



INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv



A Review on Classifying and Prioritizing User Review-Based Software Requirements

Amran Salleh^a, Mar Yah Said^{a,*}, Mohd Hafeez Osman^a, Sa'adah Hassan^a

^a Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang, Selangor, Malaysia

Corresponding author: *maryah@upm.edu.my

Abstract—User reviews are a valuable source of feedback for software developers, as they contain user requirements, opinions, and expectations regarding app usage, including dislikes, feature requests, and reporting bugs. However, extracting and analyzing user requirements from user reviews is ineffective due to the large volume, unstructured nature, and varying quality of the reviews. Therefore, further research is not just necessary but crucial to effectively explore methods to gather informative and meaningful user feedback. This study aims to investigate, analyze, and summarize the methods of requirement classification and prioritization techniques derived from user reviews. This review revealed that leveraging opinion mining, sentiment analysis, natural language processing, or any stacking technique can significantly enhance the extraction and classification processes. Additionally, an updated matrix taxonomy has been developed based on a combination of definitions from various studies to classify user reviews into four main categories: information seeking, feature request, problem discovery, and information giving. Furthermore, we identified Naive Bayes, SVM, and Neural Networks algorithms as dependable and suitable for requirement classification and prioritization tasks. The study also introduced a new 4-tuple pattern for efficient requirement prioritization, which included elicitation technique, requirement classification, additional factors, and higher range priority value. This study highlights the need for better tools to handle complex user reviews. Investigating the potential of emerging machine learning models and algorithms to improve classification and prioritization accuracy is crucial. Additionally, further research should explore automated classification to enhance efficiency.

Keywords—User reviews; requirements prioritization; requirements classification; mobile apps; user requirements.

Manuscript received 4 Jun. 2024; revised 9 Aug. 2024; accepted 12 Oct. 2024. Date of publication 30 Nov. 2024.
International Journal on Informatics Visualization is licensed under a Creative Commons Attribution-Share Alike 4.0 International License.



I. INTRODUCTION

In today's swiftly changing technology landscape, software evolution and maintenance have emerged as integral components of software engineering activities [1] to ensure the software remains functional and cost-effective [2]. Software evolution involves periodic updates to enhance features or eliminate unused functions. Meanwhile, software maintenance involves transforming, modifying, and updating software to meet customer requirements.

Furthermore, it is imperative to acknowledge that user feedback, mainly through mobile app reviews, is a rich source of information [3]–[5], as they contain user requirements, opinions, and expectations regarding app usage, including dislikes, requests for new features, and bug reports [6], [7]. Therefore, it plays a crucial role in shaping the future of software evolution and maintenance.

However, extracting and analyzing user requirements from user reviews is challenging due to the large volume,

unstructured nature, and varying quality of the reviews. According to a study conducted by [8], user reviews generate large volumes [9] of data with noisy characteristics, as also stated by [10]. This data will grow exponentially and rapidly, leading to difficulties in classifying and prioritizing the software requirements derived from the reviews. Furthermore, Fereidouni et al. [11] state that identifying critical user feedback from many reviews remains an ongoing challenge. Therefore, further research is necessary to effectively explore methods to gather informative and meaningful user feedback [12]. Moreover, one of the problems faced is the unstructured nature of user reviews [13], [14]. The lack of structure in these reviews hinders the recognition of precise software requirements, as they often contain unclear and incomplete requirements. Another problem is the noise from the internet slang, shortcut grammar, and bad formatting [15]. Additionally, del Sagrado et al. [16] also stated the difficulty in selecting appropriate requirements for inclusion in the following app release. These

factors often raise doubts about the relevance and validity of the information extracted from the reviews. Another problem and challenge in user reviews is the varying quality of user reviews [17], [18], which can make the identification of helpful user feedback a very challenging task. The lack of uniform terminology can obstruct the development of practical software requirements. Therefore, this study addresses the research problem of how to classify and prioritize software requirements from user reviews. Table I below shows our study's comparison aspects that are different from the existing literature.

Hence, this study aims to investigate, analyze, and summarize the methods of requirement classification and prioritization techniques derived from user reviews. To achieve this aim, we have crafted five Research Questions (RQs) to delve into the critical analysis of recently conducted research on classification and prioritization tasks based on eliciting user requirements from user reviews:

- **RQ1:** What are the key challenges of incorporating user reviews for requirement classification and prioritization in software development?
- **RQ2:** What techniques are commonly used in research studies to enhance the understanding of user needs with short text inputs, aiming to improve the technique of user requirement classification and prioritization?
- **RQ3:** What are the predominant classifications of user reviews recommended in the preceding study?
- **RQ4:** How did the previous scholars classify user requirements?
- **RQ5:** What is a proper machine learning technique to be leveraged to enhance the accuracy of requirement classification and improve the requirement prioritization process in software development projects?

This study's method followed Kitchenham et al.'s guidelines [19]. By providing a comprehensive review, the study contributes to the field of software engineering and to the research and practice of user review processing in software requirements engineering. This study also proposes a matrix user review of the taxonomy based on the definition and topic. Besides, this study proposes a novel pattern involving a 4-tuple structure that consists of an elicitation technique, requirement classification, additional factors, and priority value for each requirement.

The structure of this paper is divided into the following sections. Section II introduces the materials and method. Section III reports the results and discussion. Furthermore, Section IV will present conclusions and potential future work.

II. MATERIALS AND METHODS

For our review, we followed the guidelines of Kitchenham et al. [19]. The selection of Kitchenham et al.'s method for this review is primarily due to its alignment with the specific needs of software engineering research, as discussed in the SEGRESS guidelines [19], while PRISMA was initially developed for the medical and healthcare fields. Besides that, Kitchenham's approach offers tailored guidelines that better accommodate the diverse nature of software engineering studies, including mapping studies. This makes it more suitable for ensuring comprehensive and contextually relevant systematic literature reviews in this field.

A. Eligibility Criteria

Initially, we selected the studies based on their titles. After removing duplicate papers, we applied the following inclusion criteria (ICs) and exclusion criteria (ECs):

1) Inclusion criteria:

- IC1 Articles published between 2015 and 2023.
- IC2 Articles are written in English.
- IC3 Articles published in computer science.
- IC4 Type of articles in the conference or journal format.
- IC4 Articles related to user review-based requirement engineering.
- IC5 Articles available in digital format

2) Exclusion criteria:

- EC1 The paper is not peer-reviewed, book, review, or in press.
- EC2 Articles that are not relevant to user review-based requirement engineering.
- EC3 Articles that are duplicates or have substantial overlaps with other articles.
- The EC4 website has no author and cannot be verified.

B. Information Sources

This study's data sources are gathered from four online digital libraries: ScienceDirect, IEEE Xplore, Scopus, and Web of Science.

C. Search Strategy

The search strategy was meticulously designed following the SEGRESS guidelines [19] to thoroughly review the literature on the classification and prioritization of user review-based software requirements. The strategy aimed to ensure an exhaustive collection of relevant studies, minimizing the risk of missing pertinent literature. The process was structured according to best practices outlined by [19], emphasizing transparency and replicability.

The search was executed across four major academic databases: ScienceDirect, IEEE Xplore, Scopus, and Web of Science. These databases were selected for their extensive coverage of software engineering and related disciplines, ensuring access to a wide range of peer-reviewed articles, conference papers, and other relevant publications.

A comprehensive search string was constructed to identify studies on the elicitation, prioritization, and classification of software requirements derived from user reviews. The search string was carefully crafted using Boolean operators "AND" and "OR" to combine key terms and their synonyms, allowing for flexibility and inclusiveness in the search process. For instance, a representative search string used in the databases was:

TITLE-ABS-KEY(requirement AND (elicitation OR gathering) AND (classification OR classify OR grouping) AND (prioritization OR ranking OR weighting OR sorting))

This search string reflects the topic's multifaceted nature, ensuring that studies covering various requirement elicitation and prioritization aspects were captured. The search terms were derived from an extensive preliminary literature review and iteratively refined to optimize search results. Additionally, alternative search terms were considered and

included in the search strategy to capture studies that may use different terminology. For instance, synonyms for "prioritization," such as "ranking" or "weighting," were included to ensure comprehensive coverage. This approach aligns with the guidance provided by [19] on enhancing the breadth and depth of search processes in systematic reviews.

The search process guidelines suggest presenting the full electronic search strategy for at least one database, as shown

in Table 5 [19]. This practice was followed to allow for the replication of the search process and to enhance the transparency of the review. Regarding search limitations, the search was restricted to English-language publications due to the reviewers' linguistic constraints and the predominance of English in academic publications in software engineering. No restrictions were placed on publication dates to capture foundational and recent studies.

TABLE I
OUR STUDY VS. EXISTING LITERATURE

Aspect	Our Literature Study	Existing Studies
Challenges in Incorporating User Reviews	Identifies specific challenges in using user reviews for requirement classification and prioritization.	[20]: Discuss challenges in opinion mining from mobile app store reviews but not specifically for requirement classification and prioritization. [21]: Highlight challenges developers face with app store feedback but focus more on overall software engineering practices than just requirements.
Techniques to Enhance Understanding of User Needs	Investigates techniques that improve understanding of user needs through short text inputs.	[22]: Examine various automated tools and technologies for requirements elicitation but do not focus on short text inputs. [23]: Provide a comprehensive review of deep learning models for text classification, including sentiment analysis but not specifically tied to requirements classification.
Predominant Classifications of User Reviews	Seeks to identify major classifications used in previous research.	[24]: Classify non-functional requirements but do not specifically address user reviews. [25]: Use zero-shot learning for requirements classification but focus on functional vs. non-functional and security requirements.
Classification Methods in Previous Studies	Aims to understand the classification methods used by previous scholars.	[26]: Review requirement prioritization techniques and their empirical evaluations but do not focus on classification methods. [27]: Analyze various requirements prioritization techniques but mainly discuss prioritization rather than classification methods.
Machine Learning Techniques for Enhancing Accuracy	Investigates suitable machine learning techniques to enhance the accuracy of requirement classification and prioritization.	[28]: Propose a semi-automated requirements prioritization method using RankBoost and weighted PageRank. [29]: Review test case prioritization using genetic algorithms, highlighting the potential of machine learning but in the context of testing rather than requirements classification.

D. Selection Process

The researcher carefully read the titles and abstracts of each article to check if they were relevant. Then, they read the entire article to confirm their validity and find the needed information. This process ensured that the results contributed to evidence-based research. Most recent studies employed the Systematic Reviews and Meta-Analyses (PRISMA) method, like those cited in [30]–[32], among others.

The detailed explanation of how the PRISMA technique was used in these studies is as follows:

1) *Identification*: In this step, the researcher searched for relevant studies in different databases, such as IEEE Xplore, Scopus, ScienceDirect, and Web of Science. The researcher used specific and synonym keywords to define the search strategy. This initial phase of the systematic review resulted in 5262 publications related to the study topic from the four databases.

2) *Screening*: In this step, the researcher removed duplicate articles using the duplicate function in Mendeley desktop repositories. Initially, 4731 publications were excluded, leaving 531 papers for further review based on specific inclusion and exclusion criteria. In total, 22 publications were rejected due to duplication.

3) *Eligibility*: In this step, 509 articles were prepared for review. During this stage, the researcher used Python to implement a high-level for screening keywords and article titles and a manual screening to ensure met the inclusion criteria and aligned with the current research objectives. Consequently, 376 data/papers/articles were excluded as they did not qualify due to the out-of-field, title not significantly, abstract not related to the study's objective, and no full-text access. A total of 133 articles were selected (see Fig. 1).

4) *Inclusion*: In this step, the researcher included the final set of studies in the systematic review and meta-analysis. The researcher extracted data from the studies, synthesized the results, and reported the findings.

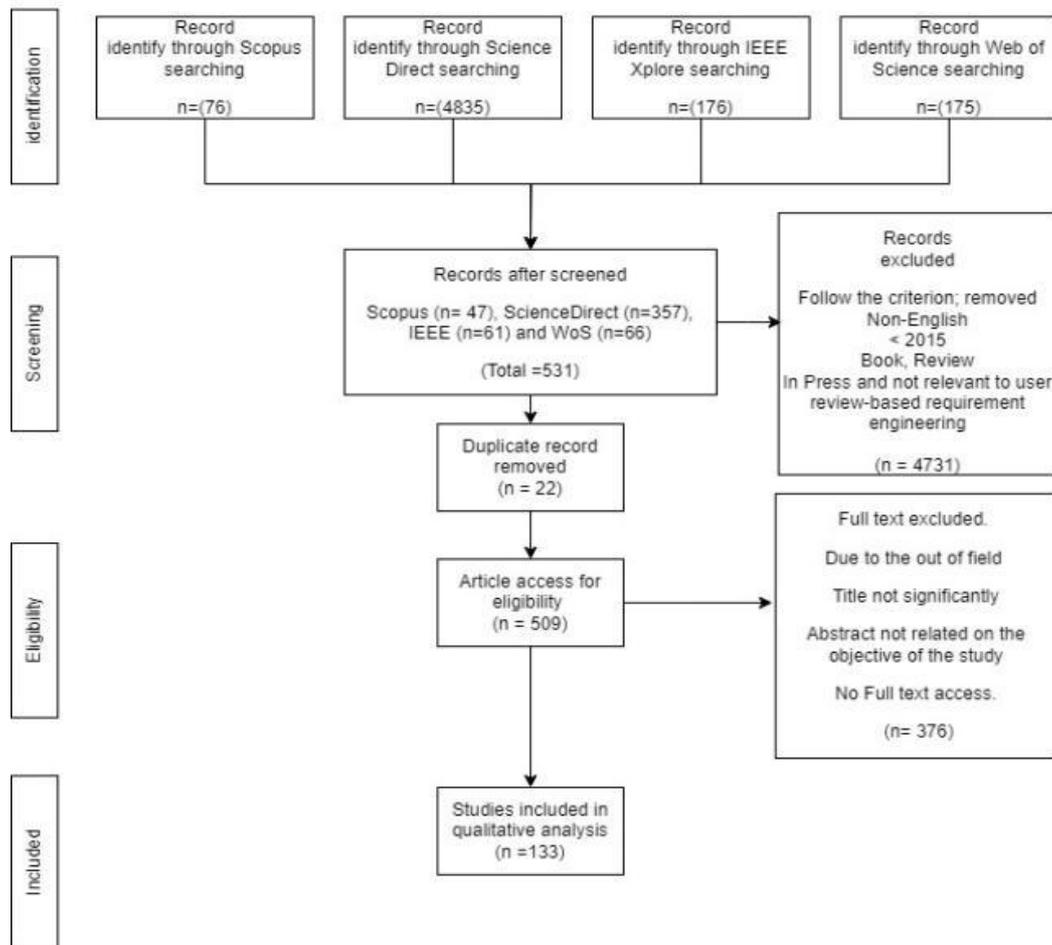


Fig. 1 Flow diagram of search study

III. RESULTS AND DISCUSSION

This section presents our findings for the research questions (RQs) defined in Section I.

A. Challenges in incorporating user reviews (RQ1)

The analysis revealed several challenges and limitations when dealing with user reviews, such as informal forms, unstructured formats, noise, short text, morphological complexities, and dialects, as shown in Table II. It is important to note that the sixth element, dialects, is only present in user reviews that are not in English.

TABLE II
CHALLENGES IN USER REVIEW

#	Challenges in user review	Non-English Review (Authors)	English Review (Authors)
1	Informal form	[33], [34]	[35], [36]
2	Unstructured format	[37], [38]	[39], [40]
3	Noisy word	[41]–[43]	[44], [45]
4	Short text	[46]–[49]	[50], [51]
5	Morphological complexity	[52]–[54]	[55], [56]
6	Dialect	[57]–[61]	

1) *Informal form*: Analysis of user feedback often faces challenges when feedback is received in an informal form. This makes it difficult to obtain meaningful information.

Traditional Natural Language Processing (NLP) models have difficulty dealing with the informal nature of the language users use, which requires a deep understanding of the true meaning behind their words. According to [33], informal expressions such as slang and abbreviations complicate the analysis process. Understanding and addressing these issues in user feedback analysis demands the development of a sophisticated natural language processing model.

2) *Unstructured format*: This irregularity complicates the analysis process, especially when identifying valuable information [62]. According to [63], unstructured text is generally more complicated to process and analyze. To address this problem, we need innovative techniques to understand and extract meaningful information from unstructured feedback. This will help make the analysis and interpretation of feedback data more efficient.

3) *Noise*: According to [64], short texts usually have fewer characters but more noise, affecting classification accuracy. To address this issue, integrating external knowledge with the model helps to understand short texts better and learn additional information [65]. Another approach involves reducing noise by removing unnecessary characters like hashtags, URLs, numbers, punctuation marks, and other symbols [66]. This method improves data clarity, similar to Yang et al. [67], removing noise before applying collocation and part-of-speech (POS) techniques to filter out

meaningless phrases. Both approaches highlight the importance of noise reduction in data preprocessing.

4) *Short text*: Users often write short [68] and incomplete reviews, making analysis difficult. Short texts frequently do not follow proper syntax and typically contain colloquial terms (e.g., LOL, pwd, etc.), phonetic spellings, and other new words [69]. As a result, traditional NLP methods cannot be easily applied because of the short text length, leading to the problem of sparse features [70]. Short texts can often be ambiguous [71], [72] because they frequently contain words or phrases with multiple meanings. Some words have several distinct meanings, and some different words can mean the same thing. This ambiguity makes it difficult to determine the intended meaning of a short text [73]. Therefore, special techniques such as classification and association rule mining are needed for analyzing short texts [74].

5) *Morphological*: Morphology is the study of the internal structure of words, including how words are formed and their relationship to other words in the same language. The form of a word can change depending on the context, which makes it difficult for NLP systems to process. This involves analyzing root words, prefixes, and suffixes [75]. The complexity of a language's morphology can cause problems for part-of-speech tagging because root words can transform into thousands of different forms, leading to data scattering issues, as noted in the context of the Arabic language [53]. Additionally, a language's morphological complexities and various dialects make semantic analysis particularly challenging [52].

6) *Dialects*: Users often use dialects and slang rather than formal language in their social media communications [61]. This can be challenging for NLP systems when analyzing user feedback. For example, [76] implemented a processing unit specifically for Arabic posts. This unit was designed to improve the data quality by removing redundant content, repeated posts, and irrelevant information like timestamps and 'likes'. This approach to data preprocessing is essential to ensure the accuracy of the following analyses.

B. Techniques to Enhance Understanding of User Needs (RQ2)

User reviews represent an important, valuable source [77] of end-user feedback, aiding developers in identifying, categorizing, and prioritizing their software development requirements. Yet, user reviews typically form a short, informal text with a noisy tone, presenting challenges in extracting and analyzing pertinent information. Consequently, previous researchers have suggested and implemented diverse techniques to improve the comprehension of user needs from such concise texts. These techniques include:

1) *Opinion mining*: This technique aims to extract relevant central information such as opinions, sentiments, emotions, or attitudes from comments and map them to software requirements. For example, [78] employed opinion mining to identify relevant parts of comments and associate them with both functional and non-functional requirements. Similarly, [79] emphasizes applying opinion-mining techniques to extract valuable insights from user reviews, which can inform software evolution and future research directions.

2) *Sentiment analysis*: This technique calculates the polarity and subjectivity of user reviews and identifies whether the corpus expresses positive, negative, or neutral sentiments. For instance, [66] used sentiment analysis to assign quantitative values to user reviews based on their polarity and subjectivity scores.

3) *Natural Language Processing (NLP)* is the most widely utilized technique in every study, and it involves processing and understanding natural language texts using various methods. Syntactic parsing, semantic analysis, word embeddings, or transformer models are the techniques that are being used in NLP. For example, Panichella et al. [80] used the Stanford Typed Dependencies parser to represent the grammatical relations between words in sentences and extract features for classification. Hua et al. [81] used knowledge-intensive approaches based on lexical-semantic analysis to improve the accuracy of short-text understanding.

TABLE III
NOVEL MATRIX TAXONOMY FOR USER REVIEW

#	Taxonomy	Definition	Topic
1	Information seeking	Efforts to acquire knowledge or assistance from other developers [80], users, or the software provider [82].	Question
2	Feature requests	Expressions of ideas, suggestions, or needs for enhancements, whether they pertain to the product, services, or their functionalities [80] or specifically to the software or product functionality [82], are essentially requests for improvement [38].	Feature request, content request, promise, improvement request, idea, suggestion, and shortcoming
3	Problem discovery	Statements that define issues and unexpected behaviors [80], or those expressing dissatisfaction or describing anomalies with the software or product [82], indicate potential improvement areas.	Bug report, issues, dissatisfaction, emotion
4	Information giving	Communications that update other developers on planned updates [80] or those that express satisfaction or inform other users or sellers about product functionality [82] play a crucial role in the collaborative development process.	Feature information, Satisfaction, emotion

C. Predominant Classifications of User Reviews (RQ3)

User reviews provide valuable feedback from end-users, helping developers enhance their software products. Previous studies have suggested classifying user reviews in various

ways, including bug reports, functional and non-functional [62], [83]–[86], aspects, usability, user experience [76], and ratings [87]. This study created a new matrix taxonomy by combining definitions from different studies and topics. Its purpose is to categorize user reviews into four main types:

information seeking, feature request, problem discovery, and information giving, as detailed in Table III.

D. Classification Methods in Previous Studies (RQ4)

User reviews represent a precious source of end-user feedback, aiding developers in enhancing their software products. However, user reviews are not homogeneous. Depending on the review and why it was written, they can differ in their types and aspects. Therefore, previous studies have proposed different ways to classify user reviews, such as:

1) *Manual classification*: some researchers have manually labeled user reviews based on predefined categories, such as bug reports, non-functional requirements, usability, user experience, or ratings. Most researchers ([62], [80], [88]) used manual classification for their studies.

2) *Pre-labeled datasets*: Meanwhile, some researchers have used existing datasets labeled by other sources, such as app developers or third-party platforms, to train and evaluate their classification models. For instance, [35] and [89] used pre-labeled datasets for their studies.

3) *Specific aspects*: Prior researchers have focused on specific aspects or attention mechanisms of user reviews, such as user ideas or text mining techniques, and used them as the basis for their classification methods. For instance, Wouters

et al. [90] used user ideas in their study for classification, while Asadabadi et al. [91] used text-mining techniques to extract features for their classification.

4) *Novel approaches*: Prior researchers have studied and proposed novel approaches to user review classification, such as assigning a unique identifier to each requirement, using a hierarchical structure to organize requirements, or using clustering techniques to group similar requirements (Garcia-Lopez et al. [86]). For example, Cai et al. [12] proposed a unique identifier approach. explored a hierarchical structure approach and studies on a clustering technique approach.

5) *Similarity values*: Some researchers have used similarity values as extracted features to classify user reviews based on their relevance or importance to the software development process. Raharjana et al. [66] explored and utilized similarity values to classify user reviews based on their polarity and subjectivity.

Moreover, using the stacking classifier ensemble strategy can also improve classification accuracy by up to 89 percent Sai et al. [92]. For the same reason [93], the experiment showed that combining BERT with CNN, BERT with RNN, and BERT with BiLSTM resulted in good performance, especially regarding accuracy. Combining different methods within one model can enhance classification accuracy (see Table IV).

TABLE IV
OVERVIEW OF METHOD(S) USED BY VARIOUS AUTHORS

Author(s)	Classification Methods											
	LR	CNN	LSTM	Word2Vec	FastText	BERT	ANN	RNN	SVM	Naïve Bayes	RF	BDT
Elhassan et al. [52]		☐	☐	☐	☐							
Sharma et al. [94]		☐						☐				
Sai et al. [92]	☐								☐	☐	☐	
Al-Buraihy et al. [41]									☐			
Agathangelou & Katakis [95]		☐					☐	☐				
Yucel et al. [96]												☐
Mandhasiya et al. [97]			☐			☐						
Qureshi et al. [98]	☐											
Alturayef et al. [99]	☐					☐						

Note: LR: Logistic Regression, RF: Random Forest, BDT: boosted decision trees.

E. Machine Learning Techniques for Enhancing Accuracy (RQ5)

Machine learning techniques use data and algorithms to find patterns and make predictions or decisions based on the analysis. These methods can be applied for various purposes, such as classification and prioritization. Machine learning can:

1) *Automatically analyze user reviews [100] and extract the central information*: user needs, preferences, sentiments, ratings, bug reports, or feature requests.

2) *Categorize the user reviews into several types of requirements classification*: functional or non-functional, usability or user experience, enhancement, or new features.

3) *Assign priority values or ranks to user reviews*: it is based on their importance, urgency, feasibility, or customer satisfaction.

4) *Evaluate the different classification and prioritization models*: performance and accuracy.

Some of the standard machine learning techniques that have been proposed and used for requirement classification and prioritization are:

1) *Naive Bayes*: This technique is based on applying Bayes' theorem, which calculates the probability of a class given a set of features. Naive Bayes assumes that the features are independent of the class. It is simple, fast, and effective for text classification tasks. For example, Maalej and Nabil [6] used Naive Bayes to classify user reviews into bug reports, non-functional requirements, usability, user experience, and ratings.

2) *Support Vector Machine (SVM)*: This technique is based on finding a hyperplane that separates the data points into different classes with the maximum margin. SVM can handle linear and non-linear classification problems using

different kernel functions. SVM is robust [101], [102], accurate [103], and efficient for text classification tasks. For example, Binkhonain and Zhao [83] used SVM to classify user reviews based on non-functional requirements and usability.

3) *Neural Networks*: This technique simulates the structure and function of biological neurons in a network. Neural networks can learn complex and non-linear relationships between inputs and outputs using multiple layers of neurons with different activation functions. They are robust, flexible, and scalable for text classification tasks [104]–[106]. For example, Bhatia and Sharma [107] used neural networks to select the top k features for model training and evaluation.

F. Discussion

Requirement prioritization assigns importance or urgent values to user requirements and ranks them according to their relative significance for software development. For instance, Hujainah et al. [108] used a Binary Search Tree (BST) to rank requirements, while Aziz et al. [109] utilized the Kano model to identify which requirements satisfy the customers the most. Similarly, [110] utilized the Kano model to gather requirements and prioritize service improvements. Bhatia & Sharma [107] proposed using ANOVA f-value to rank a feature, and Asadabadi et al. [91] suggested weighting importance according to the time of review posting.



Fig. 2 Essential elements required for effective requirement prioritization

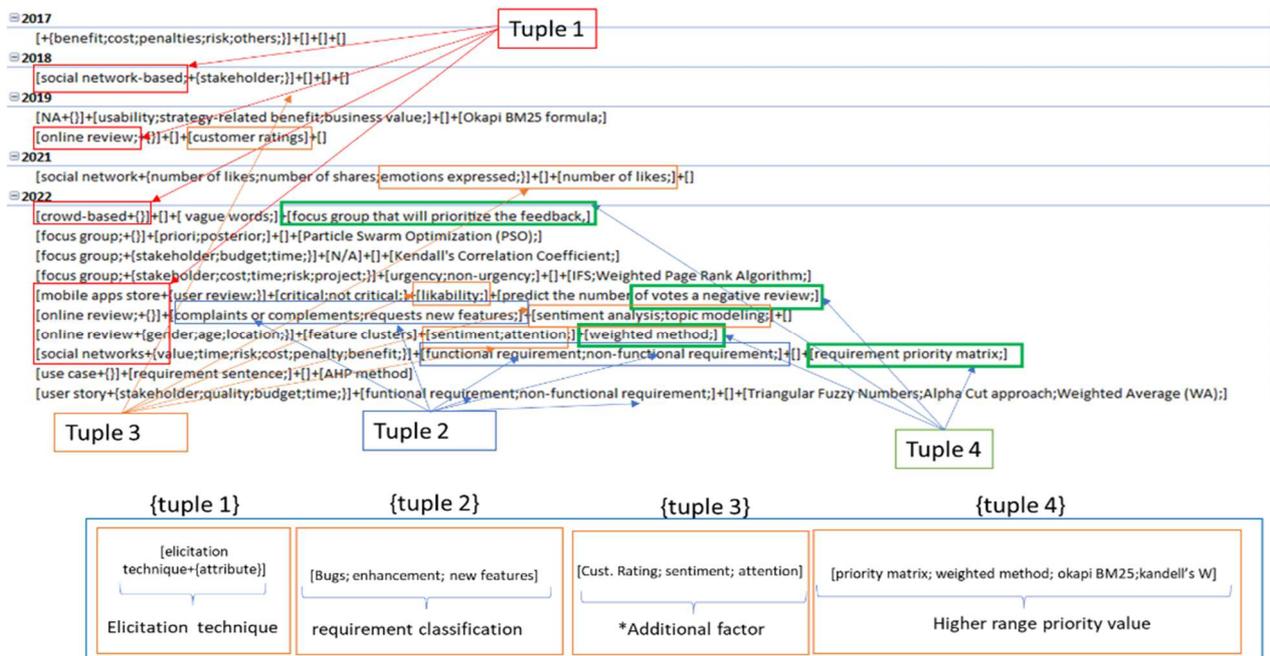


Fig. 3 Example tuple implemented in research

TABLE V
ESSENTIAL ELEMENTS REQUIRED FOR REQUIREMENT PRIORITIZATION

#	Tuple	Description
1	Elicitation technique	Requirements elicitation involves obtaining information from stakeholders to understand their needs and expectations. Standard techniques include stakeholder analysis (identifying impacted parties) and brainstorming (generating ideas). Other methods include interviews, surveys, and user reviews. Each technique has advantages and disadvantages, depending on the project context.
2	Requirement classification	involves grouping user requirements into different categories based on their characteristics, such as functional, non-functional, usability, user experience, bugs, enhancement, or new features. Developers can analyze and manage requirements better using this approach.
3	Additional factor	Factors like ratings, sentiment, and attention mechanisms can influence requirement prioritization. These helps measure importance and assign priority values to user requirements.
4	Higher range priority value	A higher range priority values represent the urgency of user requirements. Various methods, such as Best Worst Scaling, Kendall's W, or Okapi BM25, can be used. These values help rank requirements and select the most valuable ones for the next release.

Through the literature search, the researcher found a few common elements that need to be considered to develop

effective requirement prioritization or any research regarding prioritization. These multifaceted elements have been

collectively defined as a tuple within the context of this study. The tuples are (1) elicitation technique, (2) requirement classification, (3) additional factor, and (4) higher range priority value. These four tuples are derived from synthesis readings (see Table VI) and revealed a novel tuple pattern in prioritization to perform effective requirement prioritization.

Table V explains each tuple and needs to be considered as an essential element, as presented in Fig. 2 and Fig. 3. This requirement prioritization can help developers select the most valuable and feasible requirements for the next release and allocate resources and time accordingly.

TABLE VI
EXTRACTION AND MATCH TUPLE PATTERN

Year	Extract and match tuple pattern	Author
2021	[+{}][energy;entertainment;health;safety;other;]+[+[]]	[107]
	[+{}][high;medium;low;]+[importance;cost;]+[WSM method]	[108]
	[social network;+{number of likes;number of shares;emotions expressed;}]++[number of likes;]+[]	[111]
2022	[focus group;+{stakeholder;budget;time;}]++[N/A]+[Kendall's Correlation Coefficient;]	[112]
	[focus group;+{stakeholder;cost;time;risk;project;}]++[urgency;non-urgency;]+[IFS;Weighted Page Rank Algorithm;]	[113]
	[mobile apps store;+{user review;}]++[critical;not critical;]+[likability;]+[predict the number of votes a negative review;]	[11]
	[online review;+{gender;age;location;}]++[feature clusters]+[sentiment;attention;]+[weighted method;]	[12]
	[social network;+{value;time;risk;cost;penalty;benefit;}]++[functional requirement;non-functional requirement;]+[requirement priority matrix;]	[85]
	[use case;+{}][requirement sentence;]+[AHP method]	[114]
2023	[user story;+{stakeholder;quality;budget;time;}]++[functional requirement;non-functional requirement;]+[Triangular Fuzzy Numbers;Alpha Cut approach;Weighted Average (WA);]	[84]
	[+{}][poor;mixed;good;]+[time of reviews;review usefulness;]+[[]]	[91]

IV. CONCLUSION

Requirement prioritization is essential in software development. It enables developers to effectively identify and rank user requirements based on their significance, facilitating informed decision-making regarding resource allocation and project planning. The study highlights four critical elements—elicitation technique, requirement classification, additional factors, and higher range priority value—that collectively enhance the prioritization process.

The study also underscores the critical importance of extracting software requirements from user reviews, highlighting the challenges posed by such feedback's informal and unstructured nature. Proposing a new taxonomy for categorizing user reviews and advocating for advanced machine learning techniques paves the way for future research to enhance the adaptability and efficiency of requirements engineering processes.

ACKNOWLEDGMENT

We thank the Faculty of Computer Science and Information Technology (FCSIT) of the University Putra Malaysia (UPM) for financial assistance.

REFERENCES

[1] E. Erhieyovwe and E. E. Ogheneovo, "Software Maintenance and Evolution: The Implication for Software Development," *West African J. Ind. Acad. Res.*, vol. 7, no. 1, pp. 81–92, 2013.

[2] V. Rajlich, "Software evolution and maintenance," in *Future of Software Engineering Proceedings*, Dec. 2014, pp. 133–144. doi:10.1145/2593882.2593893.

[3] M. P. S. S. Gôlo, A. F. Araújo, R. G. Rossi, and R. M. Marcacini, "Detecting relevant app reviews for software evolution and maintenance through multimodal one-class learning," *Inf. Softw. Technol.*, vol. 151, no. July, p. 106998, 2022, doi:10.1016/j.infsof.2022.106998.

[4] R. Madani, A. Ez-Zahout, and F. Omary, "Extracting contextual insights from user reviews for recommender systems: a novel method," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 35, no. 1, pp. 542–550, 2024, doi: 10.11591/ijeecs.v35.i1.pp542-550.

[5] S. F. He, X. H. Pan, Y. M. Wang, D. G. Zamora, and L. Martínez, "A novel Multi-Criteria Decision Making framework based on Evidential

Reasoning dealing with missing information from online reviews," *Inf. Fusion*, 2024, doi: 10.1016/j.inffus.2024.102264.

[6] W. Maalej and H. Nabil, "Bug report, feature request, or simply praise? On automatically classifying app reviews," in *2015 IEEE 23rd International Requirements Engineering Conference, RE 2015 - Proceedings*, 2015, pp. 116–125. doi: 10.1109/RE.2015.7320414.

[7] S. Malgaonkar, S. A. Licorish, and B. T. R. Savarimuthu, "Prioritizing user concerns in app reviews – A study of requests for new features, enhancements and bug fixes," *Inf. Softw. Technol.*, 2022, doi:10.1016/j.infsof.2021.106798.

[8] H. Wu, W. Deng, X. Niu, and C. Nie, "Identifying Key Features from App User Reviews," in *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, Jul. 2021, pp. 922–932. doi: 10.1109/ICSE43902.2021.00088.

[9] K. Selvasheela, A. M. Abirami, and A. K. Askarunisa, "Effective Customer Review Analysis Using Combined Capsule Networks with Matrix Factorization Filtering," *Computer Systems Science and Engineering*, 2023. doi: 10.32604/csse.2023.029148.

[10] T. Yang, C. Gao, J. Zang, D. Lo, and M. R. Lyu, "TOUR: Dynamic Topic and Sentiment Analysis of User Reviews for Assisting App Release," Sep. 2021, pp. 708–712. doi: 10.1145/3442442.3458612.

[11] M. Fereidouni, A. Mosharraf, U. Farooq, and A. B. Siddique, "Proactive Prioritization of App Issues via Contrastive Learning," in *Proceedings - 2022 IEEE International Conference on Big Data, Big Data 2022*, 2022. doi: 10.1109/BigData55660.2022.10020586.

[12] M. Cai, Y. Tan, B. Ge, Y. Dou, G. Huang, and Y. Du, "PURA: A Product-and-User Oriented Approach for Requirement Analysis From Online Reviews," *IEEE Syst. J.*, vol. 16, no. 1, pp. 566–577, 2022, doi:10.1109/JSYST.2021.3067334.

[13] K. Zahoor and N. Z. Bawany, "Explainable artificial intelligence approach towards classifying educational android app reviews using deep learning," *Interact. Learn. Environ.*, 2023, doi:10.1080/10494820.2023.2212708.

[14] N. Saraswathi, T. Sasi Rooba, and S. Chakaravathi, "Improving the accuracy of sentiment analysis using a linguistic rule-based feature selection method in tourism reviews," *Meas. Sensors*, 2023, doi:10.1016/j.measen.2023.100888.

[15] R. R. Mekala, A. Irfan, E. C. Groen, A. Porter, and M. Lindvall, "Classifying User Requirements from Online Feedback in Small Dataset Environments using Deep Learning," 2021, pp. 139–149. doi:10.1109/RE51729.2021.00020.

[16] J. del Sagrado, J. A. Sierra Ibañez, and I. M. del Águila, "An estimation of distribution algorithm based on interactions between requirements to solve the bi-objective Next Release Problem," *J. Syst. Softw.*, 2023, doi: 10.1016/j.jss.2023.111632.

[17] E. Afful-Dadzie, A. Afful-Dadzie, and S. B. Egala, "Social media in health communication: A literature review of information quality," *Health Information Management Journal*, 2023. doi:10.1177/1833358321992683.

- [18] W. Maalej, V. Biryuk, J. Wei, and F. Panse, "On the Automated Processing of User Feedback," *arXiv Prepr. arXiv*, pp. 1–28, 2024.
- [19] B. Kitchenham, L. Madeyski, and D. Budgen, "SEGRESS: Software Engineering Guidelines for RReporting Secondary Studies," Jun. 2023. doi:10.1109/TSE.2022.3174008.
- [20] N. Genc-Nayebi and A. Abran, "A systematic literature review: Opinion mining studies from mobile app store user reviews," *J. Syst. Softw.*, 2017, doi: 10.1016/j.jss.2016.11.027.
- [21] A. A. Al-Subaihini, F. Sarro, S. Black, L. Capra, and M. Harman, "App Store Effects on Software Engineering Practices," *IEEE Trans. Softw. Eng.*, 2021, doi: 10.1109/TSE.2019.2891715.
- [22] H. Meth, M. Brhel, and A. Maedche, "The state of the art in automated requirements elicitation," *Inf. Softw. Technol.*, vol. 55, no. 10, pp. 1695–1709, Jul. 2013, doi: 10.1016/j.infsof.2013.03.008.
- [23] W. Dawei, R. Alfred, J. H. Obit, and C. K. On, "A Literature Review on Text Classification and Sentiment Analysis Approaches," *Lect. Notes Electr. Eng.*, vol. 724, pp. 305–323, 2021, doi: 10.1007/978-981-33-4069-5_26.
- [24] P. Loucopoulos, J. Sun, L. Zhao, and F. Heidari, "A systematic classification and analysis of NFRs," in *19th Americas Conference on Information Systems, AMCIS 2013 - Hyperconnected World: Anything, Anywhere, Anytime*, 2013, vol. 1, pp. 208–217.
- [25] W. Alhoshan, A. Ferrari, and L. Zhao, "Zero-shot learning for requirements classification: An exploratory study," *Inf. Softw. Technol.*, vol. 159, p. 107202, Oct. 2023, doi:10.1016/j.infsof.2023.107202.
- [26] F. A. Bukhsh, Z. A. Bukhsh, and M. Daneva, "A systematic literature review on requirement prioritization techniques and their empirical evaluation," *Comput. Stand. Interfaces*, vol. 69, p. 103389, Feb. 2020, doi: 10.1016/j.csi.2019.103389.
- [27] F. Hujainah, R. B. A. Bakar, M. A. Abdulgaber, and K. Z. Zamli, "Software Requirements Prioritisation: A Systematic Literature Review on Significance, Stakeholders, Techniques and Challenges," *IEEE Access*, 2018, doi: 10.1109/ACCESS.2018.2881755.
- [28] F. Shao, R. Peng, H. Lai, and B. Wang, "DRank: A semi-automated requirements prioritization method based on preferences and dependencies," *J. Syst. Softw.*, 2017, doi: 10.1016/j.jss.2016.09.043.
- [29] A. Bajaj and O. P. Sangwan, "A Systematic Literature Review of Test Case Prioritization Using Genetic Algorithms," *IEEE Access*, vol. 7, pp. 126355–126375, 2019, doi: 10.1109/ACCESS.2019.2938260.
- [30] J. Dąbrowski, E. Letier, A. Perini, and A. Susi, "Analysing app reviews for software engineering: a systematic literature review," *Empir. Softw. Eng.*, vol. 27, no. 2, p. 43, Sep. 2022, doi: 10.1007/s10664-021-10065-7.
- [31] M. Srivani, A. Murugappan, T. Mala, S. M., A. Murugappan, and M. T., "Cognitive computing technological trends and future research directions in healthcare — A systematic literature review," *Artif. Intell. Med.*, vol. 138, p. 102513, 2023, doi: 10.1016/j.artmed.2023.102513.
- [32] S. Stradowski and L. Madeyski, "Machine learning in software defect prediction: A business-driven systematic mapping study," *Inf. Softw. Technol.*, vol. 155, p. 107128, 2023, doi:10.1016/j.infsof.2022.107128.
- [33] D. Cirqueira, M. Fontes Pinheiro, A. Jacob, F. Lobato, and A. Santana, "A Literature Review in Preprocessing for Sentiment Analysis for Brazilian Portuguese Social Media," 2019, pp. 746–749. doi:10.1109/WI.2018.00008.
- [34] M. Garouani and J. Kharroubi, "MAC: An Open and Free Moroccan Arabic Corpus for Sentiment Analysis," *Lect. Notes Networks Syst.*, vol. 393, pp. 849–858, 2022, doi: 10.1007/978-3-030-94191-8_68.
- [35] A. F. De Araújo and R. M. Marcacini, "RE-BERT: automatic extraction of software requirements from app reviews using BERT language model," in *SAC '21: The 36th ACM/SIGAPP Symposium on Applied Computing*, Oct. 2021, pp. 1321–1327. doi:10.1145/3412841.3442006.
- [36] M. Tubishat, N. Idris, and M. Abushariah, "Explicit aspects extraction in sentiment analysis using optimal rules combination," *Futur. Gener. Comput. Syst.*, vol. 114, pp. 448–480, Jul. 2021, doi:10.1016/j.future.2020.08.019.
- [37] K. A. F. A. Samah, N. M. N. Azharludin, L. S. Riza, M. N. H. H. Jono, and N. A. Moketar, "Classification and visualization: Twitter sentiment analysis of Malaysia's private hospitals," *IAES Int. J. Artif. Intell.*, vol. 12, no. 4, pp. 1793–1802, 2023, doi:10.11591/ijai.v12.i4.pp1793-1802.
- [38] W. M. S. Yafooz, A. Alsaedi, and A.-H. M. Emara, "AraDS: Arabic Datasets for Text Mining Approaches," 2023. doi:10.1109/ICSCA57840.2023.10087675.
- [39] H. Harkous, S. T. Peddinti, R. Khandelwal, A. Srivastava, and N. Taft, "Hark: A Deep Learning System for Navigating Privacy Feedback at Scale," 2022, vol. 2022-May, pp. 2469–2486. doi:10.1109/SP46214.2022.9833729.
- [40] Y. Han, G. Nanda, and M. Moghaddam, "Attribute-Sentiment-Guided Summarization of User Opinions From Online Reviews," *J. Mech. Des.*, 2023, doi: 10.1115/1.4055736.
- [41] E. Al-Buraihy, R. U. Khan, W. Dan, and M. Ullah, "An ML-Based Classification Scheme for Analyzing the Social Network Reviews of Yemeni People," *Int. Arab J. Inf. Technol.*, vol. 19, no. 6, pp. 904–914, 2022, doi: 10.34028/iajit/19/6/8.
- [42] K. K. Gupta, S. Chennabasavraj, N. Garera, and A. Ekbal, "Product Review Translation: Parallel Corpus Creation and Robustness towards User-generated Noisy Text," in *ECNLP 2021 - 4th Workshop on e-Commerce and NLP, Proceedings*, 2021. doi:10.18653/v1/2021.ecnlp-1.21.
- [43] N. A. K. M. Haris, S. Mutalib, A. M. A. Malik, S. Abdul-Rahman, and S. N. K. Kamarudin, "Sentiment classification from reviews for tourism analytics," *Int. J. Adv. Intell. Informatics*, vol. 9, no. 1, pp. 108–120, 2023, doi: 10.26555/ijain.v9i1.1077.
- [44] P. R. Henao, J. Fischbach, D. Spies, J. Frattini, and A. Vogelsang, "Transfer Learning for Mining Feature Requests and Bug Reports from Tweets and App Store Reviews," Jul. 2021, pp. 80–86. doi:10.1109/rew53955.2021.00019.
- [45] N. Al Kilani, R. Tailakh, A. Hanani, N. Al Kilani, R. Tailakh, and A. Hanani, "Automatic Classification of Apps Reviews for Requirement Engineering: Exploring the Customers Need from Healthcare Applications," in *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, Jul. 2019, pp. 541–548. doi: 10.1109/snams.2019.8931820.
- [46] I. Alagha, "Leveraging Knowledge-Based Features With Multilevel Attention Mechanisms for Short Arabic Text Classification," *IEEE Access*, vol. 10, pp. 51908–51921, Sep. 2022, doi:10.1109/ACCESS.2022.3175306.
- [47] A. Motwakel *et al.*, "Convolutional Deep Belief Network Based Short Text Classification on Arabic Corpus," *Comput. Syst. Sci. Eng.*, vol. 45, no. 3, pp. 3097–3113, Aug. 2023, doi: 10.32604/csse.2023.033945.
- [48] X. Xu, B. Li, Y. Shen, B. Luo, C. Zhang, and F. Hao, "Short Text Classification Based on Hierarchical Heterogeneous Graph and LDA Fusion," *Electronics*, vol. 12, no. 12, p. 2560, Sep. 2023, doi:10.3390/electronics12122560.
- [49] S. M. Khalifa, M. I. Marie, and M. M. El-Defrawi, "Aspects Detection Model for Users' Reviews Using Machine Learning Techniques," *J. Theor. Appl. Inf. Technol.*, vol. 100, no. 19, pp. 5776–5786, 2022.
- [50] M. J. Hazar, M. Zrigui, and M. Maraoui, "Learner comments-based Recommendation system," *Procedia Comput. Sci.*, vol. 207, pp. 2000–2012, 2022, doi: 10.1016/j.procs.2022.09.259.
- [51] K. Ahkoul, M. Mustapha, M. Khadija, and M. Rachid, "A review of the text to SQL frameworks," in *ACM International Conference Proceeding Series*, 2021. doi: 10.1145/3454127.3457619.
- [52] N. Elhassan *et al.*, "Arabic Sentiment Analysis Based on Word Embeddings and Deep Learning," *Computers*, vol. 12, no. 6, 2023, doi:10.3390/computers12060126.
- [53] A. M. Nerabie, M. AlKhatib, S. S. Mathew, M. El Barachi, and F. Oroumchian, "The impact of Arabic part of speech tagging on sentiment analysis: A new corpus and deep learning approach," 2021, vol. 184, pp. 148–155. doi: 10.1016/j.procs.2021.03.026.
- [54] A. Elsaid, A. Mohammed, L. F. Ibrahim, and M. M. Sakre, "A Comprehensive Review of Arabic Text Summarization," *IEEE Access*, vol. 10, pp. 38012–38030, 2022, doi:10.1109/access.2022.3163292.
- [55] G. Lo Bosco, G. Pilato, and D. Schicchi, "DeepEva: A deep neural network architecture for assessing sentence complexity in Italian and English languages," *Array*, vol. 12, p. 100097, 2021, doi:10.1016/j.array.2021.100097.
- [56] M. Popović, A. Way, A. Poncelas, and M. Brkić Bakarić, "On Machine Translation of User Reviews," in *International Conference Recent Advances in Natural Language Processing*, Nov. 2021, pp. 1109–1118. doi:10.26615/978-954-452-072-4_124.
- [57] I. Afyouni, B. Kakah, and M. Bazazeh, "Arabic Geo-Social Event Detection using a Hybrid Learning Architecture," 2023. doi:10.1109/SmartNets58706.2023.10216180.
- [58] M. Hadwan, M. Al-Sarem, F. Saeed, and M. A. Al-Hagery, "An Improved Sentiment Classification Approach for Measuring User Satisfaction toward Governmental Services' Mobile Apps Using Machine Learning Methods with Feature Engineering and SMOTE Technique," *Appl. Sci.*, vol. 12, no. 11, 2022, doi:10.3390/app12115547.

- [59] A. A. Al Shamsi and S. Abdallah, "Text Mining Techniques for Sentiment Analysis of Arabic Dialects: Literature Review," *Adv. Sci. Technol. Eng. Syst. J.*, 2021, doi: 10.25046/aj0601112.
- [60] T. Mahmud, M. Ptaszynski, J. Eronen, and F. Masui, "Cyberbullying detection for low-resource languages and dialects: Review of the state of the art," *Inf. Process. Manag.*, 2023, doi:10.1016/j.ipm.2023.103454.
- [61] A. H. Hussein, I. F. Moawad, and R. M. Badry, "Arabic Sentiment Analysis for Multi-dialect Text using Machine Learning Techniques," *Int. J. Adv. Comput. Sci. Appl.*, 2021, doi:10.14569/ijacsa.2021.0121286.
- [62] M. Lu and P. Liang, "Automatic Classification of Non-Functional Requirements from Augmented App User Reviews," *ACM Int. Conf. Proceeding Ser.*, vol. Part F1286, pp. 344–353, Jul. 2017, doi:10.1145/3084226.3084241.
- [63] G. Alexander, M. Bahja, and G. F. Butt, "Automating Large-scale Health Care Service Feedback Analysis: Sentiment Analysis and Topic Modeling Study," *JMIR Med. Informatics*, 2022, doi:10.2196/29385.
- [64] L. Xu, Z. Xue, and H. Huang, "Short text semantic feature extension and classification based on LDA," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 715, no. 1, p. 12110, Sep. 2020, doi: 10.1088/1757-899X/715/1/012110.
- [65] Y. Zhou, C. Li, G. Huang, Q. Guo, H. Li, and X. Wei, "A Short-Text Similarity Model Combining Semantic and Syntactic Information," *Electronics*, vol. 12, no. 14, p. 3126, Aug. 2023, doi:10.3390/electronics12143126.
- [66] I. K. Raharjana, V. Aprillya, B. Zaman, A. Justitia, and S. S. M. Fauzi, "Enhancing software feature extraction results using sentiment analysis to aid requirements reuse," *Computers*, 2021, doi:10.3390/computers10030036.
- [67] C. Yang, L. Wu, C. Yu, and Y. Zhou, "A Phrase-Level User Requests Mining Approach in Mobile Application Reviews: Concept, Framework, and Operation," *Information*, vol. 12, no. 5, p. 177, Jul. 2021, doi: 10.3390/info12050177.
- [68] A. Pradhan, M. R. Senapati, and P. K. Sahu, "ABET: an affective emotion-topic method of biterns for emotion recognition from the short texts," *J. Ambient Intell. Humaniz. Comput.*, 2023, doi:10.1007/s12652-022-03799-9.
- [69] N. Jha and A. Mahmoud, "Using frame semantics for classifying and summarizing application store reviews," *Empir. Softw. Eng.*, 2018, doi: 10.1007/s10664-018-9605-x.
- [70] W. Pang, "Short Text Classification via Term Graph," *arXiv Prepr. arXiv2001.10338*, Sep. 2020.
- [71] S. Kinariwala and S. Deshmukh, "Short text topic modelling using local and global word-context semantic correlation," *Multimed. Tools Appl.*, 2023, doi: 10.1007/s11042-023-14352-x.
- [72] N. Choudhary, C. C. Aggarwal, K. Subbian, and C. K. Reddy, "Self-supervised Short-text Modeling through Auxiliary Context Generation," *ACM Trans. Intell. Syst. Technol.*, 2022, doi: 10.1145/3511712.
- [73] Z. H. Amur, Y. Kwang Hooi, H. Bhanbhro, K. Dahri, and G. M. Soomro, "Short-Text Semantic Similarity (STSS): Techniques, Challenges and Future Perspectives," *Appl. Sci.*, vol. 13, no. 6, p. 3911, Sep. 2023, doi: 10.3390/app13063911.
- [74] A. Al-Hawari, H. Najadat, and R. Shatnawi, "Classification of application reviews into software maintenance tasks using data mining techniques," *Softw. Qual. J.*, vol. 29, no. 3, pp. 667–703, Jul. 2021, doi: 10.1007/s11219-020-09529-8.
- [75] M. Maziyah Mohamed, M. J. Yap, Q. W. Chee, and D. Jared, "Malay Lexicon Project 2: Morphology in Malay word recognition," *Mem. Cogn.*, 2023, doi: 10.3758/s13421-022-01337-8.
- [76] M. Itani, C. Roast, and S. Al-Khayatt, "Developing Resources for Sentiment Analysis of Informal Arabic Text in Social Media," *Procedia Comput. Sci.*, vol. 117, pp. 129–136, 2017, doi:10.1016/j.procs.2017.10.101.
- [77] R. Meneses, C. Brito, B. Lopes, and R. Correia, "Satisfaction and Dissatisfaction in Wine Tourism: A User-Generated Content Analysis," *Tour. Hosp. Res.*, 2023, doi: 10.1177/14673584231191989.
- [78] V. M. A. de Lima et al., "Temporal dynamics of requirements engineering from mobile app reviews," *PeerJ*, vol. 8, 2022, doi:10.7717/peerj-cs.874.
- [79] S. Okuboyejo, "Examining Users' Concerns while Using Mobile Learning Apps," *Int. J. Interact. Mob. Technol.*, 2021, doi:10.3991/ijim.v15i15.22345.
- [80] S. Panichella, A. Di Sorbo, E. Guzman, C. A. Visaggio, G. Canfora, and H. C. Gall, "How can i improve my app? Classifying user reviews for software maintenance and evolution," 2015, pp. 281–290. doi:10.1109/icsm.2015.7332474.
- [81] W. Hua, Z. Wang, H. Wang, K. Zheng, and X. Zhou, "Short text understanding through lexical-semantic analysis," *2015 IEEE 31st Int. Conf. Data Eng.*, pp. 495–506, Aug. 2015, doi:10.1109/icde.2015.7113309.
- [82] K. Srisopha, P. Behnamghader, and B. Boehm, "Do consumers talk about the software in my product? An Exploratory Study of IoT Products on Amazon," *CLEI Electronic J.*, vol. 22, no. 1, 2019, doi:10.19153/cleiej.22.1.1.
- [83] M. Binkhonain and L. Zhao, "A machine learning approach for hierarchical classification of software requirements," *Mach. Learn. with Appl.*, vol. 12, p. 100457, 2023, doi:10.1016/j.mlwa.2023.100457.
- [84] N. H. Borhan, H. Zulzalil, S. Hassan, and N. M. Ali, "A Hybrid Prioritization Approach by integrating non-Functional and Functional User Stories in Agile-Scrum Software Development (i-USPA):A preliminary study," in *2022 IEEE International Conference on Computing, ICOCO 2022*, 2022, doi:10.1109/icoco56118.2022.10031863.
- [85] R. Devadas and N. G. Cholli, "Multi aspects based requirements prioritization for large scale software using deep neural lagrange multiplier," in *1st IEEE International Conference on Smart Technologies and Systems for Next Generation Computing, ICSTSN 2022*, 2022, pp. 1–6. doi: 10.1109/icstsn53084.2022.9761298.
- [86] D. Garcia-Lopez, M. Segura-Morales, and E. Loza-Aguirre, "Improving the quality and quantity of functional and non-functional requirements obtained during requirements elicitation stage for the development of e-commerce mobile applications: an alternative reference process model," *IET Softw.*, vol. 14, no. 7th International Conference on Software Process Improvement (CIMPS) CL-CUCEI, Guadalajara, MEXICO, pp. 148–158, 2020, doi: 10.1049/iet-sen.2018.5443.
- [87] W. Maalej et al., "On the automatic classification of app reviews," *Requir. Eng.*, vol. 21, no. 3, pp. 311–331, Jul. 2016, doi:10.1007/s00766-016-0251-9.
- [88] M. A. H. Wadud, M. M. Kabir, M. F. Mridha, M. A. Ali, M. A. Hamid, and M. M. Monowar, "How can we manage Offensive Text in Social Media - A Text Classification Approach using LSTM-BOOST," *Int. J. Inf. Manag. Data Insights*, vol. 2, no. 2, p. 100095, 2022, doi:10.1016/j.jjime.2022.100095.
- [89] Y. Han and M. Moghaddam, "Analysis of sentiment expressions for user-centered design," *Expert Syst. Appl.*, 2021, doi:10.1016/j.eswa.2021.114604.
- [90] J. Wouters, A. Menkveld, S. Brinkkemper, and F. Dalpiaz, "Crowd-based requirements elicitation via pull feedback: method and case studies," *Requir. Eng.*, vol. 27, no. 4, pp. 429–455, 2022, doi:10.1007/s00766-022-00384-6.
- [91] M. R. Asadabadi, M. Saberi, N. S. Sadghiani, O. Zwikael, and E. Chang, "Enhancing the analysis of online product reviews to support product improvement: integrating text mining with quality function deployment," *J. Enterp. Inf. Manag.*, vol. 36, no. 1, pp. 275–302, 2023, doi: 10.1108/jeim-03-2021-0143.
- [92] K. J. Sai, S. S. Chakravarthi, S. Sountharajan, and E. Suganya, "Ensemble Learning Solution for the Aspect-based Sentimental Analysis on IMDB reviews," 2023, doi:10.1109/wispnet57748.2023.10134144.
- [93] A. Bello, S. C. Ng, and M. F. Leung, "A BERT Framework to Sentiment Analysis of Tweets," *Sensors*, 2023, doi:10.3390/s23010506.
- [94] S. P. Sharma, L. Singh, and R. Tiwari, "Prediction of customer review's helpfulness based on sentences encoding using CNN-BiGRU model," *J. Auton. Intell.*, vol. 6, no. 3, 2023, doi:10.32629/jai.v6i3.699.
- [95] P. Athangelou and I. Katakis, "Balancing between holistic and cumulative sentiment classification," *Online Soc. Networks Media*, vol. 29, 2022, doi: 10.1016/j.osnem.2022.100199.
- [96] A. Yucel, M. Caglar, H. Ahady Dolatsara, B. George, and A. Dag, "Predicting hotel reviews from sentiment: a multinomial classification framework," *J. Model. Manag.*, vol. 17, no. 2, pp. 697–714, 2022, doi:10.1108/JM2-09-2020-0255.
- [97] D. G. Mandhasiya, H. Murfi, A. Bustamam, and P. Anki, "Evaluation of Machine Learning Performance Based on BERT Data Representation with LSTM Model to Conduct Sentiment Analysis in Indonesian for Predicting Voices of Social Media Users in the 2024 Indonesia Presidential Election," 2022, pp. 441–446. doi:10.1109/icoiact55506.2022.9972206.

- [98] M. A. Qureshi *et al.*, "Sentiment Analysis of Reviews in Natural Language: Roman Urdu as a Case Study," *IEEE Access*, vol. 10, pp. 24945–24954, 2022, doi: 10.1109/access.2022.3150172.
- [99] N. Alturayef, H. Aljamaan, and J. Hassine, "An automated approach to aspect-based sentiment analysis of apps reviews using machine and deep learning," *Autom. Softw. Eng.*, vol. 30, no. 2, p. 30, Oct. 2023, doi: 10.1007/s10515-023-00397-7.
- [100] S. H. Lee, H. Lee, and J. H. Kim, "Enhancing the Prediction of User Satisfaction with Metaverse Service Through Machine Learning," *Comput. Mater. Contin.*, 2022, doi: 10.32604/cmc.2022.027943.
- [101] A. V. Asimit, I. Kyriakou, S. Santoni, S. Scognamiglio, and R. Zhu, "Robust Classification via Support Vector Machines," *Risks*, 2022, doi: 10.3390/risks10080154.
- [102] D. B. M. Preethi, M. G. P. Kavyasri, and M. V. S. Akshaya, "Analysis on Automobile Reviews using Machine Learning Techniques," *Int. J. Res. Appl. Sci. Eng. Technol.*, 2023, doi: 10.22214/ijraset.2023.51918.
- [103] A. A. Qureshi, M. Ahmad, S. Ullah, M. N. Yasir, F. Rustam, and I. Ashraf, "Performance evaluation of machine learning models on large dataset of android applications reviews," *Multimed. Tools Appl.*, 2023, doi: 10.1007/s11042-023-14713-6.
- [104] S. Lyu and J. Liu, "Convolutional recurrent neural networks for text classification," *J. Database Manag.*, 2021, doi:10.4018/JDM.2021100105.
- [105] S. M. Mishal and M. M. Hamad, "Text Classification Using Convolutional Neural Networks," *Fusion Pract. Appl.*, 2022, doi:10.54216/FPA.070105.
- [106] H. J. Alantari, I. S. Currim, Y. Deng, and S. Singh, "An empirical comparison of machine learning methods for text-based sentiment analysis of online consumer reviews," *Int. J. Res. Mark.*, 2022, doi:10.1016/j.ijresmar.2021.10.011.
- [107] K. Bhatia and A. Sharma, "Sector classification for crowd-based software requirements," 2021, pp. 1312–1320. doi:10.1145/3412841.3442005.
- [108] F. Hujainah, R. Binti Abu Bakar, A. B. Nasser, B. Al-haimi, and K. Z. Zamli, "SRPTackle: A semi-automated requirements prioritisation technique for scalable requirements of software system projects," *Inf. Softw. Technol.*, 2021, doi: 10.1016/j.infsof.2020.106501.
- [109] N. A. A. Aziz, S. Hassan, and N. Admodisastro, "Prioritizing based on crowd preferences to requirements elicited from crowd's sentiments," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 8 S, pp. 341–346, 2019.
- [110] M. Zhao, M. Liu, C. Xu, and C. Zhang, "Classifying travellers' requirements from online reviews: an improved Kano model," *Int. J. Contemp. Hosp. Manag.*, 2024, doi: 10.1108/ijchm-06-2022-0726.
- [111] V. Gupta, "Comment on 'A Social Network Based Process to Minimize In-Group Biasedness during Requirement Engineering,'" *IEEE Access*. 2021. doi: 10.1109/access.2021.3073379.
- [112] A. Martinis, D. Tzimos, V. C. Gerogiannis, and L. H. Son, "A Multiple Stakeholders' Software Requirements Prioritization Approach based on Intuitionistic Fuzzy Sets," in *CTISC 2022 - 2022 4th International Conference on Advances in Computer Technology, Information Science and Communications*, 2022. doi:10.1109/ctisc54888.2022.9849773.
- [113] A. Gupta and C. Gupta, "A novel collaborative requirement prioritization approach to handle priority vagueness and inter-relationships," *J. King Saud Univ. - Comput. Inf. Sci.*, 2022, doi:10.1016/j.jksuci.2019.12.002.
- [114] M. Z. Naufal Maulana and D. Siahaan, "Use Case-Based Analytical Hierarchy Process Method for Software Requirements Prioritization," in *Proceeding - 6th International Conference on Information Technology, Information Systems and Electrical Engineering: Applying Data Sciences and Artificial Intelligence Technologies for Environmental Sustainability, ICITISEE 2022*, 2022. doi:10.1109/icitisee57756.2022.10057944.