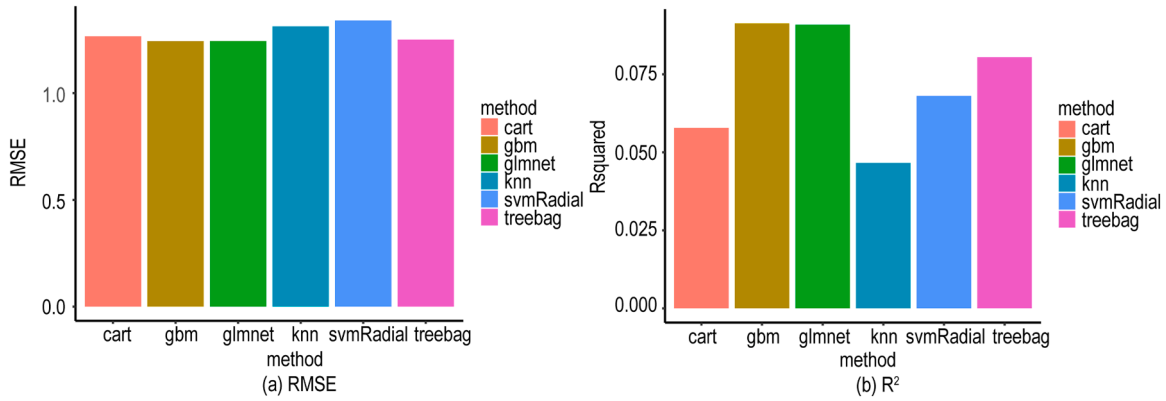


Fig. 6. Performance of different models on the test set.

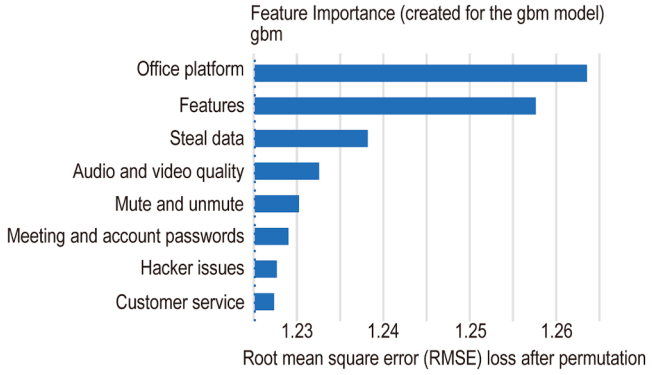


Fig. 7. Variable importance measures for the explanatory variables included in the GBM model.

a small difference in their contribution to the model. The last-ranked customer service has almost the same lower variable importance as hacker issues. It is important to note that the ranking of variable importance should not be taken directly as the order in which user requirements are processed. These explanatory variables considered here are thematic representations of negative sentiment, and variable importance may be influenced by the negative sentiment itself. For example, the majority of negative reviews pertaining to customer service award a single star. As a consequence, variations in the value of this explanatory variable have minimal or negligible impact on the dependent variable, which is evidently the user's intensely negative sentiment towards the subject matter. Therefore, explanatory variables with low variable importance do not imply insignificance here. The diversity of user ratings for the explanatory variable decreases with the importance of the variable in this study; therefore, this requirement must be addressed immediately.

Next, we use a partial-dependence (PD) plot to show how the expected value predicted by the model behaves as a function of the selected explanatory variables. Fig. 8 shows partial dependence plots of the eight discovered explanatory variables used to build the GBM model. They illustrate the marginal effects of each explanatory variable on the ratings after considering the average joint effects of the other explanatory variables. Even while it may not provide a full explanation, it may indicate general tendencies and give a solid foundation for interpretation [113,114]. The two explanatory variables that have a positive correlation with the rating are office platform and features, which means that the higher the proportion of their topics, the higher the rating. The fluctuation of the office platform value (the proportion in negative reviews) among all explanatory variables will make the maximum fluctuation of the predicted value (rating), which also confirms the previous results of variable importance. At the same time, it shows that as the proportion of this topic increases, the user attitude will be relatively better. This may be related to the positive attitude of people working from home [115].

The PD configuration files for the four variables (Customer service, Hacker issues, Meeting and account passwords, and Mute and unmute) are similar. They all fluctuate in a small range around the value of 1.75. Among them, the explanatory variable with the least impact on the model prediction is customer service, indicating that the topic has the lowest rating diversity. This shows that when the topic is mentioned in negative reviews, the negative attitude of users will remain almost unchanged. The results are consistent with previous studies, indicating that customer service quality will significantly affect the satisfaction level of mobile apps [116,117]. The fluctuation of the average predicted value of hacker issues is almost the same as that of the customer service, indicating that both will cause users to give lower ratings, but users are more tolerant of hacker issues. This is obvious because when users seek customer support for hacker issues and do not receive satisfactory answers, they are more likely to be dissatisfied with customer service. In addition, two topics (Meeting and account passwords and Mute and unmute) have been indentified that can cause low user scoring behavior. Upon browsing related reviews, it was found that the incorrect password used when attending the meeting and logging in to the account caused users a lot of trouble.

The automatic unmute of the system is also a key issue in user feedback. Interestingly, there is a clear negative correlation between audio and video quality, steal data, and ratings. When they reach stability, these variables have lower average predictive values than the other variables, and stealing data has the lowest value. This shows that with the increase of the proportion of the two topics, user attitudes will deteriorate further, and users who talk about steal data will have a stronger intention of low rating. In other words, stealing data is the most pressing user requirement that needs to be addressed today, followed by video and audio quality. This seems obvious, as users tend to be more disgruntled and angrier when their privacy is at risk of compromise [118]. In summary, when improving the quality of WCS, priority should be given to stealing data-related requirements, protecting the user's privacy information, and maintaining the security of their networks. The specific priorities for user requirements are shown in Table 4. We combined Figs. 7. and 8 to rank the user requirements. In Fig. 7, the order of variable importance for each user requirement directly correlates with the initial order of user requirement priority. This is because a lower variable importance indicates a smaller change in rating when the user mentions the requirement in their reviews, i.e., the smaller the change in the user's negative attitude. Afterwards, we observed the trend of rating change for each user requirement according to Fig. 8, and found that steal data and audio and video quality, although they have high variable importance in Fig. 7, the negative attitude of the user is not shifting to better but to worse, where steal data leads to the lowest

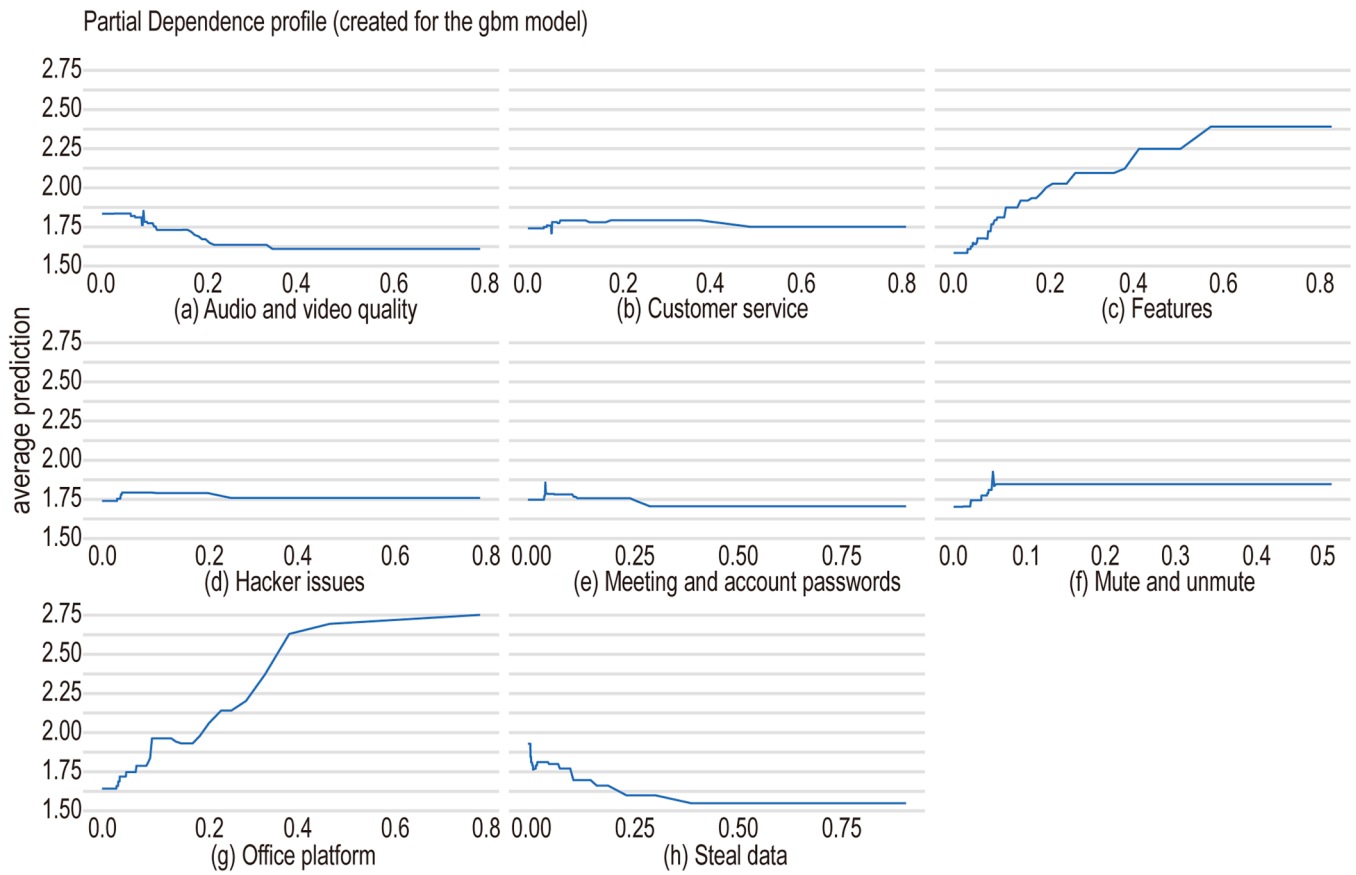


Fig. 8. Partial-dependence profiles for the GBM model and selected explanatory variables for the negative reviews.

Table 4
Prioritization of user requirements.

Prioritization	User requirements
1	Steal data
2	Audio and video quality
3	Customer service
4	Hacker issues
5	Meeting and account passwords
6	Mute and unmute
7	Features
8	Office platform

negative attitude of the user, so we adjusted the initial ordering to put steal data at the top of the priority of the user's requirements to solve. Next comes audio and video quality, and the rest of the user requirements are ranked in the original order to get the final user requirement resolution priority.

4. Discussion and conclusion

Martin et al. [119] used the phrase "app store analysis" to describe the burgeoning research in software engineering using app store data. Although app designers have accessed and analyzed vast amounts of data in app stores such as Apple App Store and Google Store to release and quickly update apps, it is not easy to effectively apply app store analytics to improve user loyalty and satisfaction [120,121].

The findings of this research are intended to help WCS developers improve service quality and product development based on users' pressing requirements. The data analysis began with the extraction of negative reviews. Based on the presence of these negative reviews, eight critical requirements of users in the WCS were identified in an automated manner using topic modeling. These requirements consisted of app bugs, feature enhancements, and new feature requests that users urgently expressed in informative reviews. After that, network analysis was used to examine the relationships between the pressing requirements. We also used machine learning to clarify the connection between these requirements and user rating behavior. Finally, the work used XAI to interpret the best-performing model and generate appropriate priorities based on the interpretation results for the user's urgent requirements.

Among the user requirements observed in Fig. 8, the requirement that causes users to generate the strongest low rating willingness is to steal data. This is consistent with previous research findings. According to Ebrahimi and Mahmoud [122], complaints about privacy and ethics have the greatest negative impact on app ratings. Steal data issues are strongly associated with hacker issues, but user attitudes towards the former are worse than towards the latter (refer to Figs. 5 and 8). Thus, it is crucial to uphold the ethical principles of the enterprise and protect the privacy and security of users. It is worth noting that regarding the popularity of the steal data topic (Fig. 4(e)), the sudden spikes and dips indicate dramatic fluctuations in user requirements and reflect that timely and proper handling by organizations can help alleviate user concerns and reduce negative impact. The next user requirement that needs to be addressed is related to video and audio quality. This is related to the properties of the web conferencing system itself. Other user requirements include customer service issues (leads to negative emotions and sparks online reviews), hacker issues that prevent users from continuing meetings normally, meeting links and login account password issues (results in users being unable to join meetings), the automatic unmute issue which causes user embarrassment, issues with improved or missing functionality (leading to inconvenience in usage) and office problems (i.e., developers may need to address incomplete office components). Notably, when we pre-processed the data, we found

that app developers did not respond to negative reviews promptly, potentially exacerbating the impact of negative reviews shared by users in the app store.

Although the final result of prioritizing user requirements is somewhat obvious and more in line with daily experience, this is also theoretical proof of the correctness of this type of daily experience because daily experience is not always reliable in guiding managers to make decisions. In addition, similar results are mentioned in the literature [116,117,122], confirming the validity of our study. However, our findings diverge from the previous literature in three key aspects: 1) we found the user requirement "meeting and account passwords," whereas most of the previous research on this type of problem has focused on security, with almost no research exploring the user requirement itself (e.g., displaying an incorrect password when logging in). 2) The user requirements of "mute and unmute" have also been overlooked in the previous literature. 3) Some of the user requirements in the findings have been identified by existing research, but no research has been conducted to prioritize their solutions.

4.1. Implications for practice

Unlike apps in many other fields, the WCS was widely used during the COVID-19 pandemic to provide basic video conference services to different populations. However, the significant increase in demand for teleconferencing and the influx of users from different fields (such as the education industry) have put enormous pressure on the WCS. Given these complex and difficult situations, the framework proposed in this study can apply and support the WCS to not only respond to but also deeply understand the unprecedented user requirements. Negative reviews posted by users may have a greater influence than positive word-of-mouth [123]. Therefore, mobile app companies need to adopt proactive strategies to reduce or eliminate negative reviews' impact on their products and businesses. Mobile app companies must understand the impact of negative user reviews from the app store platform and devise scientific and reasonable strategies for the problems behind them to reduce costs and increase efficiency. Since users tend to compare products and services offered by competitors in app store reviews, monitoring competitors in app stores should also be part of a long-term strategy [124]. Competitor data in app stores provides mobile app companies with an excellent opportunity to gain business intelligence and develop better competitive strategies. Knowledge gained by comparing competitor data may help mobile app companies seize market opportunities and avoid risks. To sum up, mobile app companies can only increase their business value by using the big data and big data analysis technologies available in the app store to enhance the quality and worth of their products and services.

4.2. Implications for research

The study emphasizes how difficult it is to analyze app stores because of their complicated and heterogeneous data structures and the sheer volume humans cannot fully handle. Therefore, this research contributes to the existing literature on app stores. Our investigation demonstrates that app developers can better comprehend users' urgent requirements in real-time by using the huge amount of data found in app stores. Previous research on app requirements prioritization is often limited and does not fully address the prioritization of many informative reviews. Some existing automated methods fail to provide app developers with an explanation of the results, which leads to developers' incomprehension and distrust [125]. In this study, we first perform text mining on negative reviews to obtain users' urgent requirements and then prioritize these requirements by explaining their relationship to ratings. Model-interpretation-based prioritization methods are domain-agnostic, do not require manual labeling of large numbers of reviews, and the research can be extended to other categories of apps by replacing the dataset. The identified requirements prioritization may be a useful starting point for requirements specialists and researchers engaged in requirements prioritization activities [13].

4.3. Threats to validity

In this section, we will discuss the potential threats to the validity of our experiments from four different angles: construct, internal, external, and conclusion [126].

Construct validity: Threats to construct validity involve the relationship between theory and experiment, especially if the design of the experiment is chosen to suit the purpose of the study. Different approaches may lead to experimental bias. We mitigated this threat by applying the widely used structural topic model. Secondly, we used the relationship between topic shares and ratings to prioritize urgent user requirements. However, app developers (and other stakeholders) may also choose other approaches to prioritize urgent user requirements (e.g., frequency). The impact of such methods was not examined in this study. Furthermore, following other similar work, our approach assumes that reviews with low star ratings are more reflective of the user's immediate requirements than other reviews. However, there may also be pressing user requirements in highly rated reviews.

Another threat is that the required words taken from the reviews may be combined with other phrases or meaningless terms, which might hurt the topic model's performance and make it more difficult to comprehend the results. Several pre-processing measures were used to mitigate this threat, such as removing emoticons and filtering out stop words. However, this possibility remains, as we did not examine which preprocessing method was the most appropriate; instead, we adopted a general approach.

Internal validity: The threat to internal validity is primarily representative of our model. The empirical choice of parameters (e.g., $\alpha$and β) and the determination of hyperparameters may affect the experimental results. Therefore, we used default parameters and did not perform hyperparameter tuning to minimize this threat. Another threat is whether the results of our study are correctly derived from our data. In particular, the adequacy of the methods used to extract themes. Therefore, we considered a different number of topics around the optimal number of topics shown by the metrics, such as 9. The authors finally agreed by observing and discussing the keywords and representative reviews under these topics.

External validity: External validity is related to the generality of our results. Our dataset consists of over 38,000 reviews posted between 1 January 2019 and 31 June 2022, which may not represent all review types under the videoconferencing app. In addition, as we only considered WCS on mobile devices, the results may not apply to WCS on PC. Although our proposed prioritization method is designed for all mobile apps, there may be differences in user requirements of the same app across different app stores and even for the Apple App Store; we did not study the differences between different categories. However, we believe that our approach is flexible. Therefore, dealing with other similar problems (e.g., replacing datasets) is easy, depending on the prioritization study or app requirements.

Conclusion validity: Threats to the validity of conclusions involve dealing with the relationship between treatment and outcome, especially the capacity to draw the right conclusions about the relationship between treatment and observation. In this research, we used appropriate big data analytic techniques on the data to address our study concerns. We used different metrics, typically to measure the connection between the number of topics and reviews and to predict the model's performance. This is to minimize the potential for biased interpretations that can arise from using a single metric, as interpretations are often subjective.

4.4. Conclusion and future research

In this paper, we proposed an app store analysis framework and

demonstrated its validity by prioritizing user requirements for three popular video conferencing apps. Specifically, we first used a crawler to collect user reviews of Microsoft Teams, ZOOM Cloud Meetings, and Google Meet in the Apple App Store from the Qimai data platform. Second, we performed sentiment analysis on user reviews using the Sentistrength tool to identify negative reviews. Third, STM models were used to automatically identify eight key user requirements underlying the negative reviews. Fourth, the Gephi network analysis tool was employed to investigate the relationships between these pressing requirements. Fifth, six machine learning models were used to determine the link between these requirements and user ratings. Finally, XAI was used to interpret the best-performing models (GBM) and generate appropriate prioritization for users' urgent requirements based on the interpretation results. The study results showed that the eight types of users' urgent requirements were prioritized in the following order: Steal data, Audio and video quality, Customer service, Hacker issues, Meeting and account passwords, Mute and unmute, Features, and Office platform.

Furthermore, the novel prioritizing technique suggested in this research only requires little manual processing. In particular, the priority results of this method are interpretable. In other words, we believe the key to developing an automatic prioritization method is intelligently combining multiple approaches (e.g., topic modeling, sentiment analysis, XAI, etc.). As a result, the approach suggested in this research has significant consequences for the development and maintenance of apps, especially where app developers cannot manually filter all reviews to obtain prioritized insights. It should be noted that our approach is scalable and can be used to develop applications in any other field without requiring extensive manual labor. While a prior study has shown that most mobile app developers utilize user reviews in app stores to account for user requirements, these firms overlook ratings and place a greater emphasis on the number of requirement reviews [33]. We believe our strategy will make it easier and faster for app developers to prioritize user requirements.

Despite our contribution to this study, there are several avenues for future research. Online review systems include false reviews and ratings that do not correspond to reviews [127]. These erroneous reviews may affect the validity of our proposed framework for analyzing app stores. In future work, we will try to identify and delete these reviews in advance, making the analysis results based on the developed framework more practical. The XAI-based prioritization method could be improved to obtain better results for requirement ranking. Furthermore, we will verify the effectiveness of the obtained user requirement priorities for app improvement. For example, we will try to contact the developers of the surveyed videoconferencing apps and provide them with the results of our experiment to obtain their perceptions and satisfaction regarding the prioritization of requirements. An experiment could also be designed in which students from various universities who have used these apps (e.g., 100 people) are randomly invited to independently evaluate the results of our requirements ranking to determine the reasonableness of the order. In addition, considering the time, effort, and financial constraints, we did not extensively compare our method with existing methods for prioritizing user requirements; thus, we will attempt to compare our approach with existing methods of prioritizing requirements in app reviews. This is fertile ground for future study. In addition, we will explore the interpretability of the model as well as compare the effects of using different XAI tools. Finally, we will explore whether app developers find the user requirements prioritization results obtained using XAI more understandable and agreeable compared to those obtained from previous studies utilizing other methods.

CRediT authorship contribution statement

Shizhen Bai: Conceptualization, Methodology, Writing – review & editing. **Songlin Shi:** Methodology, Software, Formal analysis, Data curation, Writing – original draft. **Chunjia Han:** Conceptualization, Methodology, Formal analysis, Writing – original draft, Writing – review & editing. **Mu Yang:** Software, Formal analysis, Data curation, Writing – original draft. **Brij B. Gupta:** Software, Formal analysis, Data curation. **Varsha Arya:** Conceptualization, Methodology.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgment

This work was supported by the Natural Science Foundation of Heilongjiang Province of China (Grant No. LH2021G014).

References

[1] Z.A. Gokgoz, M.B. Ataman, G.H. van Bruggen, There's an app for that! Understanding the drivers of mobile application downloads, J. Bus. Res. 123 (2021) 423–437, https://doi.org/10.1016/j.jbusres.2020.10.006.
[2] M. Iqbal, App revenue data. https://www.businessofapps.com/dataapp-revenues/, 2022 (accessed 12 March 2024).
[3] L. Ceci, Number of apps available in leading app stores Q3 2022. https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/, 2022 (accessed 15 March 2024).
[4] T. Avinadav, P. Levy, Value of information in a mobile app supply chain under hidden or known information superiority, Int. J. Prod. Econ. 248 (2022) 108467, https://doi.org/10.1016/j.ijpe.2022.108467.
[5] Y.J. Lee, H. Ghasemkhani, K. Xie, Y. Tan, Switching decision, timing, and app performance: an empirical analysis of mobile app developers' switching behavior between monetization strategies, J. Bus. Res. 127 (2021) 332–345, https://doi.org/10.1016/j.jbusres.2021.01.027.
[6] K. Biesialska, X. Franch, V. Muntés-Mulero, Big data analytics in Agile software development: a systematic mapping study, Inf. Softw. Technol. 132 (2021) 106448, https://doi.org/10.1016/j.infsof.2020.106448.
[7] J. Dąbrowski, E. Letier, A. Perini, A. Susi, Analysing app reviews for software engineering: a systematic literature review, Empir. Softw. Eng. 27 (2) (2022) 43, https://doi.org/10.1007/s10664-021-10065-7.
[8] Y. Liu, X. Tang, A. Bush, Intra-platform competition: the role of innovative and refinement evolution in app success, Inform. Manage. 58 (7) (2021) 103521, https://doi.org/10.1016/j.im.2021.103521.
[9] H. Gao, C. Guo, G. Bai, D. Huang, Z. He, Y. Wu, J. Xu, Sharing runtime permission issues for developers based on similar-app review mining, J. Syst. Softw. 184 (2022) 111118, https://doi.org/10.1016/j.jss.2021.111118.
[10] C. Mu, Application of user research in E-commerce app design, in: International Conference on Human-Computer Interaction, Cham, 2021, pp. 120–130, https://doi.org/10.1007/978-3-030-77750-0_8.
[11] J. Dąbrowski, E. Letier, A. Perini, A. Susi, Mining and searching app reviews for required eng.: evaluation and replication studies, Inf. Syst. 114 (2023) 102181, https://doi.org/10.1016/j.is.2023.102181.
[12] S. Shafiq, A. Mashkoor, C. Mayr-Dorn, A. Egyed, A literature review of using machine learning in software development life cycle stages, IEEE Access 9 (2021) 140896–140920, https://doi.org/10.1109/ACCESS.2021.3119746.
[13] F.M. Kifetew, A. Perini, A. Susi, A. Siena, D. Muñante, I. Morales-Ramirez, Automating user-feedback driven requirements prioritization, Inf. Softw. Technol. 138 (2021) 106635, https://doi.org/10.1016/j.infsof.2021.106635.
[14] L. Qi, Q. He, F. Chen, X. Zhang, W. Dou, Q. Ni, Data-driven web APIs recommendation for building web applications, IEEE Trans. Big Data 8 (3) (2020) 685–698, https://doi.org/10.1109/TBDATA.2020.2975587.
[15] L. Etaiwi, S. Hamel, Y.G. Guéhéneuc, W. Flageol, R. Morales, Order in chaos: prioritizing mobile app reviews using consensus algorithms, in: 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC), Madrid, Spain, 2020, pp. 912–920, https://doi.org/10.1109/COMPSAC48688.2020.0-151.
[16] W. Brunotte, L. Chazette, K. Korte, Can explanations support privacy awareness? A research roadmap, in: 2021 IEEE 29th International Requir. Eng. Conference Workshops (REW), Notre Dame, USA, 2021, pp. 176–180, https://doi.org/10.1109/REW53955.2021.00032.
[17] K.A. Karl, J.V. Peluchette, N. Aghakhani, Virtual work meetings during the COVID-19 pandemic: the good, bad, and ugly, Small Gr. Res. 53 (3) (2022) 343–365, https://doi.org/10.1177/10464964211015286.
[18] J. Hacker, J. Vom Brocke, J. Handali, M. Otto, J. Schneider, Virtually in this together–how web-conferencing systems enabled a new virtual togetherness

during the COVID-19 crisis, Eur. J. Inf. Syst. 29 (5) (2020) 563–584, https://doi.org/10.1080/0960085x.2020.1814680.
[19] S. Goldsworthy, M. Verkuyl, Facilitated virtual synchronous debriefing: a practical approach, Clin. Simul. Nurs. 59 (2021) 81–84, https://doi.org/10.1016/j.ecns.2021.06.002.
[20] J.B. Schmitt, J. Breuer, T. Wulf, From cognitive overload to digital detox: psychological implications of telework during the COVID-19 pandemic, Comput. Hum. Behav. 124 (2021) 106899, https://doi.org/10.1016/j.chb.2021.106899.
[21] K.M. Kuhn, The constant mirror: self-view and attitudes to virtual meetings, Comput. Hum. Behav. 128 (2022) 107110, https://doi.org/10.1016/j.chb.2021.107110.
[22] V. Grover, R. Kohli, Revealing your hand: caveats in implementing digital business strategy, MIS Quart. 37 (2) (2013) 655–662, https://doi.org/10.5555/2535658.2535679.
[23] P.R. Wijaya, P.V. Crisgar, M.D.F. Pakpahan, E.Y. Syamsuddin, M.O. Hasanuddin, Implementation of motor vehicle tracking software-as-a-service (SaaS) application based on progressive web app, in: 2021 International Symposium on Electronics and Smart Devices (ISESD), Bandung, Indonesia, 2021, pp. 1–6, https://doi.org/10.1109/ISESD53023.2021.9501600.
[24] H. Alamri, C. Maple, S. Mohamad, G. Epiphaniou, Do the right thing: a privacy policy adherence analysis of over two million apps in Apple IOS app store, Sensors 22 (2022) 8964, https://doi.org/10.3390/s22228964.
[25] S. Flensburg, S.S. Lai, Datafied mobile markets: measuring control over apps, data accesses, and third-party services, Mob. Media Commun. 10 (1) (2022) 136–155, https://doi.org/10.1177/20501579211039066.
[26] M.A. Hadi, F.H. Fard, Evaluating pre-trained models for user feedback analysis in software engineering: a study on classification of app-reviews, Empir. Softw. Eng. 28 (4) (2023) 88, https://doi.org/10.1007/s10664-023-10314-x.
[27] M. Sharma, D. Aggarwal, D. Pahuja, Categorization and classification of Uber reviews, in: Advances in Computing and Intelligent Systems: Proceedings of ICACM 2019, Beijing, China, 2020, pp. 347–355, https://doi.org/10.1007/978-981-15-0222-4_31.
[28] M. Bhatia, A. Kumar, R. Beniwal, An optimized classification of apps reviews for improving requirement engineering, Recent Adv. Comput. Sci. Commun. 14 (5) (2021) 1390–1399, https://doi.org/10.2174/2213275912666190716114919.
[29] S. Malgaonkar, S.A. Licorish, B.T.R. Savarimuthu, Automatically generating taxonomy for grouping app reviews-a study of three apps, Softw. Qual. J. 30 (2) (2022) 483–512, https://doi.org/10.1007/s11219-021-09570-1.
[30] J. Jiao, T.W. Simpson, Z. Siddique, Product family design and platform-based product development: a state-of-the-art review, J. Intell. Manuf. 18 (2007) 5–29, https://doi.org/10.1007/s10845-007-0003-2.
[31] J. Dąbrowski, E. Letier, A. Perini, A. Susi, Finding and analyzing app reviews related to specific features: a research preview, in: International Working Conference on Requir. Eng.: Foundation for Software Quality, Cham, 2019, pp. 183–189, https://doi.org/10.1007/978-3-030-15538-4_14.
[32] S. Malgaonkar, S.A. Licorish, B.T.R. Savarimuthu, Prioritizing user concerns in app reviews–A study of requests for new features, enhancements and bug fixes, Inf. Softw. Technol. 144 (2022) 106798, https://doi.org/10.1016/j.infsof.2021.106798.
[33] R. Chen, Q. Wang, W. Xu, Mining user requirements to facilitate mobile app quality upgrades with big data, Electron. Commer. Res. Appl. 38 (2019) 100889, https://doi.org/10.1016/j.elerap.2019.100889.
[34] C. Yang, L. Wu, C. Yu, Y. Zhou, A phrase-level user requests mining approach in mobile application reviews: concept, framework, and operation, Information 12 (5) (2021) 177, https://doi.org/10.3390/info12050177.
[35] P. Carmona, A. Dwekat, Z. Mardawi, No more black boxes! Explaining the predictions of a machine learning XGBoost classifier algorithm in business failure, Res. Int. Bus. Finance. 61 (2022) 101649, https://doi.org/10.1016/j.ribaf.2022.101649.
[36] C. Yáñez-Márquez, Toward the bleaching of the black boxes: minimalist machine learning, IT Prof. 22 (4) (2020) 51–56, https://doi.org/10.1109/MITP.2020.2994188.
[37] M.R. Islam, M.U. Ahmed, S. Barua, S. Begum, A systematic review of explainable artificial intelligence in terms of different application domains and tasks, Appl. Sci. 12 (3) (2022) 1353, https://doi.org/10.3390/app12031353.
[38] A. Soni, A. Saxena, P. Bajaj, A methodological approach for mining the user requirements using apriori algorithm, J. Cases Inf. Technol. 22 (4) (2020) 1–30, https://doi.org/10.4018/JCIT.2020100101.
[39] N. Burkart, M.F. Huber, A survey on the explainability of supervised machine learning, J. Artif. Intell. Res. 70 (2021) 245–317, https://doi.org/10.1613/jair.1.12228.
[40] J. Zhou, A.H. Gandomi, F. Chen, A. Holzinger, Evaluating the quality of machine learning explanations: a survey on methods and metrics, Electronics (Basel) 10 (5) (2021) 593, https://doi.org/10.3390/electronics10050593.
[41] L. Chazette, K. Schneider, Explainability as a non-functional requirement: challenges and recommendations, Requir. Eng. 25 (4) (2020) 493–514, https://doi.org/10.1007/s00766-020-00333-1.
[42] C. Pornprasit, C. Tantithamthavorn, J. Jiarpakdee, M. Fu, P. Thongtanunam, PyExplainer: explaining the predictions of just-in-time defect models, in: 2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE), Melbourne, Australia, 2021, pp. 1–2, https://doi.org/10.1109/ASE51524.2021.9678763.
[43] B. Jamasb, R. Akbari, S.R. Khayami, On the applicability of explainable artificial intelligence for software requirement analysis, arxiv preprint arxiv:2302.05266 (2023). https://doi.org/10.48550/arXiv.2302.05266.
[44] F. Dalpiaz, D. Dell'Anna, F.B. Aydemir, S. Çevikol, Requirements classification with interpretable machine learning and dependency parsing, in: 2019 IEEE 27th International Requir. Eng. Conference (RE), Jeju, Korea (South), 2019, pp. 142–152, https://doi.org/10.1109/RE.2019.00025.
[45] D. Clements, E. Giannis, F. Crowe, M. Balapitiya, J. Marshall, P. Papadopoulos, T. Kanij, An innovative approach to develop persona from application reviews, in: International Conference on Evaluation of Novel Approaches to Software Engineering 2023, Prague, Czechia, 2023, pp. 701–708, https://doi.org/10.5220/0011996000003464.
[46] D. Mcaleese, M. Linardakis, A. Papadaki, Quality and presence of behaviour change techniques in mobile apps for the Mediterranean diet: a content analysis of Android google play and Apple app store apps, Nutrients 14 (6) (2022) 1290, https://doi.org/10.3390/nu14061290.
[47] J. Aritonang, R. Rokhim, Big data analysis of paid and free applications in google playstore and apple app store to know application characteristics and monetization opportunities for new startup in Indonesia, in: Proceedings of the International Conference on Business and Management Research (ICBMR 2020), 2020, pp. 205–210, https://doi.org/10.2991/aebmr.k.201222.030. Online.
[48] S. Sadiq, M. Umer, S. Ullah, S. Mirjalili, V. Rupapara, M. Nappi, Discrepancy detection between actual user reviews and numeric ratings of Google App store using deep learning, Expert Syst. Appl. 181 (2021) 115111, https://doi.org/10.1016/j.eswa.2021.115111.
[49] A. Juneja, N.N. Das, Big data quality framework: pre-processing data in weather monitoring application, in: 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India, 2019, pp. 559–563, https://doi.org/10.1109/COMITCon.2019.8862267.
[50] M.S. Hossain, M.K. Uddin, M.K. Hossain, M.F. Rahman, User sentiment analysis and review rating prediction for the blended learning platform app, in: G Trajkovski, M Demeter, H Hayes (Eds.), Applying Data Science and Learning Analytics Throughout a Learner'S Lifespan, IGI Global, Hershey, Pennsylvania, 2022, pp. 113–132, https://doi.org/10.4018/978-1-7998-9644-9.ch006.
[51] L. Zheng, S. Lin, Motivation, appearance focus, and exclusion on gay dating app "Blued" in China: content and network analysis of textual self-presentation, J. Sex. Res. (2023) 1–13, https://doi.org/10.1080/00224499.2023.2218345.
[52] S.F. Verkijika, B.N. Neneh, Standing up for or against: a text-mining study on the recommendation of mobile payment apps, J. Retail. Consum. Serv. 63 (2021) 102743, https://doi.org/10.1016/j.jretconser.2021.102743.
[53] G. Allon, G. Askalidis, R. Berry, N. Immorlica, K. Moon, A. Singh, When to be agile: ratings and version updates in mobile apps, Manage. Sci. 68 (6) (2022) 4261–4278, https://doi.org/10.1287/mnsc.2021.4112.
[54] S. Dhar, I. Bose, Walking on air or hopping mad? Understanding the impact of emotions, sentiments and reactions on ratings in online customer reviews of mobile apps, Decis. Support Syst. 162 (2022) 113769, https://doi.org/10.1016/j.dss.2022.113769.
[55] I. Triantafyllou, I.C. Drivas, G. Giannakopoulos, How to utilize my app reviews? A novel topics extraction machine learning schema for strategic business purposes, Entropy 22 (11) (2020) 1310, https://doi.org/10.3390/e22111310.
[56] J. Wanner, L.V. Herm, C. Aniesch, How much is the black box? The value of explainability in machine learning models, in: ECIS 2020 Research-in-Progress Papers, Marrakech, Morocco, 2020.
[57] C. Duffy, Apple co-founder Steve Wozniak says Apple Card discriminated against his wife. https://edition.cnn.com/2019/11/10/business/goldman-sachs-apple-card-discrimination/index.html, 2019 (accessed 13 March 2024).
[58] G. Sevilla, Zoom vs. Microsoft Teams vs. Google Meet: which top videoconferencing app is best. https://www.pcmag.com/news/zoom-vs-microsoft-teams-vs-google-meet-a-videoconferencing-face-off, 2020 (accessed 13 March 2024).
[59] N.H. Gauthier, M.I. Husain, Dynamic security analysis of zoom, Google meet and Microsoft teams, in: Silicon Valley Cybersecurity Conference: First Conference, SVCC 2020, San Jose, CA, USA, 2021, pp. 3–24, https://doi.org/10.1007/978-3-030-72725-3_1.
[60] K.A. Siddiqui, S. Ahmad, Comparative study of alternative teaching and learning tools: google meet, microsoft teams, and zoom during COVID-19. Teaching in the Pandemic Era in Saudi Arabia, Brill, Leiden, 2022, pp. 120–129, https://doi.org/10.1163/9789004521674_008.
[61] J. Yingkongdee, K. Jantarakolica, S. Sukhaparamate, T. Jantarakolica, C. Sajjanit, P. Teekasap, Factors affecting the technology acceptance of E-learning through Google Meet, MS Team and Zoom, Rev. Int. Geograph. Educ. Online 11 (12) (2021) 1–6.
[62] D. Geradin, D. Katsifis, The antitrust case against the Apple app store, J. Compet. Law Econ. 17 (3) (2021) 503–585, https://doi.org/10.1093/joclec/nhab003.
[63] M. Li, Y. Han, K.Y. Goh, H. Cavusoglu, Mobile app portfolio management and developers' performance: an empirical study of the apple app store, Inform. Manag. 59 (8) (2022) 103716, https://doi.org/10.1016/j.im.2022.103716.
[64] P. Wen, M. Chen, A new analysis method for user reviews of mobile fitness apps, in: human-Computer interaction. human values and quality of life: thematic area, HCI 2020, in: Held as Part of the 22nd International Conference, HCII 2020, Copenhagen, Denmark, 2020, pp. 188–199, https://doi.org/10.1007/978-3-030-49065-2_14.
[65] S. Venkatakrishnan, A. Kaushik, J.K. Verma, Sentiment analysis on google play store data using deep learning, in: P. Johri, J. Verma, S. Paul (Eds.), Applications of Machine Learning, Springer, Singapore, 2020, pp. 15–30, https://doi.org/10.1007/978-981-15-3357-0_2.
[66] U. Khaira, R. Johanda, P.E.P. Utomo, T. Suratno, Sentiment analysis of cyberbullying on twitter using SentiStrength, Ind. J. Art. Intell. Data Mining. 3 (1) (2020) 21–27, https://doi.org/10.24014/ijaidm.v3i1.9145.

[67] B. Siregar, M. Misyuari, E. Nababan, Person's multiple intelligence classification based on tweet post using sentiStrength and processed on the Apache spark framework, J. Phys. Conf. Ser. 1882 (1) (2021) 012125, https://doi.org/10.1088/1742-6596/1882/1/012125.

[68] M. Thelwall, The heart and soul of the web? in: J. Holyst (Ed.), Sentiment Strength Detection in the Social Web With SentiStrength Springer, Cham, 2017, pp. 119–134, https://doi.org/10.1007/978-3-319-43639-5_7. Cyberemotions.

[69] M.O. Diaz Jr, A domain-specific evaluation of the performance of selected web-based sentiment analysis platforms, Int. J. Softw. Eng. Comput. Syst. 9 (1) (2023) 1–9, https://doi.org/10.15282/ijsecs.9.1.2023.1.0105.

[70] S. Gouthami, N.P. Hegde, Automatic sentiment analysis scalability prediction for information extraction using sentistrength algorithm, in: Proceedings of Third International Conference on Advances in Computer Engineering and Communication Systems, Singapore, 2023, pp. 21–30, https://doi.org/10.1007/978-981-19-9228-5_3.

[71] R.W. Hardian, P.E. Prasetyo, U. Khaira, T. Suratno, Analisis sentiment kuliah daring di media sosial twitter selama pandemi Covid-19 menggunakan algoritma sentistrength: online lecture sentiment analisys on twitter social media during the Covid-19 pandamic using sentistrength algorithm, MALCOM 1 (2) (2021) 138–143, https://doi.org/10.57152/malcom.v1i2.15.

[72] J. Hartmann, M. Heitmann, C. Siebert, C. Schamp, More than a feeling: accuracy and application of sentiment analysis, Int. J. Res. Mark. 40 (1) (2023) 75–87, https://doi.org/10.1016/j.ijresmar.2022.05.005.

[73] J. Chung, J. Lee, J. Yoon, Understanding music streaming services via text mining of online customer reviews, Electron. Commer. Res. Appl. 53 (2022) 101145, https://doi.org/10.1016/j.elerap.2022.101145.

[74] M.E. Roberts, B.M. Stewart, D. Tingley, C. Lucas, J. Leder-Luis, S.K. Gadarian, B. Albertson, D.G. Rand, Structural topic models for open-ended survey responses, Am. J. Pol. Sci. 58 (4) (2014) 1064–1082, https://doi.org/10.1111/ajps.12103.

[75] M. Anandarajan, C. Hill, T. Nolan, M. Anandarajan, C. Hill, T. Nolan, Probabilistic topic models, in: F. Dai, R. Maitra (Eds.), Practical Text Analytics: Maximizing the Value of Text Data, Springer, Cham, 2019, pp. 117–130, https://doi.org/10.1007/978-3-319-95663-3.

[76] M.E. Roberts, B.M. Stewart, D. Tingley, Stm: an R package for structural topic models, J. Stat. Softw. 91 (2019) 1–40, https://doi.org/10.18637/jss.v091.i02.

[77] J.H. Kim, J. Jang, Y. Kim, D. Nan, A structural topic model for exploring user satisfaction with mobile payments, Comput. Mate. Cont. 73 (2) (2022) 3815–3826, https://doi.org/10.32604/cmc.2022.029507.

[78] D. Vaughan, M. Dancho, Furrr: apply mapping functions in parallel using futures, R package version 0.3.1. https://github.com/DavisVaughan/furrr, 2022 (accessed 13 March 2024).

[79] S.J. Weston, I. Shryock, R. Light, P.A. Fisher, Selecting the number and labels of topics in topic modeling: a tutorial, Adv. Methods Pract. Psychol. Sci. 6 (2) (2023) 25152459231160105, https://doi.org/10.1177/25152459231160105.

[80] J. Liu, C. Hong, B. Yook, C.E.O. as, chief crisis officer" under COVID-19: a content analysis of CEO open letters using structural topic modeling, Int. J. Strategic Commun. 16 (3) (2022) 444–468, https://doi.org/10.1080/1553118X.2022.2045297.

[81] P. Rani, J. Shokeen, A survey of tools for social network analysis, Int. J. Web Eng. Technol. 16 (3) (2021) 189–216, https://doi.org/10.1504/IJWET.2021.119879.

[82] D. Knoke, S. Yang, Social Network Analysis, SAGE publications, Thousand Oaks, 2019.

[83] N. Chouchani, M. Abed, Online social network analysis: detection of communities of interest, J. Intell. Inform. Syst. 54 (1) (2020) 5–21, https://doi.org/10.1007/s10844-018-0522-7.

[84] N.A. Prabowo, B. Pujiarto, F.S. Wijaya, L. Gita, D. Alfandy, Social network analysis for user interaction analysis on social media regarding e-commerce business, Int. J. Inform. Inform. Syst. 4 (2) (2021) 95–102, https://doi.org/10.47738/ijiis.v4i2.106.

[85] J.L. Chang, H. Li, J.W. Bi, Personalized travel recommendation: a hybrid method with collaborative filtering and social network analysis, Curr. Issues Tour. 25 (14) (2022) 2338–2356, https://doi.org/10.1080/13683500.2021.2014792.

[86] N. Akhtar, M.V. Ahamad, Graph tools for social network analysis, in: E. Marsh (Ed.), Research Anthology on Digital Transformation, Organizational Change, and the Impact of Remote Work, IGI Global, 2021, pp. 485–500, https://doi.org/10.4018/978-1-7998-7297-9.ch025.

[87] M. Bastian, S. Heymann, M. Jacomy, Gephi: an open source software for exploring and manipulating networks, in: Proceedings of the International AAAI Conference on Web and Social Media, Chicago, IL, USA, 2009, pp. 361–362, https://doi.org/10.1609/icwsm.v3i1.13937.

[88] M. Jacomy, T. Venturini, S. Heymann, M. Bastian, ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software, PLoS ONE 9 (6) (2014) e98679, https://doi.org/10.1371/journal.pone.0098679.

[89] A. Bruns, H. Snee, How to Visually Analyse Networks Using Gephi, SAGE Publications, Limited, London, 2022, https://doi.org/10.4135/9781529609752.

[90] H. Jung, B.G. Lee, Research trends in text mining: semantic network and main path analysis of selected journals, Exp. Syst. Appl. 162 (2020) 113851, https://doi.org/10.1016/j.eswa.2020.113851.

[91] N. Samatan, A. Fatoni, S. Murtiasih, Disaster communication patterns and behaviors on social media: a study social network# BANJIR2020 on Twitter, Hum. Soc. Sci. Rev. 8 (4) (2020) 27–36, https://doi.org/10.18510/hssr.2020.844.

[92] M. Kuhn, Building predictive models in R using the caret package, J. Stat. Softw. 28 (2008) 1–26, https://doi.org/10.18637/jss.v028.i05.

[93] C. Rudin, Why black box machine learning should be avoided for high-stakes decisions, in brief, Nat. Rev. Methods Prim. 2 (2022) 81, https://doi.org/10.1038/s43586-022-00172-0.

[94] N.T. Le, J.W. Wang, D.H. Le, C.C. Wang, T.N. Nguyen, Fingerprint enhancement based on tensor of wavelet subbands for classification, IEEE Access 8 (2020) 6602–6615, https://doi.org/10.1109/ACCESS.2020.2964035.

[95] P. Subramani, K. Srinivas, R.B. Kavitha, R. Sujatha, B.D. Parameshachari, Prediction of muscular paralysis disease based on hybrid feature extraction with machine learning technique for COVID-19 and post-COVID-19 patients, Pers. Ubiquit. Comput. 27 (3) (2023) 831–844, https://doi.org/10.1007/s00779-021-01531-6.

[96] P. Biecek, DALEX: explainers for complex predictive models in R, J. Mach. Learn. Res. 19 (84) (2018) 3245–3249, https://doi.org/10.48550/arXiv.1806.08915.

[97] F. Günther, S. Fritsch, Neuralnet: training of neural networks, R J 2 (2010) 30–38, https://doi.org/10.32614/RJ-2010-006.

[98] P. Biecek, T. Burzykowski, Explanatory Model analysis: Explore, Explain and Examine Predictive Models, Chapman and Hall/CRC, New York, 2021, https://doi.org/10.1080/00401706.2022.2091871.

[99] P. Hall, N. Gill, An Introduction to Machine Learning Interpretability, O'Reilly Media, Sebastopol, 2019.

[100] L.H. Le, Q.A. Ha, Effects of negative reviews and managerial responses on consumer attitude and subsequent purchase behavior: an experimental design, Comput. Hum. Behav. 124 (2021) 106912, https://doi.org/10.1016/j.chb.2021.106912.

[101] J. Dąbrowski, E. Letier, A. Perini, A. Susi, Mining user opinions to support requirement engineering: an empirical study, in: International Conference on Advanced Information Systems Engineering, Grenoble, France, 2020, pp. 401–416, https://doi.org/10.1007/978-3-030-49435-3_25.

[102] F. Dalpiaz, M. Parente, RE-SWOT: from user feedback to requirements via competitor analysis, in: Requirements Engineering: Foundation for Software Quality: 25th International Working Conference, Essen, Germany, 2019, pp. 55–70, https://doi.org/10.1007/978-3-030-15538-4_4.

[103] C. Tao, H. Guo, Z. Huang, Identifying security issues for mobile applications based on user review summarization, Inf. Softw. Technol. 122 (2020) 106290, https://doi.org/10.1016/j.infsof.2020.106290.

[104] T. Zhang, Z. He, A. Mukherjee, Monitoring negative sentiment scores and time between customer complaints via one-sided distribution-free EWMA schemes, Comput. Ind. Eng. 180 (2023) 109247, https://doi.org/10.1016/j.cie.2023.109247.

[105] N. Korfiatis, P. Stamolampros, P. Kourouthanassis, V. Sagiadinos, Measuring service quality from unstructured data: a topic modeling application on airline passengers' online reviews, Expert Syst. Appl. 116 (2019) 472–486, https://doi.org/10.1016/j.eswa.2018.09.037.

[106] N. Singer, N. Perlroth, Zoom's Security Woes Were No Secret to Business Partners Like Dropbox, International New York Times, 2020. https://easycloudsolutions.com/2020/05/06/zooms-security-woes-were-no-secret-to-business-partners-like-dropbox/. May 6,accessed 13 March 2024.

[107] J. Olson, K. Walters, F. Appunn, L. Grinnell, C. Mcallister, The value of webcams for virtual teams, Int. J. Manag. Inform. Syst. 16 (2) (2012) 161–172, https://doi.org/10.19030/ijmis.v16i2.6915.

[108] D. Kagan, G.F. Alpert, M. Fire, Zooming into video conferencing privacy and security threats, arXiv preprint arXiv:2007.01059 (2020). https://doi.org/10.48550/arXiv.2007.01059.

[109] E.G.S. Nascimento, A.N. Furtado, R. Badaró, L. Knop, The New Technologies in the Pandemic Era, J. Bio. Technol. Health. 3 (2) (2020) 134–164, https://doi.org/10.34178/jbth.v3i2.122.

[110] I.K.A. Aryadinata, D. Pangesti, G.B. Anugerah, I.E. Aditya, Y. Ruldeviyani, Sentiment analysis of 5 G network implementation in Indonesia using twitter data, in: 2021 6th International Workshop on Big Data and Information Security (IWBIS), Depok, Indonesia, 2021, pp. 23–28, https://doi.org/10.1109/IWBIS53353.2021.9631863.

[111] Z.P. Neal, How small is it? Comparing indices of small worldliness, Netw. Sci. 5 (1) (2017) 30–44, https://doi.org/10.1017/nws.2017.5.

[112] S. Das, I. Tsapakis, Interpretable machine learning approach in estimating traffic volume on low-volume roadways, Int. J. Transp. Sci. Technol. 9 (1) (2020) 76–88, https://doi.org/10.1016/j.ijtst.2019.09.004.

[113] B.M. Greenwell, pdp: an R package for constructing partial dependence plots, R J 9 (1) (2017) 421, https://doi.org/10.32614/RJ-2017-016.

[114] J. Moosbauer, J. Herbinger, G. Casalicchio, M. Lindauer, B. Bischl, Explaining hyperparameter optimization via partial dependence plots, Adv. Neural Inf. Process. Syst. 34 (2021) 2280–2291, https://doi.org/10.48550/arXiv.2111.04820.

[115] C. Zhang, M.C. Yu, S. Marin, Exploring public sentiment on enforced remote work during COVID-19, J. Appl. Psychol. 106 (6) (2021) 797, https://doi.org/10.1037/apl0000933.

[116] D. Chakraborty, Customer satisfaction towards food service apps in Indian metro cities, FIIB, Bus. Rev. 8 (3) (2019) 245–255, https://doi.org/10.1177/2319714519844651.

[117] A.P. Correia, C. Liu, F. Xu, Evaluating videoconferencing systems for the quality of the educational experience, Dis. Educ. 41 (4) (2020) 429–452, https://doi.org/10.1080/01587919.2020.1821607.

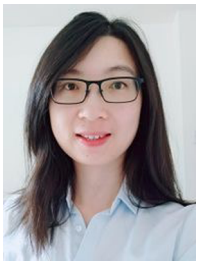
[118] D. Harborth, S. Pape, Investigating privacy concerns related to mobile augmented reality Apps–A vignette based online experiment, Comput. Hum. Behav. 122 (2021) 106833, https://doi.org/10.1016/j.chb.2021.106833.

[119] W. Martin, F. Sarro, Y. Jia, Y. Zhang, M. Harman, A survey of app store analysis for software engineering, IEEE Trans. Softw. Eng. 43 (9) (2016) 817–847, https://doi.org/10.1109/tse.2016.2630689.
[120] J.J. Aman, J. Smith-Colin, W. Zhang, Listen to E-scooter riders: mining rider satisfaction factors from app store reviews, Transp. Res. D Transp. Environ. 95 (2021) 102856, https://doi.org/10.1016/j.trd.2021.102856.
[121] T.L. James, Z. Qiao, W. Shen, G.A. Wang, W. Fan, Competing for temporary advantage in a hypercompetitive mobile app market, Mis Quar. 47 (3) (2023) 1177–1212, https://doi.org/10.25300/MISQ/2022/15079.
[122] F. Ebrahimi, A. Mahmoud, Unsupervised summarization of privacy concerns in mobile application reviews, in: Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering, Michigan, 2022, pp. 1–12, https://doi.org/10.1145/3551349.3561155.
[123] M.S. Ullal, C. Spulbar, I.T. Hawaldar, V. Popescu, R. Birau, The impact of online reviews on e-commerce sales in India: a case study, Econ. Res. 34 (1) (2021) 2408–2422, https://doi.org/10.1080/1331677X.2020.1865179.
[124] H. Liu, Y. Wang, Y. Liu, S. Gao, Supporting features updating of apps by analyzing similar products in App stores, Inform. Sci. 580 (2021) 129–151, https://doi.org/10.1016/j.ins.2021.08.050.
[125] S. Lim, A. Henriksson, J. Zdravkovic, Data-driven requirements elicitation: a systematic literature review, SN Comput. Sci. 2 (1) (2021) 1–35, https://doi.org/10.1007/s42979-020-00416-4.
[126] C. Wohlin, P. Runeson, M. Höst, M.C. Ohlsson, B. Regnell, A. Wesslén, Experimentation in Software Engineering, Springer Science & Business Media, New York, 2012. https://link.springer.com/book/10.1007/978-3-642-29044-2.
[127] S.C. Lee, Y. Jang, C.H. Park, Y.S. Seo, Feature Analysis for Detecting Mobile Application Review Generated by AI-Based Language Model, J. Inform. Proc. Syst. 18 (5) (2022) 650, https://doi.org/10.3745/JIPS.02.0182.

Shizhen Bai, born in 1962, is a second-class professor at Harbin University of Commerce and a doctoral supervisor. His main research areas are digital supply chain and big data business analysis. He has presided over more than 10 national projects, such as the Major Project of Late Funding for Philosophy and Social Science of the Ministry of Education of China, the National Natural Science Foundation of China, and the National Social Science Foundation of China, etc. He has published more than 100 papers in important academic journals, such as IEEE Transactions on Engineering Management.

Songlin Shi is a postgraduate student of Management Science and Engineering at Harbin University of Commerce. Mr. Songlin Shi's research interests are text analytics, machine learning, explainable artificial intelligence and sustainability.

Dr **Chunjia Han** is an Associate Professor in Business Innovation at Birkbeck, University of London, UK. By focusing on the application, development, impact of digital technologies (e.g., big data analytics, AI, and social media) in influencing practices in business and economy, his research contributes to our knowledge on innovation management, tourism management, digital marketing, and digital economy.



Dr. **Mu Yang** is an Associate Professor in Business Analytics at Birkbeck, University of London. She started as a computer scientist with expertise in data anonymisation, user privacy, artificial intelligence and blockchain. With extensive research experience in data analytics, her research interests expand to Tourism Management, Marketing Analytics, Big Data Analytics, Open Innovation, Digital Economy, Privacy and Social Dynamics.

Brij B. Gupta is working as Director of International Center for AI and Cyber Security Research and Innovations, and Distinguished Professor with the Department of Computer Science and Information Engineering (CSIE), Asia University, Taiwan. In more than 17 years of his professional experience, he published over 500 papers in journals/conferences including 35 books and 11 Patents with over 24,000 citations. He has received numerous national and international awards including Canadian Commonwealth Scholarship (2009), Faculty Research Fellowship Award (2017), MeitY, GoI, IEEE GCCE outstanding and WIE paper awards and Best Faculty Award (2018 & 2019), NIT KKR, respectively. Prof. Gupta was selected for 2022 Clarivate Web of Science Highly Cited Researchers in Computer Science. He was also selected in the 2022, 2021 and 2020 Stanford University's ranking of the world's top 2 % scientists. He is also a visiting/adjunct professor with several universities worldwide. He is also an IEEE Senior Member (2017) and also selected as 2021 Distinguished Lecturer in IEEE CTSoc. Dr Gupta is also serving as Member-in-Large, Board of Governors, IEEE Consumer Technology Society (2022–2024). Prof Gupta is also leading IJSWIS, IJSSCI, STE and IJCAC as Editor-in-Chief. Moreover, he is also serving as lead-editor of a Book Series with CRC and IET press. He also served as TPC members in more than 150 international conferences also serving as Associate/Guest Editor of various journals and transactions. His research interests include information security, Cyber physical systems, cloud computing, blockchain technologies, intrusion detection, AI, social media and networking.

Varsha Arya did Master's degree from Rajasthan University, India in 2015 and has been working as a researcher for the last 7 years. She published more than 25 papers in top journals and conferences. Her research interests include business administration, technology management, Cyber physical systems, cloud computing, healthcare and networking. Currently, she is doing research at Asia University, Taiwan, China