2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)

Non-functional Requirements Prioritization: A Systematic Literature Review

Khush Bakht Ijaz, Irum Inayat
Software Engineering and Automation Lab (SEAL)
National University of Computer and Emerging Sciences (NUCES)
Islamabad, Pakistan
i171097@nu.edu.pk , irum.inayat@nu.edu.pk

Faiza Allah Bukhsh
Datamanagement and Bioinformatics
Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS)
University of Twente (UT)
Enchede, The Netherlands
f.a.bukhsh@utwente.nl

***Abstract*—Continuous delivery and rapidly changing requirements in agile environments force the developers to put non-functional requirements (NFRs) on halt till maintenance phase. However, neglecting NFRs during prioritization phase may lead to inaccurate estimations for software projects resulting in high maintenance cost and failures. The subjective and uncertain nature of non-functional requirements makes them unfit to be prioritized using conventional prioritization methods. Although the existing literature reports on inadequate consideration given to NFRs prioritization, still no comprehensive systematic effort has been done to report the limitations and evaluation mechanisms of existing NFRs prioritization approaches. Requirements engineering society lacks a broad understanding of NFRs prioritization approaches and the challenges which need to be overcome. Therefore, we aim to investigate (i) the existing NFR prioritization techniques and their validation mechanisms, (ii) the role of Artificial Intelligence (AI) in NFRs prioritization, and (iii) the limitations of existing NFRs prioritization techniques. For this, we reviewed the literature published from 2008 till present and extracted 30 studies. The results reveal twenty-five NFRs prioritization techniques out of which only three are AI based. The major limitations we have come across are that most of the NFRs prioritization techniques are not scalable to large datasets, inter-dependencies between functional requirements (FRs) and NFRs are ignored, and the uncertainties associated with NFRs are not considered at all. However, the literature suggests that AI-based techniques and Fuzzy logic may be used to solve issues such as uncertainties i.e. ambiguities, vagueness, and subjective opinions of stakeholders. This review adds to the existing body of knowledge on NFRs and motivates the practitioners to focus on the NFR prioritization by highlighting the limitations of the existing methods.**

Keywords— Non-functional requirements; quality attributes; prioritization; quality requirements; systematic literature review.

I. Introduction

Non-functional requirements (NFRs) are often neglected during the software development process [1]. Neglecting NFRs during requirements prioritization phase may lead to inaccurate estimations for software projects resulting in high maintenance cost and even failures. Literature has just a handful of approaches that consider both functional and non-functional requirements for prioritization i.e. such as Analytic Hierarchy Process (AHP), Hybrid Assessment Method (HAM) and Integrated Prioritisation Approach (IPA) [2]. Among these approaches, AHP [3] is the most commonly used approach for NFRs prioritization. Although the success of a software system demands equal consideration given to NFRs along with FRs, the current literature clearly portrays the insufficient attention given to NFRs prioritization. A few NFRs prioritization approaches have been proposed by researchers but there is a lack of work on reporting the limitations and challenges posed by these approaches. Requirements engineering community demands a comprehensive review of existing literature on NFRs prioritization approaches and their limitations. Also, very little is known on the broader aspects like exploring the role of Artificial Intelligence in NFRs prioritization. Neglecting NFRs is unfortunate, yet common in the industry [4][5].

A survey conducted with Australian organizations on exploring the management practices of quality requirements showed that only one of six organizations paid attention to quality attributes during the prioritization phase [4]. However, another study focused on identifying the quality requirements prioritization practices followed in eleven successful software development companies revealed that most common way of prioritizing quality requirements is using ad hoc processes i.e. numerical assignment technique [5]. There are a certain number of existing techniques in literature [6] used for NFR prioritization e.g. prioritization through goal-decomposition [7], prioritization through architecture feedback [8],[9] and NFRs prioritization algorithm [10]. Still, it is uncommon for the industry practitioners to prioritize NFRs as they prioritize FRs.

The literature presents only one systematic literature review [11] with the purpose of assessing the current state of the art on NFRs prioritization. The main limitation of this study is that it relies on a relatively small number of studies (only 9 studies on NFRs prioritization). Another issue is with the limited number of databases searched for the identification of potential studies. Search Query has been deployed on 3

978-1-7281-3421-5/19/$31.00 ©2019 IEEE
DOI 10.1109/SEAA.2019.00064

databases only. Apart from this, the research methods have not been described in detail. Studies selection procedure seems uncovered. Another major observation is that its major focus is on reviewing empirical studies on NFRs prioritization and does not cover the evaluation details of existing NFRs prioritization approaches, their limitations. Moreover, this study does not cover all of the studies published on the topic after 2017. However, there is a pressing need to further explore the literature in the area of NFRs prioritization. Thus our study makes a first attempt towards conducting a comprehensive systematic literature review for NFRs prioritization techniques. We aim to cover the information about existing NFRs prioritization techniques, their limitations, evaluation mechanisms used and to explore the role of AI in overcoming the challenges of existing approaches. To the best of our knowledge, no such work has been done before.

Rest of the paper is organized as follows: Section 2 presents the research approach comprising of the review plan, research strategy, search criteria, research objectives, and questions. Section 3 presents the results and findings of this study. Section 4 involves the discussion of results. In Section 5 threats to validity have been discussed. Section 6 concludes the work and in section 7 study limitations and future work has been discussed.

II. RESEARCH METHOD

This systematic literature review has been conducted by following the guidelines of Kitchenham and Charters [12]. As a sample guideline [13] has been followed.

A. Plan Review Phase

First of all, objectives were identified and their corresponding questions were formulated. Then search strategy was designed and criteria for research were defined. Then according to the search strategy and search criteria this research was conducted.

1) Research Objectives and Research Questions:

The objectives of this systematic literature review are:

1. To identify existing techniques and approaches for prioritizing non-functional requirements.
2. To investigate the role of Artificial Intelligence in NFRs prioritization.
3. To explore the evaluation mechanisms of existing NFRs prioritization approaches.
4. To identify the limitations of existing NFRs prioritization approaches.

To answer these research objectives following research questions have been formulated:

RQ1. Which approaches are available for prioritizing non-functional requirements according to published studies?
RQ2. What role has been played by Artificial Intelligence in NFRs prioritization?
RQ3. Which of the existing NFRs prioritization approaches have been validated and how?
RQ4. What are the shortcomings of existing prioritization approaches for NFRs?

2) Search Strategy and Search Criteria:

Guidelines of Kitchenham and Charters [12] were followed in order to conduct this systematic literature review. After formulating research objectives and research questions next step was the formation of search strategy for searching research papers from 4 electronic databases: IEEE, ACM, Springer and Science Direct. After the formation of search criteria, search keywords were extracted from research questions and then search query was formulated. Keywords were divided into two sets. Table 1 summarizes the search strategy and search criteria.

3) Inclusion and Exclusion Criteria:

After inserting search query in search databases, Inclusion and exclusion criteria were applied to search results.

a) Inclusion criteria:

- All studies written in English were accepted.
- Studies published during the years 2008-present were considered in order to cover the maximum number of studies published in recent times and also to limit the scope of our work.
- All the studies related to search keywords were considered.
- Studies specifically focusing on NFRs prioritization were included.
- The studies extracted using the search query and snowballing technique were considered only.

b) Exclusion criteria:

- After reading the title and abstract, all irrelevant studies were excluded.
- Studies which were not majorly focused on NFRs prioritization were excluded. These studies might also include the ones that talk about prioritization of NFRs as sub topic i.e. [14].
- All books in search results were excluded.
- Studies written in languages other than English were excluded.
- Studies that were not related to requirement engineering domain were excluded.

TABLE 1. SEARCH STRATEGY & SEARCH CRITERIA

Electronic Databases	IEEE, ACM, Springer and Science Direct
Publication year	2008-Present
Language	English
Query (Deployed on ACM, IEEE and SPRINGER)	((requirements AND (non-functional OR extra-functional OR quality)) OR NFRs) AND (prioritization)

B. Conducting the Review

In order to conduct this systematic literature review, two steps were performed: Selection of results from four databases and data extraction.

1) Study Selection:

The multistep process was followed for study selection. First, the search queries were directly deployed and an initial set of studies was retrieved, followed by applying exclusion/inclusion criteria in the first round by inspecting title and abstract, then whole text was read to select the final pool in the second round. This completed the first iteration of studies retrieval. After deploying the query, Step-by-step study retrieval in iteration 1 is shown in Table 2.

Iteration 2 involved the retrieval of studies through snowballing. The purpose was not to miss out any relevant study that might not be retrieved directly by inserting a search query in electronic databases. We retrieved a total of eight studies. Four studies ([10], [15], [16], [17]) were retrieved through backward snowballing and other four ([6], [18], [19], [20]) through forward snowballing process.

TABLE 2. RESULT COUNT BEFORE AND AFTER APPLYING INCLUSION AND EXCLUSION CRITERIA (ITERATION 1)

Database	Direct Search Hits	Round1 (Title and abstract)	Round2 (Full-text read)
IEEE	199	34	14
ACM	127	7	2
Springer	1124	22	5
Science Direct	144	14	1
Total			**22**

2) Data Extraction and Analysis:

From the final set of studies containing 30 published research papers, the relevant information was extracted that was sufficient to answer research questions. For example, the main focus of research questions was to identify prioritization techniques for Non-functional requirements, limitations of those techniques, their validation, and role of Artificial Intelligence in NFRs prioritization. Information extracted from the research papers includes the year of publication, author name, conference/ journal/ workshop name, references, proposed work, methodology, findings, evaluation mechanism, conclusion, future work, and limitations. Information from each of these sections was then documented separately in a word file in the form of summary. The summaries helped in finding relevant information for answering research questions easily.

3) Quality Assessment Criteria:

In order to evaluate the quality of studies, few quality criteria questions from [12] were selected. Table 3 shows the results of the quality assessment of selected studies.

TABLE 3. QUALITY ASSESSMENT CRITERIA

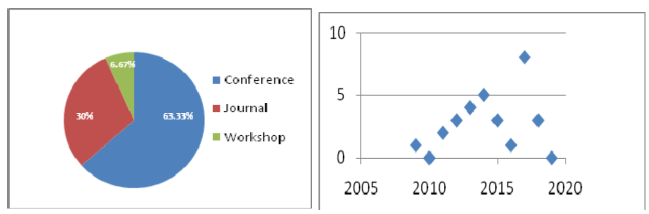
Criteria	Answers	Overall response of studies
Is the research problem clearly mentioned in the studies?	None Poor Good N 1 2 3 4 5	Yes, around 80% of studies clearly mentioned the research problem.
Does the study clearly state the aims of the research?	None Poor Good N 1 2 3 4 5	Yes, almost all studies clearly defined the aim of the research.

III. FINDINGS OF REVIEW

In the following sections, results of the systematic literature review are presented, including the answers of research questions.

A. Overview of studies

30 studies were reviewed which were relevant to the research objectives of this study and it was found that 63.33 % of those studies were published in conference, 30% were published in journals and only 6.67% of them were published in the workshop as shown by Fig. 1a. Table 4 provides brief details of these studies. Moreover, most of the results were obtained from IEEE. After reviewing each subsequent study it was found that 83.33% of the studies presented some approaches for non-functional requirement prioritization, and remaining 16.67% focused on exploring the state of the art on NFRs prioritization. It is also observed that most of the work on NFRs prioritization has been done in recent years i.e. the majority of studies have been published since 2011 as shown in Fig. 1b.



(a) **(b)**

Fig. 1. (a) Publication type-wise distribution of studies **(b)** Year-wise distribution of studies

TABLE 4. DETAILS OF REVIEWED STUDIES

Ref	Publication Name	Publication type	No
[21], [18], [22], [23]	IEEE SOUTHEAST CONFERENCE	Conference	4
[24]	ACIS	Conference	1
[10]	Software Quality Journal	Journal	1
[25]	Requirements Engineering	Journal	1
[26]	International Conference on Information Science Applications (ICISA)	conference	1
[3]	Innovations in Systems and Software Engineering	Journal	1
[5], [27]	Requirements Engineering Conference (RE)	conference	2
[28]	International Colloquium in Information Science and Technology (CIST)	conference	1
[29]	International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS)	conference	1
[15]	International Journal of Software Engineering and its Applications	Journal	1
[30]	Control and System Graduate Research Colloquium (ICSGRC)	conference	1
[16]	Lecture notes on software engineering	Journal	1
[31]	Third International Workshop on software product management	Workshop	1
[19]	International Journal of Applied Information Systems (IJAIS)	Journal	1
[17]	The Scientific World Journal	Journal	1
[20]	Proceedings of the Second International Conference on Computational Intelligence and Informatics	Conference	1
[6]	International Journal of Computer Science & Business Information	Journal	1
[32]	International Conference on Information Management and Engineering	Conference	1
[4]	Euromicro Conference on Software Engineering and Advanced Applications	Conference	1
[33]	International Conference on Cloud Computing, Data Science Engineering - Confluence	Conference	1
[9]	International Workshop on the Twin Peaks of Requirements and Architecture (TwinPeaks)	Workshop	1

[34]	International Conference on Research Challenges in Information Science (RCIS)	Conference	1
[35]	Perspectives in Science Journal	Journal	1
[8], [36]	Requirements Engineering: Foundation for Software Quality	Conference	2
[37]	International Conference on Hybrid Information Technology	Conference	1

B. RQ1: Which approaches are available for prioritizing non-functional requirements in literature?

To answer RQ1, we found the studies which present approaches for prioritizing non-functional requirements. Majority of these studies incorporate Analytic Hierarchy Process (AHP) in NFRs prioritization. The studies have been presented here according to the main themes that we have identified.

1) AHP based NFRs Prioritization Approaches

Literature review shows that AHP is the most commonly used method for requirements prioritization including NFRs. An automated approach called CEPP [21] for capturing, eliciting, predicting and prioritizing NFRs is proposed which is the extended version of Capture Elicit Prioritize methodology (CEP) [18]. Both CEP and CEPP aim to extract NFRs from documents and images and to prioritize them using the αβγ framework. The αβγ framework has also been used in the study [22] which is inspired by the study [18] discussed earlier.

A prioritization approach leveraging AHP [3] focusing on the NFR's inter-relationships has been presented in [28]. The proposed approach first identifies the inter-relationships between the requirements and then prioritizes them using AHP. Another variant of AHP proposed in [24] applies AHP to system quality attributes in order to prioritize them based on their importance. This approach removes the quality attributes conflicting with the ones with high priority values. A similar elimination approach taking care of consistencies among requirements has been presented in [30]. This research [30] is basically inspired by [24] and uses impact score for eliminating the conflicting quality attributes. AHP has also been used by Garg and Singhal in their study [33] to combine it with two other prioritization approaches i.e. cost-value approach and numerical assignment. An extended version of AHP namely Primitive Cognitive Network Process (P-CNP) has been used in another study [27]. This study proposes a tool based approach namely Automatic Runtime Reappraisal of Weights (ARROW) which helps in assigning and updating weights of NFRs based on different environmental contexts. ARROW detects changes in requirements using Dynamic decision networks (DNN) and re-adjusts the weights of NFRs at runtime. DNN also keeps track of history and helps in performing tradeoffs among NFRs.

AHP has also been integrated with NFRs framework [10] to present a combined approach [34] that is capable of managing the tradeoffs between NFRs. NFRs framework presented in [10] is another approach to prioritize system Quality Attributes in order to achieve project and business objectives. Other NFRs prioritization proposals include the hybrid approach that prioritizes FRs based on NFRs [33]. The main purpose of study [33] is to present an approach that is the combination of some existing prioritization approaches i.e. cost-value approach, numerical assignment, and AHP.

2) Approaches for Simultaneous Prioritization of FRs and NFRs

Hybrid Assessment Method (HAM) presented in [26] defines criteria for prioritization and performs pair-wise comparisons of NFRs as used in [33] for defining tradeoffs. HAM based approaches prioritize requirements by considering NFRs as criteria and FRs as alternatives. Hence, HAM based approaches simultaneously prioritize both FRs and NFRs. Requirements prioritization through Tensor decomposition presented in [25] involves simultaneous prioritization of NFRs and FRs. NFRs are prioritized along with the FRs by keeping into consideration their relationship with the stakeholders. This approach makes use of multi linear algebra. Similar to this, some other simultaneous and separate prioritization approaches for NFRs and FRs have been discussed and validated in [35]. This study finds the accuracy and complexity of three NFRs prioritization approaches by using different sized projects comprising 13, 34, and 56 requirements. The results show that the approach that prioritizes NFRs by considering their relationship with FRs outperforms the rest. Another NFRs prioritization approach is NFR planning method for agile processes (NORPLAN) presented in [23]. NORPLAN is a part of the Non-functional requirements model for Agile process (NORMAP) methodology discussed in [18]. NORPLAN is basically concerned with developing a framework for agile processes that include both FRs and NFRs. It uses measures for quality and project management and risk-driven algorithm for prioritizing quality attributes. NORPLAN is essentially used to compute threats associated with non-functional requirements prioritization. The study [23] also suggests two other risk-based approaches known as riskiest requirements first and riskiest requirement last.

3) Architecture based NFRs prioritization Approaches

A few architecture based approaches for NFRs prioritization have also been introduced in literature i.e. Prioritization through architecture feedback [8][9]. This approach first identifies the NFRs and then ranks them as per stakeholders view. After prioritization, an architecture model is created, then evaluated, design space is explored using a tool named Per-Opteryx, and tradeoffs are analyzed. After this, the stakeholders again prioritize NFRs according to the required quality levels and update the proposed architecture. The proposed approach is effective in prioritizing architecturally significant NFRs as the software architects can analyze the tradeoffs and conflicts among quality attributes by exploring design space thus prioritizing quality attributes based on architecture evaluation feedback. The approach does not handle requirements uncertainties such as ambiguities, vagueness, and incompleteness. However, it is claimed that this approach has handled the "conflict" type of requirements inter-dependency.

4) Approaches for Prioritising NFRs based on Historically Similar Projects

A few NFRs prioritization approaches use historical information from similar project requirements for NFR prioritization. For instance, in [29] priority values are assigned to NFRs based on the information retrieved from already existing projects. This approach performs better in terms of accuracy and time consumption as compared to other approaches using AHP (e.g. [15]).

5) Quality Goals Prioritization Approach

Another approach that involves quality goals elicitation and prioritization is presented in [31]. First of all the quality goals are gathered through brainstorming sessions and then are prioritized by asking the stakeholders to cast votes to a particular quality goal based on its importance value.

6) Security Requirements Prioritization Approaches

A security requirements prioritization approach uses misuse cases to prioritize security requirements [32]. First of all, the use cases and misuse cases are identified and a link is established among them. Next step is to calculate the risk associated with misuse cases. After this, security requirements are identified and then prioritized by calculating total mitigated risk (TMR) value. Priorities to security requirements are assigned based on the TMR value. A similar approach for security requirements prioritization has been suggested in [37]. The purpose is to prioritize security requirements by analyzing security threats, assets and the possible mitigation mechanisms that are modeled in the form of a graph. A major limitation of these approaches is that their scope is only limited to security requirements and cannot be generalized to other non-functional requirements.

7) Other NFRs Prioritization Approaches

Another study focuses on quality requirements prioritization [36]. The idea is to incorporate green strategies into software engineering phases in order to ensure minimum resource utilization. The prioritization here is performed to identify which quality requirements are most important for addressing the incorporated green strategies.

Above mentioned NFRs prioritization techniques and approaches have been identified from 30 research papers. On the whole, we found 25 NFRs prioritization techniques and approaches, out of which 22 have been explained above. Other 3 are Artificial Intelligence based and are explained in RQ2.

C. *RQ2: What role has been played by Artificial Intelligence in non-functional requirement prioritization?*

Our motivation behind discovering the role of Artificial Intelligence (AI) NFRs prioritization is to identify the strengths and capabilities of these approaches in overcoming the challenges posed by conventional prioritization approaches.

The literature reports on the challenges faced by requirement engineers during the prioritization of NFRs i.e. uncertainties present in requirements [38]. Similar types of uncertainties are also present in NFRs. A major form of uncertainty in NFRs prioritization is doubtful preference values assigned by stakeholders. As NFRs are typically ignored in earlier stages of the software development life cycle, the decision makers tend to have limited knowledge available about the requirements and their importance [39]. Thus they are not sure about what preference value should be assigned to particular NFR with respect to its relationship or dependency with FR. As a lot of stakeholders are involved in prioritization and their diverse opinions should be taken into account. This also leads to uncertainties which are not usually handled by existing non-functional requirements prioritization techniques [40]. Ambiguity, incompleteness, and vagueness are also major forms of uncertainties encountered during requirements prioritization [38]. To effectively deal with such uncertainties in non-functional requirements prioritization there is a need to develop a solution. In this regard, Artificial Intelligence can play a vital role. Fuzzy logic is an Artificially Intelligent approach that can serve the purpose as it is able to handle such sort of uncertainties effectively [41]. Out of the selected sample of 30 studies, only four studies were found that incorporated Artificial Intelligence in NFRs prioritization process. Brief details of AI-based approaches for non-functional requirements prioritization are given below.

Integrated prioritization approach (IPA) presented in [16], [17] is a quantitative approach that uses alpha cut and fuzzy logic to produce a list of prioritized NFRs. During prioritization, the relationships of NFRs with corresponding functional requirements are kept into consideration. Although IPA considers simultaneous prioritization of functional and non-functional requirements, it has been evaluated with only 20 requirements of Simulated Banking Software comprising of 15 FRs and 5 NFRs. Next, Neuro-Fuzzy based approach has been discussed in [19]. The approach works by first identifying software quality requirements and then assigning ranks to them using binary search tree algorithm. The biggest strength of this approach is that it considers the positive and negative relationships between quality attributes while prioritizing them. However, its evaluation is performed using only 29 requirements of Sale System and Financial System. Next, an approach [20] makes use of AHP and its fuzzified form to prioritize non-functional requirements using a cognitive decision support system. The main advantage provided by this approach is that it considers both types of requirements (functional and non-functional) for prioritization. However, nothing can be said about its scalability as its evaluation has been performed with only 11 requirements of the Online Food Ordering System.

It is considerable to note that the main drawback of all of the above mentioned AI-based approaches is that their scalability has not been evaluated. The validation has been performed with a very small number of requirements which raises questions about their strengths.

D. RQ3. Which of the existing NFRs prioritization approaches have been validated and how?

We found that some existing NFRs prioritization approaches lack validation for instance ([8][9][21][22][24][33]). Rest of them are summarized in table 5.

TABLE 5. VALIDATION DETAILS OF EXISTING NFRS PRIORITIZATION APPROACHES

Ref	Approach	Evaluation
[18]	Capturing, Eliciting, and Prioritizing (CEP)	This approach has been validated using 26 requirements documents from European Union eProcument.
[10]	NFR prioritization algorithm	Data from a real software project namely "Exam assessment system" has been used to evaluate the study. Just seven NFRs have been considered for evaluation.
[26]	Hybrid Assessment Method (HAM) based approach	The approach has been evaluated by means of the experiment using 20 requirements of simulated banking software including 15 FRs and 5 NFRs.
[3]	Analytic Hierarchy Process (AHP)	The approach has been illustrated in the context of Remote monitoring system for medical patients via considering only 3 quality attributes.
[27]	Automatic Runtime Reappraisal of Weights (ARRoW)	Arrow approach has been demonstrated using the case study of remote data monitoring (RDM).
[23]	Non-functional Requirements Planning for Agile Processes (NORPLAN)	Validation has been done through visual simulation and using eProcument as a case study including 42 requirements having 26 FRs.
[29]	Prioritization Based on Historical Similar Project	A comparative analysis is performed and the proposed approach is compared with existing NFRs prioritization technique [15] in terms of computation time and accuracy. The analysis is performed on a Software project of tool for automatic analysis and comparison of different release planning methods containing 24 FRs and 6 NFRs in 6th increment.
[15]	Prioritization based on dependencies and usage count	Effectiveness of proposed work is illustrated by applying the technique on a case study of software project of tool for automatic analysis and comparison of different release planning methods containing 24 FRs and 6 NFRs in 6th increment.
[30]	Conflict-free quality attributes achieving approach	The approach has been evaluated using an online purchasing system as a case study. Only 5 NFRs have been considered for evaluation.
[31]	Lightweight elicitation and analysis approach	The proposed method has been applied and improved in 4 cases in 4 different companies having different contexts.
[32]	Enhanced misuse cases based approach	Effectiveness of the proposed approach is demonstrated by applying it on E-commerce web application case study.
[25]	Requirements prioritization based on Tensor decomposition	A controlled experiment was conducted for comparing the proposed approach with AHP in terms of accuracy, ease of use and actual time consumption. 15 real requirements containing 10 FRs and 5 NFRs of the online banking system have been used.
[34]	AHP integrated with NFRs framework	No empirical evaluation has been performed. However, for demonstration purpose, credit card system case study has been used.
[37]	Threat modeling and tree-structured Graph approach	Evaluation is done on a real system that is E-commerce web application.
[36]	Prioritizing NFRs by incorporating Green strategies	Effectiveness has been demonstrated by conducting an empirical study by involving 19 teams of students who designed the software system. The EV-Mobility project has been used as a case study.
[16], [17]	Integrated prioritization approach (IPA)	Evaluation is performed using 20 requirements of simulated banking software including 15 FRs and 5 NFRs.
[19]	Neuro-Fuzzy based approach	The evaluation has been performed using Sale system and financial system as a case study containing 13 FRs 16 NFRs
[20]	Cognitive Decision Support System	The approach has been evaluated by conducting an experiment on the mobile based application as a case study.

RQ4: What are the shortcomings of existing prioritization approaches for NFRs?

Most common limitation of the NFRs prioritization approaches reviewed is the small number of requirement used for evaluation purpose for instance ([3],[10],[16],[17],[18],[19],[20],[23][25],[26],[27],[29]). Since, most of the approaches (e.g. [15] and [34]) use AHP for prioritization which is suitable only for small number of requirements i.e. 20. Furthermore, some of the approaches (e.g. [34]) are not empirically evaluated and don't consider any interdependencies among requirements.

Another serious issue with the published work so far is the lack of validation of the proposed approaches (i.e. [8],[9],[21],[24],[33]). For instance, NORPLAN needs to be validated with real-world agile requirements planning groups [23] and the αβγ framework [18],[21],[22] needs to be tested with variable number of requirements.

The scope of architecture based NFRs prioritization approaches [8], [9] is limited only to quantitatively evaluated NFRs only. However, some of the proposed approaches like [32],[37] are only applicable to security requirements. The limitation associated with Conflict-free quality attributes achieving approach presented in [30] is that there are high chances of correlation impact score to be incorrect. An important limitation worth considering in the lightweight elicitation analysis approach presented in [31] is that it is difficult to say that quality goals prioritized using this approach are assigned with correct priorities. Similarly, the NFRs prioritization algorithm presented in [10] performs computations by taking inputs from the stakeholders that can lead to conflicts thus effecting the decision making. In addition to this, the limitations associated with some of the studies (e.g. [36]) with students as participants involve bias because of the involvement of teams of students.

IV. DISCUSSION OF RESULTS

Prioritizing NFRs is an equally important activity as of FRs that must be performed by the requirement engineers in the early stages of the software development life cycle. This systematic literature review identified the existing approaches for NFRs prioritization. We supported four main research questions and successfully identified existing NFRs prioritization approaches, their validation mechanisms, and their limitations. Not only this, but we also discovered the role of Artificial Intelligence in NFRs prioritization. We identified various themes and accordingly classified the extracted studies. We found that various NFRs prioritization techniques use the Analytic Hierarchy Process (AHP). However, AHP has its own limitations and still demands subjective judgment [11]. Since there are important factors such as inter-dependencies among requirements and uncertainties involved in NFRs which may not be handled seamlessly by subjective analysis based techniques like AHP. Unfortunately, literature reports just a handful of studies in which prioritization of

NFRs is performed by considering the FR-NFR interdependencies.

We also found some approaches which focus on simultaneous prioritization of FRs and NFRs. Moreover, some approaches have specifically been developed for security requirements prioritization. A few architecture based approaches have also been discovered through this review. However, these proposed approaches have several limitations. Most common of which is the lack of validation with real-world data. This leads to a crippling behavior of the suggested approaches for NFRs prioritization towards a larger number of requirements. Literature shows (e.g.[2], [42]) that the proposed techniques for NFR prioritization are not guaranteed to be flexible to deal with volatile requirements. For instance, in the case of AHP based approaches where pair-wise comparisons are performed and change in one requirement can possibly cause a ripple effect that might require all comparisons to be computed again.

Above all, the literature reports on various challenges related to requirements uncertainties faced by requirement engineers during the prioritization of NFRs i.e. inherent uncertainties, handling dependencies among functional and non-functional requirements and dealing with diverse opinions of stakeholders related to priority values [38]. Modern AI-based techniques i.e. Fuzzy Logic can help to solve the issues related to uncertainties with much ease. This is because Fuzzy based techniques allow most of the uncertainties to be modeled very easily [43]. According to a study [41], fuzzy logic can be used to handle various kinds of uncertainties i.e. ambiguities, imprecision, vagueness and subjective opinions of stakeholders. Some AI and fuzzy logic based approaches such as Interactive Genetic Algorithm [44] and Fuzzy Analytic Hierarchy Process (AHP) [45] have been developed for prioritizing requirements. Both of the techniques are capable of dealing with uncertainties in requirements i.e. Subjectivity, vagueness, and imprecise judgments. Moreover, these techniques successfully overcome the limitations of AHP regarding the applicability to large scale software projects. Nevertheless, the applicability of these approaches for NFRs prioritization has yet to be explored.

Therefore, we can infer that use of fuzzy and AI can play a positive role in NFR prioritization and can uproot the limitations of existing techniques including a number of requirements, uncertainties, dependencies and subjectivity of stakeholders' opinion. This systematic literature review finds only four studies that used the concepts of AI for NFRs prioritization. This opens up new venues for the researchers and practitioners to explore the use of AI-based techniques to deal with the uncertain nature of requirements.

V. THREATS TO VALIDITY

Major threats to the validity of this study include the limited inclusion and exclusion criteria and limited keywords used in query design. The search query was formulated keeping in view the research questions of the study. Hence, there is a chance of missing out some of the relevant. However, to mitigate this threat, we used forward and backward snowballing to not to miss out on any relevant study that our query could have not identified earlier.

VI. CONCLUSION

Purpose of this systematic literature review was to discover the current state of the art on NFRs prioritization. 30 research studies from four electronic databases named IEEE, ACM, Springer and Science direct were extracted. Review findings show that each of the existing NFRs prioritization approaches has some limitations. Most of the prioritization approaches cannot deal with a large number of requirements, requirements uncertainties, and inter-dependencies between FRs and NFRs. However, it is discovered that incorporating AI in requirement prioritization can solve these problems to a certain extent. No doubt, application of AI in NFR prioritization is an interesting area of research but currently, very few studies have been conducted in this dimension. Our review provides an overall view of the literature available on NFR prioritization. In addition, it highlights the limitations of the existing techniques and opens up avenues for researchers to conduct more experimental studies with a large number of requirements using AI-based approaches.

VII. LIMITATIONS & FUTURE WORK

One of the major limitations of this study can be the tenure we have selected for review. Although, we chose it carefully after skimming through the literature and finding it to be a fertile phase for research in the selected area still there are chances of getting some studies beyond this term. Another limitation can be our limited inclusion and exclusion criteria. For instance, we have chosen studies published in the English language only. We might have missed significant work done in this domain in other languages. The search engines we did not include in our search might also contain some work on the subject. In the future, we plan to extend this review by mitigating some of the limitations and also by expanding our research questions to prioritizing factors, requirements interdependencies and exploring methods other than AI for prioritization. We also aim to propose an AI-based technique to prioritize both functional and non-functional requirements keeping in view the inter-dependencies, uncertainties, and scalability in mind based on our learning from this review.

REFERENCES

[1] Q. L. Nguyen, "Non-functional requirements analysis modeling for software product lines," in *2009 ICSE Workshop on Modeling in Software Engineering*, 2009, pp. 56–61.

[2] V. Gupta, D. S. Chauhan, and K. Dutta, "Exploring prioritization through systematic literature surveys and case studies," *SpringerPlus*, vol. 4, Sep. 2015.

[3] M. Kassab and N. Kilicay-Ergin, "Applying analytical hierarchy process to system quality requirements prioritization," *Innov. Syst. Softw. Eng.*, vol. 11, no. 4, pp. 303–312, Dec. 2015.

[4] L. B. Phillips, A. Aurum, and R. B. Svensson, "Managing Software Quality Requirements," in *2012 38th Euromicro Conference on Software Engineering and Advanced Applications*, 2012, pp. 349–356.

[5] R. B. Svensson *et al.*, “Prioritization of quality requirements: State of practice in eleven companies,” in *2011 IEEE 19th International Requirements Engineering Conference*, 2011, pp. 69–78.
[6] S. Baskaran, “A Survey on Prioritization Methodologies to Prioritize Non Functional Requirements,” *Int. J. Comput. Sci. Bus. Inform.*, vol. 12, no. 1, May 2014.
[7] M. Daneva, M. Kassab, M. L. Ponisio, R. Wieringa, and O. Ormandjieva, “Exploiting a Goal-Decomposition Technique to Prioritize Non-functional Requirements,” in *WER*, 2007.
[8] A. Koziolek, “Research Preview: Prioritizing Quality Requirements Based on Software Architecture Evaluation Feedback,” in *Requirements Engineering: Foundation for Software Quality*, 2012, pp. 52–58.
[9] A. Koziolek, “Architecture-driven quality requirements prioritization,” in *2012 First IEEE International Workshop on the Twin Peaks of Requirements and Architecture (TwinPeaks)*, 2012, pp. 15–19.
[10] R. Thakurta, “A framework for prioritization of quality requirements for inclusion in a software project,” *Softw. Qual. J.*, vol. 21, no. 4, pp. 573–597, Dec. 2013.
[11] V. Gupta*, S. Lohia, D. Çetinkaya, and H. Kim, “Towards a Systematic Literature Review of Non-Functional Requirement Prioritization Approaches,” *ijsela*, vol. 11, no. 2, pp. 1–6, 2017.
[12] K. BA and S. Charters, “Guidelines for performing Systematic Literature Reviews in Software Engineering,” vol. 2, Jan. 2007.
[13] I. Inayat, S. S. Salim, S. Marczak, M. Daneva, and S. Shamshirband, “A Systematic Literature Review on Agile Requirements Engineering Practices and Challenges,” *Comput Hum Behav*, vol. 51, no. PB, pp. 915–929, Oct. 2015.
[14] N. Ali, A. Martínez-Martínez, L. Ayuso-Pérez, and A. Espinoza, “Self-adaptive Quality Requirement Elicitation Process for Legacy Systems: A Case Study in Healthcare,” in *Proceedings of the Symposium on Applied Computing*, New York, NY, USA, 2017, pp. 1102–1107.
[15] S. Lohia, “Non Functional Requirement Prioritization Approach Based on Dependencies & Usage Count,” *Int. J. Softw. Eng. Its Appl.*, Jan. 2017.
[16] M. Dabbagh and S. Peck Lee, “An Approach for Prioritizing NFRs According to Their Relationship with FRs,” *Lect. Notes Softw. Eng.*, vol. 3, pp. 1–5, Jan. 2015.
[17] M. Dabbagh and S. P. Lee, “An Approach for Integrating the Prioritization of Functional and Nonfunctional Requirements,” in *TheScientificWorldJournal*, 2014.
[18] R. R. Maiti and F. J. Mitropoulos, “Capturing, eliciting, and prioritizing (CEP) NFRs in agile software engineering,” in *SoutheastCon 2017*, 2017, pp. 1–7.
[19] H. Momeni, H. Motameni, and M. Larimi, “A Neuro-Fuzzy based Approach to Software Quality Requirements Prioritization,” *Int. J. Appl. Inf. Syst.*, vol. 7, pp. 15–20, Aug. 2014.
[20] P. Saratha and G. V. Uma, “Cognitive Decision Support System for the Prioritization of Functional and Non-functional Requirements of Mobile Applications,” in *Proceedings of the Second International Conference on Computational Intelligence and Informatics*, 2018, pp. 347–358.
[21] R. R. Maiti and F. J. Mitropoulos, “Capturing, eliciting, predicting and prioritizing (CEPP) non-functional requirements metadata during the early stages of agile software development,” in *SoutheastCon 2015*, 2015, pp. 1–8.
[22] R. R. Maiti and F. J. Mitropoulos, “Prioritizing Non-Functional Requirements in Agile Software Engineering,” in *Proceedings of the SouthEast Conference*, New York, NY, USA, 2017, pp. 212–214.
[23] W. M. Farid and F. J. Mitropoulos, “NORPLAN: Non-functional Requirements Planning for agile processes,” in *2013 Proceedings of IEEE Southeastcon*, 2013, pp. 1–8.
[24] M. Dabbagh and S. P. Lee, “A Consistent Approach for Prioritizing System Quality Attributes,” in *2013 14th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, 2013, pp. 317–322.
[25] N. Misaghian and H. Motameni, “An approach for requirements prioritization based on tensor decomposition,” *Requir. Eng.*, vol. 23, no. 2, pp. 169–188, Jun. 2018.
[26] M. Dabbagh, S. P. Lee, and R. M. Parizi, “Application of Hybrid Assessment Method for Priority Assessment of Functional and Non-Functional Requirements,” in *2014 International Conference on Information Science Applications (ICISA)*, 2014, pp. 1–4.
[27] L. H. G. Paucar and N. Bencomo, “ARRoW: Tool Support for Automatic Runtime Reappraisal of Weights,” in *2017 IEEE 25th International Requirements Engineering Conference (RE)*, 2017, pp. 458–461.
[28] F. Fellir, K. Nafil, and R. Touahni, “System requirements prioritization based on AHP,” in *2014 Third IEEE International Colloquium in Information Science and Technology (CIST)*, 2014, pp. 163–167.
[29] V. Gupta, M. T. Khan, and C. Gupta, “Non-functional requirement prioritization approach based on historical similar project,” in *2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS)*, 2017, pp. 396–399.
[30] M. A. A. Imran, S. P. Lee, and M. A. M. Ahsan, “Measuring impact factors to achieve conflict-free set of quality attributes,” in *2017 IEEE 8th Control and System Graduate Research Colloquium (ICSGRC)*, 2017, pp. 174–178.
[31] J. Vanhanen, M. V. Mäntylä, and J. Itkonen, “Lightweight Elicitation and Analysis of Software Product Quality Goals: A Multiple Industrial Case Study,” in *2009 Third International Workshop on Software Product Management*, 2009, pp. 42–52.
[32] S. G. Yoo, H. P. Vaca, and J. Kim, “Enhanced Misuse Cases for Prioritization of Security Requirements,” in *Proceedings of the 9th International Conference on Information Management and Engineering*, New York, NY, USA, 2017, pp. 1–10.
[33] U. Garg and A. Singhal, “Software requirement prioritization based on non-functional requirements,” in *2017 7th International Conference on Cloud Computing, Data Science Engineering - Confluence*, 2017, pp. 793–797.
[34] M. Kassab, “An integrated approach of AHP and NFRs framework,” in *IEEE 7th International Conference on Research Challenges in Information Science (RCIS)*, 2013, pp. 1–8.
[35] R. K. Chopra, V. Gupta, and D. S. Chauhan, “Experimentation on accuracy of non functional requirement prioritization approaches for different complexity projects,” *Perspect. Sci.*, vol. 8, no. Supplement C, pp. 79–82, Sep. 2016.
[36] N. Condori Fernandez and P. Lago, “The Influence of Green Strategies Design onto Quality Requirements Prioritization,” in *Requirements Engineering: Foundation for Software Quality*, 2018, pp. 189–205.
[37] K.-Y. Park, S.-G. Yoo, and J. Kim, “Security Requirements Prioritization Based on Threat Modeling and Valuation Graph,” in *Convergence and Hybrid Information Technology*, 2011, pp. 142–152.
[38] V. Persis, “Questions Related to Uncertainty of Requirements Prioritization,” Dec. 2015.
[39] V. Persis and A. Babu, “Requirements uncertainty prioritization approach: A novel approach for requirements prioritization,” *Softw Eng Int J SEIJ*, vol. 2, pp. 37–49, Jan. 2012.
[40] P. Bajaj and V. Arora, “Multi-person Decision-making for Requirements Prioritization Using Fuzzy AHP,” *SIGSOFT Softw Eng Notes*, vol. 38, no. 5, pp. 1–6, Aug. 2013.
[41] Ruby and Dr. Balkishan, “Role of Fuzzy Logic in Requirement Prioritization,” *Int. J. Innov. Res. Sci. Eng. Technol.*, vol. 4, no. 6, Jun. 2015.
[42] P. Berander, “Evolving Prioritization for Software Product Management,” *DIVA*, 2007.
[43] D. Wu, “A Brief Introduction to Type2 Fuzzy Logic,” Aug. 2018.
[44] P. Tonella, A. Susi, and F. Palma, “Using Interactive GA for Requirements Prioritization,” in *2nd International Symposium on Search Based Software Engineering*, 2010, pp. 57–66.
[45] P. Bajaj and V. Arora, “Multi-person Decision-making for Requirements Prioritization Using Fuzzy AHP,” *SIGSOFT Softw Eng Notes*, vol. 38, no. 5, pp. 1–6, Aug. 2013.