CHALMERS
UNIVERSITY OF TECHNOLOGY

Non-Functional Requirements for Machine Learning: An Exploration of System Scope and Interest

Non-Functional Requirements for Machine Learning: An Exploration of System Scope and Interest

Khan Mohammad Habibullah, Gregory Gay, Jennifer Horkoff
{khan.mohammad.habibullah,jennifer.horkoff}@gu.se,greg@greggay.com
Chalmers | University of Gothenburg
Gothenburg, Sweden

ABSTRACT

Systems that rely on Machine Learning (ML systems) have differing demands on quality—non-functional requirements (NFRs)—compared to traditional systems. NFRs for ML systems may differ in their definition, scope, and importance. Despite the importance of NFRs for ML systems, our understanding of their definitions and scope—and of the extent of existing research—is lacking compared to our understanding in traditional domains.

Building on an investigation into importance and treatment of ML system NFRs in industry, we make three contributions towards narrowing this gap: (1) we present clusters of ML system NFRs based on shared characteristics, (2) we use Scopus search results—as well as inter-coder reliability on a sample of NFRs—to estimate the number of relevant studies on a subset of the NFRs, and (3), we use our initial reading of titles and abstracts in each sample to define the scope of NFRs over parts of the system (e.g., training data, ML model). These initial findings form the groundwork for future research in this emerging domain.

CCS CONCEPTS

• **Software and its engineering** → **Extra-functional properties**; **Requirements analysis**; • **Computing methodologies** → *Machine learning*.

KEYWORDS

Non-Functional Requirements, Machine Learning, Machine Learning Systems, Requirements Engineering

ACM Reference Format:
Khan Mohammad Habibullah, Gregory Gay, Jennifer Horkoff. 2022. Non-Functional Requirements for Machine Learning: An Exploration of System Scope and Interest. In *Workshop on Software Engineering for Responsible AI (SE4RAI'22), May 19, 2022, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3526073.3527589

1 INTRODUCTION

Machine Learning (ML) is increasingly being used in decision making and prediction applications that influence many aspects of our lives. Complex systems, referred to as ML systems, use ML to deliver critical functionality. Such systems demand high computational capabilities (often based on GPUs), process a large amount of data, and utilize complex non-deterministic algorithms [9]. Therefore, ensuring the quality of such systems is potentially more expensive and effort-intensive than traditional systems.

A thorough requirements engineering (RE) process is necessary to ensure quality. Requirements imposed on system quality are known as non-functional requirements (NFRs), and are expressed over different attributes of quality [9]. For a ML system, one might imagine constraints over attributes such as fairness, transparency, privacy, security, or safety [8].

Although significant research has been devoted to NFRs, significant challenges remain for modern system development. Even for traditional systems, NFRs are difficult and challenging to express explicitly [6], and even more difficult to verify or validate [17]. Such challenges compound for ML systems, and the identification, definition, and measurement of NFRs for ML systems has emerged as a critical problem to solve [8, 9].

Much of our accumulated knowledge concerning NFRs may be no longer relevant when dealing with ML systems, due to their complex and non-deterministic nature. Some NFRs, such as fairness, explainability, and privacy become more important. Others, such as usability or interoperability, may become less important [8, 11, 18]. New NFRs, such as retrainability, emerge. Moreover, the meaning and interpretation of NFRs for ML systems may differ from traditional systems and may not yet be well understood [1]. To date, there has been little research on NFRs for ML systems [9].

Further, "ML" is not one monolithic entity, but can be considered at different levels of granularity within a larger system [19]. When imposing NFRs over an ML system, some NFRs may apply to the algorithm that performs the learning task, while others may apply over the training data used as the basis for such decision making or over the model trained using that data. Still others may apply over the results of applying that model, or over the broader ML system that acts on those results. Therefore, the scope of consideration for NFRs (i.e., the scope of identification, definition, and measurement) for ML systems is a complex and not-yet-solved problem.

We recently conducted an interview study, which examined treatment of NFRs for ML systems in industry and reported challenges of identifying, defining and measuring NFRs [8]. Addressing these challenges will require (1) **a detailed understanding of the definition and scope** of each NFR in a ML system context, and (2), **an examination of past research** on each of these NFRs as applied to ML system development. These needs are intertwined. To date, there has been no systematic literature reviews or other secondary studies on ML system NFRs. However, performing such a study requires a clear answer to questions of scope to proceed effectively.

SE4RAI'22 , May 19, 2022, Pittsburgh, PA, USA

ACM ISBN 978-1-4503-9319-5/22/05…$15.00
https://doi.org/10.1145/3526073.3527589

We perform an exploratory study of the treatment of certain NFRs for ML systems in research literature[1]. Our goals in this study are to (1) gain an approximate idea of the extent to which select NFRs have been studied by researchers, and (2), perform an initial clarification of the scope of these NFRs for ML systems.

As a starting point, we have taken into account the NFRs identified as important in the interview study [8]. Using this set of NFRs, we divide NFRs into clusters based on shared attributes of their definitions. This enables understanding of which NFRs could be considered in conjunction. Researchers could study particular clusters, and practitioners may consider defining system quality over related NFRs. We also identify an upper limit on the number of relevant publications in the Scopus database for each NFR. Our initial estimation shows that some NFRs, such as security or transparency, have received significant focus. We select NFRs that have received less attention (e.g., maintainability or testability), and examine the titles and abstracts of 50-100 publications for each. Based on this sample, we estimate the number of relevant publications on each of the selected NFRs. This estimation enables scoping of secondary studies. Finally, based on inspection of the titles and abstracts of these samples, we perform an exploratory scoping of the selected NFRs in terms of which elements of the system they can be defined over (e.g., training data, ML algorithm, or ML model). This scoping brings further clarity to the specific definitions and treatment of these NFRs, which can benefit future research and practice on each.

Our study, while exploratory in nature, is intended to open new opportunities for future research in NFRs for ML systems. We hope to set the groundwork for future studies by clarifying the scoping and definitions for these NFRs, identifying connections between NFRs, and gaining an approximate idea of past interest in these NFRs. Our results can allow researchers to plan future studies and to identify NFRs that have not received sufficient attention. They also help enable engineers to identify which NFRs to consider in conjunction with others of interest, and to think critically about how NFRs apply to different facets of the system-under-development.

2 BACKGROUND AND RELATED WORK

NFRs for Traditional Systems: NFRs are considered essential for ensuring the quality of software, but there are no agreed guidelines on how and when NFRs should be elicited, defined, documented, and validated [7]. Moreover, there is no consensus in the requirements engineering (RE) community regarding which step of the RE process NFRs should be considered and applied [4]. Significant research has been devoted to NFRs in RE, e.g., Doerr et al. applied a systematic, experience-based, method to elicit, document, and analyze NFRs with the objective of creating a sufficient set of traceable and measurable NFRs [5]. While most work such as this focuses on NFRs for traditional software, we are focused on NFRs specifically for systems that make use of ML to deliver functionality. Although many researchers have studied NFRs for traditional systems, very few studies to date have focused on NFRs for ML systems.

Requirement Engineering for ML Systems: Although there are approaches on how to use ML to improve RE tasks, there has not been extensive research on RE for ML systems [20]. Engineering of ML systems requires different and novel approaches due to their unpredictable nature and differences in their development process. It is crucial to clearly identify and define these differences, in order to offer tailored practices [10]. For traditional systems, activities related to requirements analysis and specification are often performed in the early phases of development, with requirements used downstream as part of design, implementation and verification. However, the activity flow often differs for ML systems due to their reliance on data and the unpredictability of ML results. Upfront problem definition for ML systems can be difficult, as building a clear definition of the problem often requires iterative exploration of data and processes—more so than in typical systems. As such, RE for ML systems has many unknown and unexplored areas, including an understanding of how NFRs differ for such systems.

NFRs for ML Systems: We discussed challenges and research directions for NFRs for ML systems [9]. Some of our knowledge about NFRs for traditional systems may no longer be applicable due to the non-deterministic behavior, as well as due to additional performance demands imposed by the need to process and act on large volumes of data. Some NFRs become more important (e.g., explainability), some become less relevant (e.g., modularity), there are differing trade-offs between NFRs (e.g., increased security often causes decreased usefulness), and there is no unified collection and consideration of NFRs for ML-enabled systems. We defined research direction for NFRs for ML systems, including exploring and defining NFRs, as well as reinterpreting and redefining NFRs that already exist for traditional systems.

In a recent interview study, we examined challenges regarding NFRs for ML in industry by identifying examples of the identification and measurement of NFRs and examining the importance that practitioners place on NFRs for ML [8]. The results of the interview study found that most NFRs as defined for traditional software are still relevant for ML-enabled systems. Some NFRs, such as flexibility, efficiency, usability, portability, reusability, and usability were identified as less important by some interviewees. However, they were still considered important by other interviewees. The NFRs identified in the interview study are listed in Table 1. We have defined each NFR, often in an ML context, based on both our experience and related literature, such as research papers, websites, blogs, and forums. In addition, we reported gaps to address, including identification, definition, scope, and perceptions of NFRs in an ML context [8]. In this work, we build on these results by beginning to explore the coverage of NFRs in research literature.

ML as Part of a Larger Software System: In a ML system, the "ML" is a small part of a larger system [19]. In traditional systems, NFRs can be identified over the whole system or elements of the system. In an ML context, NFRs can also be defined over different parts of the system. These elements may differ from traditional systems, and the differing nature of these elements may lead to a differing understanding of relevant NFRs. In our preliminary NFR definitions in Table 1, we have sometimes defined NFRs in ML terms, referencing the ML model or data. However, this is not done consistently, and not all potential elements of the system are considered. In an effort to improve how NFRs for ML systems are defined, we explore the idea of NFR "scope" further in this study.

[1]While little work has been conducted on the topic of NFRs for ML systems, there is certainly relevant research on individual quality attributes, such as fairness or security.

Table 1: Important NFRs for ML systems, identified in [8].

NFRs	Definition
Accuracy	The number of correctly predicted data points out of all the data points.
Adaptability	The ability of a system to work well in different but related contexts.
Bias	A phenomenon that occurs when an algorithm produces results that are systematically prejudiced due to erroneous assumptions in the ML process.
Completeness	An indication of the comprehensiveness of available data, as a proportion of the entire data set, to address specific information requirements.
Complexity	When a system or solution has many components, interrelations or interactions, and is difficult to understand.
Consistency	A series of measurements of the same project carried out by different raters using the same method should produce similar results.
Correctness	The output of the system matches the expectations outlined in the requirements, and the system operates without failure.
Domain Adaptation	The ability of a model trained on a source domain to be used in a different—but related—domain.
Efficiency	The ability to accomplish something with minimal time and effort.
Ethics	Concerned with adding or ensuring moral behaviors.
Explainability	The extent to which the internal mechanics of ML-enabled system can be explained in human terms.
Fairness	The ability of a system to operate in a fair and unbiased manner
Fault Tolerance	The ability of a system to continue operating without interruption when one or more of its components fail.
Flexibility	The ability of a system to react to changing demands or conditions.
Integrity	The ability to ensure that data is real, accurate and safeguarded from unauthorised modification.
Interpretability	The extraction of relevant knowledge from a model concerning relationships either contained in data or learned by the model
Interoperability	The ability for two systems to communicate effectively
Justifiability	The ability to be show the output of an ML-enabled system to be right or reasonable.
Maintainability	The ease with which a system or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment
Performance	The ability of a system to perform actions within defined time or throughput bounds.
Portability	The ability to transfer a system or element of a system from one environment to another.
Privacy	An algorithm is private if an observer examining the output is not able to determine whether a specific individual's information was used in the computation.
Reliability	The probability of the software performing without failure for a specific number of uses or amount of time.
Repeatability	The variation in measurements taken by a single instrument or person under the same conditions.
Retrainability	The ability to re-run the process that generated the previously selected model on a new training set of data.
Reproducibility	One can repeatedly run your algorithm on certain datasets and obtain the same (or similar) results.
Reusability	The ability of reusing the whole or the greater part of the system component for similar but different purpose.
Safety	The absence of failures or conditions that render a system dangerous
Scalability	The ability to increase or decrease the capacity of the system in response to changing demands.
Security	Security measures ensure a system's safety against espionage or sabotage.
Testability	The ability of the system to to support testing by offering relevant information or ensuring the visibility of failures.
Transparency	The extent to which a human user can infer why the system made a particular decision or produced a particular externally-visible behaviour.
Traceability	The ability to trace work items across the development lifecycle.
Trust	A trusted system is a system that is relied upon to a specified extent to enforce a specified security, or a security policy.
Usability	How effectively users can learn and use a system.

3 METHODOLOGY

Though NFRs for traditional software are fairly well-understood, there are still gaps in our foundational knowledge on NFRs for ML systems. We are eager to learn which NFRs for ML systems have been explored by other researchers and which are yet to be investigated heavily. We also want to learn how NFRs for ML are perceived by other researchers so that the definitions and scopes of such NFRs can be refined.

Hence, we have performed an exploratory study aimed at at establishing an initial scoping of the treatment of NFRs for ML and an initial estimation of the level of research that has been conducted on these NFRs. A systematic mapping study is primarily concerned with structuring a research area [13]. As we are performing an initial exploration of the scope of NFRs for ML systems, we have adapted the systematic mapping study concept for our purposes.

Our goals in this study are to (1) gain an approximate idea of the extent to which select NFRs have been studied by researchers, and (2), perform an initial clarification of the scope of these NFRs for ML systems. Specifically, we address the following research questions:

RQ1: Can the ML system NFRs be grouped into a small number of clusters based on shared characteristics?
RQ2: Which NFRs have received the most—or least—attention in existing research literature?
RQ3: Over which elements of an ML system can NFRs be defined?

To answer these questions, we grouped the NFRs into a small number of clusters based on their shared characteristics (Sec. 3.1). We performed the initial stages of a mapping study in order to gain a rough approximation of the how much research exists on each NFR—focusing on those NFRs that have been least investigated or belong to two particular clusters of interest (Sec. 3.2). Then, based on the titles and abstracts and past experience, we identify which elements of the system that these NFRs should be defined and measured over (Section 3.3).

3.1 NFR Clustering

In Table 1, we listed the NFRs found to be important in the interview study [8]. For each, we have defined them based both on our past experience and based on their treatment in a small sampling of research papers, websites, blogs, and forums. Based on these definitions, we are interested in grouping these NFRs into a small number of clusters, where each cluster contains NFRs that have similar meaning or purpose. Researchers can use these clusters to identify which NFRs may be related and able to collectively determine the quality of a system. Researchers could also perform secondary studies on particular clusters of NFRs. Developers can also use these clusters to identify which NFRs may be relevant to their particular needs or system-under-development.

We have created these clusters primarily through discussion of the NFRs and their definitions. During a series of meetings, we read and interpreted the definitions and debated their meaning. We then discussed and decided which cluster to assign an NFR to. We have placed NFRs in clusters if they are a similar purpose within system development or could be measured in a similar manner.

For example, the explainability of a ML system refers to the extent to which its internal mechanics can be explained in human terms. Transparency refers to the ability of the system to clarify the reasoning for its decisions to a human user. These NFRs differ in their exact meaning and assessment, but are both key elements in ensuring that ML systems operate in a clear and reasonable manner. Therefore, both should reside in the same cluster.

We also created a separate cluster for those NFRs that could not be put into any of the other clusters, as they lacked any shared characteristics with the NFRs in other clusters.

Our goal at this stage is not to create a formal hierarchy, as exists for NFRs for traditional systems [2]. Rather, our interest is in creating a lightweight organizational structure for use in understanding the scoping and definition of NFRs for ML systems.

3.2 Publication Volume Estimation

In this section, we describe our strategy for estimating the number of research papers for certain NFRs.

3.2.1 Initial Paper Search. We performed a database search—including all publications up to September 2021—in order to identify the research papers that may be relevant for each NFR. We selected Scopus, a meta-database, which includes research papers from peer-reviewed journals and conferences from multiple publishers such as IEEE, ACM, and Elsevier. Scopus is considered one of the most representative and rich in content for Software Engineering research and is used in many secondary studies [12].

We identified relevant search terms and developed search strings for the database search. We first derived the major terms (e.g., machine learning, non-functional requirements). Then, we identified synonyms or alternative spelling for the major terms from related literature, and based on our discussions. We also split major terms into more specific and clear terms. For example, we split the general term "non-functional requirements" into strings based on specific NFRs. Finally, we concatenated these terms to form search strings.

We apply one search string per NFR. The string includes that NFR, as well as terms related to machine learning: **("machine learning" OR "supervised learning" OR "unsupervised learning" OR "reinforcement learning" OR "deep learning")**.

For example, to identify papers on interoperability, we have used the search string: **("machine learning" OR "supervised learning" OR "unsupervised learning" OR "reinforcement learning" OR "deep learning") AND ("interoperability")**. As a second example, to identify papers related to usability, we have used the string: **("machine learning" OR "supervised learning" OR "unsupervised learning" OR "reinforcement learning" OR "deep learning") AND ("usability")**.

The number of papers found from this step give an upper limit on the number of relevant publications. Not all of these publications are likely to be relevant, as they may not relate to the use of such properties as NFRs for a ML system. For example, several of the results for maintainability described work which used ML to predict maintainability of another system, rather than focusing on maintainability of an ML system. Therefore, in the next step, we used a sample of publications to gain a finer estimation of the number of relevant publications for a subset of the NFRs.

3.2.2 NFR Selection. This upper limit gives some indication of the research interest in each NFR. To gain a clearer estimation of the percentage of those publications that are relevant, we have chosen to focus on a subset of the list of NFRs. Some NFRs, such as performance or security, have already received significant attention from the research community. We would recommend that future secondary studies focus specifically on these topics. We have instead focused on those NFRs that have received less focus from researchers, including those with a lower number of publications as well as those that we identified as being part of two clusters of interest (the "other" cluster and a cluster centered around tailoring a system to different environments).

We created a list of the number of publications found in the search results for each NFR. At first, we sorted the NFRs based on the number of publications, in decreasing order. We then excluded those NFRs that have more than 1,200 search results. For example, we excluded accuracy, as the number of retrieved papers was more than the threshold. Based on this threshold, we excluded 16 NFRs.

We then took into account which cluster we assigned each NFR to. If an NFR has more search results than the threshold but falls into the two clusters that we selected for initial inspection, then we included that NFR for consideration. As a result, we reincorporated usability and flexibility into our estimation, as those NFRs fall into these two clusters even though those have more search results than the threshold. We perform a more detailed analysis on 20 NFRs.

3.2.3 Estimating the Number of Relevant Papers for Selected NFRs. We estimate the number of relevant publications for each selected NFR by inspecting the titles and abstracts of a sample of 50 papers. We read the title and abstract of each publication and use inclusion and exclusion criteria to filter these publications, marking them as relevant or irrelevant. Each author determined the relevancy of each paper independently. We then discussed each disagreement in a meeting, using our criteria, and formed a final list.

Inclusion Criteria: The publication must discuss an NFR from Table 1. It must focus on the definition, identification, measurement, or challenges of a NFR for a ML system, or for an element of the system (e.g., the model). It must have been published in a peer-reviewed journal, conference, or workshop. The full text must be accessible and written in English.

Exclusion Criteria: The publication is focused on topics other than NFRs for ML systems. This includes publications where ML is used to measure, improve, or predict a NFR. For example, the authors used ML to classify requirements into different NFRs [14]. In such a case, the publication is not relevant for examining how such an NFR affects the development of a ML system. The publication simply uses the NFR as an evaluation criteria, but does not discuss or describe the use of the NFR during system development. For example, if an author uses completeness as part of their evaluation of the results of a system, but the actual research has no relation to improving the completeness of a ML system, then it is excluded. The publication was not written in English, not peer-reviewed, or lacks an available full text. Editorials, abstracts, book chapters, workshop summaries, poster sessions, prefaces, article summaries, interviews, news, reviews, comments, news, reviews, tutorials, panels, and discussions are excluded.

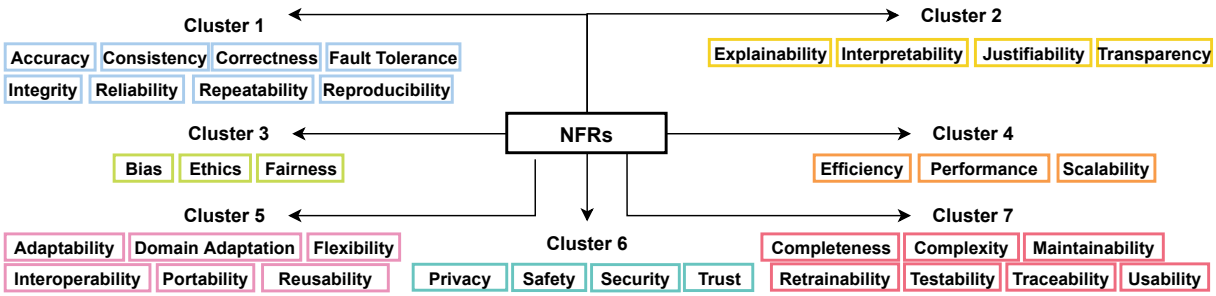

Figure 1: NFRs divided into clusters, based on shared characteristics.

Inter-coder Reliability: Following the process of individually reviewing 50 papers for selected NFRs, we calculated our agreement using Fleiss' kappa, a statistical measure for assessing ICR between a fixed number of raters. In some cases where the ICR was low, or where there were significant disagreements, we repeated the sampling process for a second set of 50 papers. In such cases, it was hoped that we could clarify our shared definition and estimation of the scope of the NFR. If the ICR either increased or stayed the same, this served to increase confidence in our understanding.

The final list of relevant papers, after discussion, gives an indication of the number of publications that may be relevant from that initial set retrieved from Scopus. This, in turn, offers an indication of research interest in the NFR.

Estimating the Number of Publications: We counted the number of publications that were deemed relevant from the first—and, in some cases, second—sample for each selected NFR. We used these counts along with the total number of papers found by Scopus to estimate the total number of included papers. This estimation is calculated by simply multiplying the total number of publications by the percentage of the sample that was deemed relevant.

For example, we found an upper limit of 851 publications for the transparency NFR. After screening 50 publications, we agreed to include 44 (88%). Extending to the full set of 851 papers, we estimate that 749 publications will actually be relevant. As a second example, we identified 214 publications for traceability. In this case, we sampled 100 publications, and decided that 10 were relevant (10%). Therefore, approximately 21 of the 214 are expected to be relevant to the treatment of the property as a NFR for ML systems.

We repeated this calculation for the rest of the selected NFRs, producing an estimation of the number of relevant papers for each. This is still a rough approximation of past research interest, but it is sufficient to provide an initial portrait of the field and to refine our own definitions and ideas regarding scope.

3.3 NFR Scope Determination

In order to clearly define or measure the attainment of an NFR, it must be understood exactly how the NFR applies to the system. This determination requires understanding whether a NFR relates to the system as a whole, or perhaps to a lower level of granularity within the system. In the case of a ML system, an NFR may be defined and measured over different aspects of the ML application. For example, an NFR may apply differently when we discuss the training data, the algorithm that uses the training data to build a model, or to the model trained on that data.

Therefore, we have first determined which elements of a ML system are particularly relevant when we discuss the NFRs for such systems. We then used our existing definitions, past experience, and the titles and abstracts of the relevant studies examined in the previous step in order to determine to which of these elements each NFR was applicable. In a series of meetings, we discussed each NFR in relation to these system elements. In each case, we made a determination by coming to an agreement and discussing any cases where we disagreed—generally by identifying an example of how that NFR is applied to that element. For example, repeatability refers to the level of variability in the behavior of the system. Repeatibility is a property of the results—or of the system as a whole—rather then a property of the model, algorithm, or training data. It is the results that vary, not the model itself. This scoping is intended as a starting point for establishing detailed definitions for each NFR in an ML system context.

4 RESULTS AND DISCUSSION

NFR Clustering (RQ1): We were able to create six different clusters, where each cluster includes the NFRs that share similar properties and purposes. For example, after analyzing their definitions, we found that ethics, bias, and fairness shares similar meanings and serve similar purposes. Therefore, we put these three NFRs into the same cluster.

The clusters are presented in Fig. 1. Cluster 1 includes NFRs that are related to assessing the functional correctness of ML systems and aspects of correctness. This includes the core correctness, as well as assessment of correctness (e.g., accuracy) and variance (e.g., reliability, consistency). Cluster 2 contains NFRs related to understanding the internal decisions or results of applying ML (e.g., transparency, explainability). The NFRs related to ethical aspects of ML systems, such as fairness and bias, form cluster 3. NFRs related to the performance (e.g., speed) of an ML system are contained in cluster 4. The qualities related to tailoring and adjustment of the ML system to different environments (e.g., flexibility, adaptability) are grouped in cluster 5. Concerns related to privacy and security are grouped together in cluster 6. The NFRs that do not share similar properties are grouped in cluster 7.

Previous work has presented NFRs in terms of a hierarchy (e.g., [3]) or as part of a interdependency graph (e.g., [4]). Our goal was not to suggest a definite hierarchy, but to group NFRs to clarify relatedness and scope, particularly for future research studies. For example, a study may focus on a particular cluster or one or two related NFRs. These clusters can also help practitioners understand the similarity

Table 2: NFRs with number of search results, number of relevant publications, kappa values (agreement on sample), and final paper volume estimation for select NFRs. We only examined a second sample in cases where we wanted to see if agreement would improve.

NFR	Search Results	Relevant (1)	Kappa (1)	Relevant (2)	Kappa (2)	Est. Pubs.
Performance	114853					
Accuracy	92669					
Efficiency	22247					
Security	19142					
Complexity	16997					
Privacy	6388					
Safety	5848					
Reliability	5620					
Bias	4118					
Scalability	3595					
Consistency	2936					
Flexibility	2764	23 (46%)	0.54			1271
Interpretability	2418					
Trust	1965					
Reproducibility	1796					
Domain Adapt.	1732	47 (94%)	0.63			1628
Usability	1270	21 (42%)	0.50	29 (58%)	0.44	635
Adaptability	1177	34 (68%)	0.50			800
Fairness	1089	45 (90%)	0.41			980
Correctness	1045	16 (32%)	0.53			334
Integrity	1015					
Transparency	851	44 (88%)	0.70			749
Explainability	706	44 (88%)	0.22			621
Fault Tolerance	553	26 (52%)	0.68			288
Interoperability	532	9 (18%)	0.45			96
Completeness	372	23 (46%)	0.40	25 (50%)	0.58	179
Portability	346	21 (42%)	0.45			145
Ethics	331	31 (62%)	-0.03			205
Reusability	321	24 (48%)	0.55			154
Maintainability	277	6 (12%)	0.30	9 (18%)	0.72	42
Traceability	214	4 (8%)	0.61	6 (12%)	0.61	21
Repeatability	171	17 (34%)	0.44			58
Testability	77	4 (8%)	0.54	2 (4%)	1.00	5
Justifiability	3	0 (0%)	1.00			0
Retrainability	0					0

of NFRs and provide guidance on which related NFRs they should consider while developing ML systems.

Estimated Number of Publications (RQ2): We used the search strings described in Sec. 3.2 to identify an upper limit on the number of relevant publications for each NFR. The number of identified publications is presented the second column of Table 2. We found the most results for performance, accuracy, and efficiency; while, repeatability, testability, and justifiability yielded the fewest results. We found no research papers in Scopus for retrainability, potentially indicating that this term is not common.

We can sum the total number of search results for each cluster, finding that cluster 4 (performance, …) has 140695 results, cluster 1 (accuracy, …) 105805, cluster 6 (security, …) 33343, cluster 7 ("other" NFRs) 19207, cluster 5 (adaptability, …) 6872, cluster 3 (bias, …) has 5538 and cluster 2 (explainability, …) has 3978 results.

We can reflect on the number of papers found via the Scopus search. The number of papers for accuracy is very high, as researchers and practitioners are very focused on prediction accuracy. We also found more papers for usability than we expected, even when excluding papers using usability as a synonym for applicability, and find it encouraging that research is focusing on human-oriented aspects.

We were surprised that no publications were found for retrainability, even though practitioners mentioned retrainability as important [8]. We hypothesize that these ideas are being discussed using alternative terms. Similarly, we were surprised by few search results on testability, but this may again be due to use of different terms. We also expected more search results on fairness, as we perceive that researchers and practitioners are focusing more on this topic. This may be due to commonality and split of results amongst bias, fairness, and ethics.

The performance and accuracy clusters (4 and 1) show the most raw results, followed by the security cluster. These results are generally in line with our expectations. We can see a particular interest in cluster 4, including performance. In ML terminology, performance often denotes a form of accuracy or correctness, as opposed to time or resource usage—as the term is often used in typical SE. Cluster 7 also has a relatively large number of results, mainly due to the inclusion of complexity.

We were surprised that clusters 2—containing explainability—and cluster 3—containing bias—yielded relatively fewer results. Even though these are perceived as hot topics in research, either the volume of papers is still relatively small, this work includes terms which differ from the NFRs included as part of our search.

Although these results are a useful staring point, we refine our estimation for a subset of NFRs to estimate how many publications are relevant. When selecting NFRs for a more detailed estimation, we focused on NFRs that are less researched (but still potentially important), and those in Clusters 5 and 7. We applied the inclusion-exclusion criteria and ICR process described in Sec. 3.2.3 for a sample of fifty papers of each selected NFR. We present the number of publications found to be relevant for each, along with the inter-coder reliability in Table 2. In some cases, we also conducted a second sample of an additional 50 publications.

We can evaluate the strength of our agreement as follows: < 0.0 is considered as poor agreement, 0.00-0.20 as slight, 0.21-0.40 as fair, 0.41-0.60 as moderate, 0.61-0.80 as substantial, and 0.81-1.00 as almost perfect [16]. We attained a substantial rage of scores in terms of ICR for the NFRs. One result (ethics) was poor, with our coding being worse than random. However, we attained fair results for three other NFRs, and moderate or better for the remaining 15. Our final estimation is shown in the final column in Table. 2.

Focusing on five of the NFRs in cluster 7, we can examine our change in agreement after discussion. After the first sample of 50 papers, the ICR scores were fair for maintainability and completeness, and moderate for usability, completeness, maintainability, traceability, and testability. After a discussion among all three authors about our perception and interpretation of the NFR definitions and the inclusion and exclusion criteria, the ICR for the second sample generally improved and ranged between moderate (e.g., completeness, maintainability, traceability) to perfect (e.g., testability). We note, however, that our ICR score for usability actually decreased in the second round. To some extent, these score also depend on the percentage of relevant publications. The less often papers are relevant (e.g., testability), the easier it is to gain high agreement.

For some NFRs, it was difficult to agree on inclusion. For example, we had particularly low agreement for ethics and explainability. For some NFRs, we can often make a clear distinction between studies focused on improving attainment of that NFR when designing a ML system versus irrelevant studies (e.g., those that use ML to predict attainment of an NFR for a traditional system). With topics like ethics and explainability, it was harder to make this type of distinction, and there were more disagreements on particular

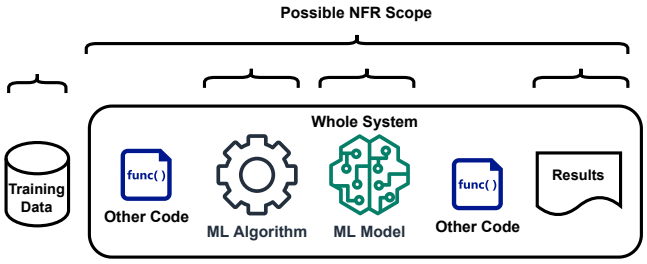

Figure 2: Possible scope for NFRs over system elements.

studies. In these cases, future work may require clearer definitions or more specific criteria.

NFR Scoping Over System Elements (RQ3): Clear definition of NFRs in a ML system context,requires understanding which specific elements of an ML system that an NFR is applicable to.

As a starting point for building this understanding, we believe that NFRs can be defined over the following parts of an ML system. The **training data** used by the ML algorithm as the basis for making decisions. The **algorithm** that performs the learning task. This includes algorithms that operate on training data, as well as those that perform learning tasks based on feedback, such as reinforcement learning agents. We also consider the specific implementation of the algorithm here.The core **model or artifact**[2] built by the algorithm for use in making decisions. For example, the algorithm may use the training data to build a model that makes decisions in new situations based on learned connections between data items. The **resulting decisions or behaviors** made as a result of applying the model. Finally, **the ML system as a whole**.

These parts are illustrated in Fig. 2. It is possible that more elements may be applicable in the future, e.g., NFRs over features of a data set or over specific types of functionality operating on the results of ML, but we start with this initial list of system elements to understand the scope of the selected NFRs.

Our overall determination of which system elements a particular NFR can be defined over is presented in Table. 3. Note that our estimation of NFR scope is an initial estimation based on our experiences and the sampled abstracts. The scope of each NFR likely will evolve over time as more data and examples are gathered.

To illustrate our determinations, we select a number of examples. For example, we determined that the NFR flexibility can be defined over the ML algorithm, the ML model, and the whole system. However, we believe it is not applicable to the training data and the results. Consider a definition of flexibility by Ladiges et al. [15], "*flexibility is an indicator for the ability of a system to react to changing demands or conditions*". We can adapt this definition to different parts of the ML system, as in the following[3]. The flexibility of an ML algorithm is *the ability of an algorithm to react to changing demands and conditions, without significant re-implementation.* The flexibility of an ML model is *the ability of a model to react to changing inputs and contexts in a useful way, without retraining.* Finally, the flexibility an ML system could use the initial definition or, more specifically, *the ability of a ML system to react to changing demands or conditions without extensive re-implementation or re-training.* On the other hand, we struggle to define flexibility over training data. It makes sense to think of the reusability of training data, e.g., to train ML systems for different context and purposes with some of the same data, but what does it mean for data itself to be flexible? Similarly, results can be reusable, but it is not clear how they can be flexible. We opt to omit these definitions from our consideration.

Similarly, the NFR usability can be defined over the ML algorithm, the ML model, the results, and the whole system; but may not be applicable over the training data. If we take the simple definition of usability from Table 1, "*how effectively users can learn and use a system*", this definition makes sense over the whole system. We can also define this NFR over specific ML elements. The usability of an ML algorithm is *how effectively users can learn and use an algorithm to train an ML model as part of a system.* The usability of an ML model is *how effectively users learn to use an ML model at run-time in order to get results.* The usability of ML results is *how effectively users can understand and apply ML results for some practical purpose.* However, we struggle to create a definition for the usability of the training data. Does a user learn data? Although a user uses data, is some data more usable than others, or is that more a matter of data quality and data appropriateness?

When processing the abstract and titles for usability, we noted that many authors used usability as more a binary term meaning applicability—e.g., usability means that data can be used to train a model. We disagree with this use of usability, as usability is more appropriate as a user-centered qualitative concept. If we exclude general applicability, we find it hard to define usability of data.

Other combinations of system elements and NFRs can be defined similarly. We can see that all NFRs can be defined over the whole system, reflecting the scope of NFRs over traditional systems. Almost all apply over the model, and most to the algorithm. Fewer NFRs apply to the training data and the results, but there is no clear pattern here. Some apply to both, others only to one.

We are working towards a framework for the definition of each NFR over each part, including a checklist on which part of the

[2]This notion also encompasses the policy learned by an agent in reinforcement learning, or other rules "learned" by the algorithm in other techniques.

Table 3: System elements that NFRs can be defined over

		System Element the NFR Can be Defined Over				
NFR	Cluster	Train. Data	Algo.	Model	Results	Whole System
Completeness	1	✔	✖	✔	✖	✔
Correctness	1	✔	✔	✔	✔	✔
Fault Tolerance	1	✖	✔	✔	✖	✔
Integrity	1	✔	✔	✔	✔	✔
Repeatability	1	✖	✖	✖	✔	✔
Explainability	2	✖	✔	✔	✔	✔
Transparency	2	✖	✔	✔	✔	✔
Ethics	3	✔	✔	✔	✔	✔
Fairness	3	✔	✔	✔	✔	✔
Adaptability	5	✔	✔	✔	✔	✔
Domain Adaptation	5	✔	✔	✔	✔	✔
Flexibility	5	✖	✔	✔	✖	✔
Interoperability	5	✖	✔	✔	✖	✔
Portability	5	✔	✔	✔	✖	✔
Reusability	5	✔	✔	✔	✖	✔
Maintainability	7	✔	✔	✔	✖	✔
Testability	7	✖	✔	✔	✔	✔
Traceability	7	✔	✔	✔	✔	✔
Usability	7	✖	✔	✔	✔	✔

[3]We note that these definitions may have significant overlap with definitions for NFRs such as adaptability, resuability, or portability, which is precisely why these NFRs are placed in the same cluster—cluster 5, in this case.

system a particular NFR can be defined. We hope that such exploration can lead to a deeper understanding of each NFR and their application to ML systems.

4.1 Threats to Validity

External Validity: We have only used Scopus, which may mean we miss relevant papers in other databases. However, Scopus is a meta-database that is rich in content on computer science research from multiple publishers. We searched papers in Scopus up to September 2021, and there may be newer papers that are missed. Future secondary studies should repeat the search process.

The search string was confined to a small set of terms and keywords, focusing on only a subset of NFRs. We could have searched for alternative terms and stems like "interoper" for interoperability. However, it would be difficult to find equivalent stems for all NFRs (e.g., security) and may have led to an unmanageable increase in the volume of papers without a significant increase in relevant results. Our goal is not to make a conclusive statement on the number of publications, but to gain an approximate idea of the interest in each NFR. A sample is sufficient for such purposes.

Internal Validity: There is potential bias in determining paper inclusion. To mitigate this risk, we defined shared inclusion criteria, each of the authors went through each title and abstract separately, and we made a collective decision in cases of disagreement. Our ICR results are often good, and performing a second sample yielded consistent or better ICR scores for all but one NFR.

The clusters we created may be subjective to our experiences and opinions. NFRs could be arranged differently, but we believe our clusters are a good starting point to help organize and direct future research. Further work may add to or adjust the clusters as new evidence is found.

Our consideration of the scope of NFR definitions may also be subjective. We made these judgements in agreement between all authors, discussing difficult cases. We have tried to justify our selection for a sample of NFRs. Future work will adjust our scoping decisions when more evidence or examples are found.

5 CONCLUSIONS

In this work, we aimed at understanding and exploring definitions, scope, and the extent of existing research on NFRs for ML systems, as we believe that both the research community and industry lack knowledge on NFRs for ML systems compared to understanding in traditional systems. The results show that researchers have focused on many NFRs for ML systems, but the amount of attention directed to each NFR differs drastically. Some NFRs received more attention and were explored more (e.g., peformance, accuracy, efficiency) compared to other NFRs (e.g., maintainability, traceability). Although such differences were expected, it is useful estimate interest with concrete numbers.

We created six clusters of NFRs based on the similarity of characteristics and meaning of NFRs, and one cluster of NFRs which does not share similar properties, with the objective of helping researchers to focus on a particular cluster for their future systematic review studies. These clusters will also help practitioners to understand the similarity of NFRs and provide them a direction on which NFRs they need to consider while developing ML systems.

We defined NFRs over different granular levels of the ML systems based on the meaning and purpose of those NFRs. This can help practitioners to understand on which part of the ML system a particular NFR can be considered while developing ML systems. Our future work includes a comprehensive mapping study to identify the current state-of-the-art on selected NFRs for ML systems research, and a framework to guide consideration of NFRs over different elements of ML systems.

REFERENCES

[1] Reuben Binns. 2018. Fairness in machine learning: Lessons from political philosophy. In *Conference on Fairness, Accountability and Transparency*. PMLR, 149–159.

[2] Barry W Boehm, John R Brown, and Mlity Lipow. 1976. Quantitative evaluation of software quality. In *Proceedings of the 2nd international conference on Software engineering*. 592–605.

[3] Joseph P Cavano and James A McCall. 1978. A framework for the measurement of software quality. In *Proceedings of the software quality assurance workshop on Functional and performance issues*. 133–139.

[4] Lawrence Chung, Brian A Nixon, Eric Yu, and John Mylopoulos. 2012. *Non-functional requirements in software engineering*. Vol. 5. Springer Science & Business Media.

[5] Joerg Doerr, Daniel Kerkow, Tom Koenig, Thomas Olsson, and Takeshi Suzuki. 2005. Non-functional requirements in industry-three case studies adopting an experience-based NFR method. In *13th IEEE International Conference on Requirements Engineering (RE'05)*. IEEE, 373–382.

[6] Jonas Eckhardt, Andreas Vogelsang, and Daniel Méndez Fernández. 2016. Are" non-functional" requirements really non-functional? an investigation of non-functional requirements in practice. In *Proceedings of the 38th International Conference on Software Engineering*. 832–842.

[7] Martin Glinz. 2007. On non-functional requirements. In *15th IEEE International Requirements Engineering Conference (RE 2007)*. IEEE, 21–26.

[8] Khan Mohammad Habibullah and Jennifer Horkoff. 2021. Non-functional Requirements for Machine Learning: Understanding Current Use and Challenges in Industry. In *2021 IEEE 29th International Requirements Engineering Conference (RE)*. IEEE, 13–23.

[9] Jennifer Horkoff. 2019. Non-functional requirements for machine learning: Challenges and new directions. In *2019 IEEE 27th International Requirements Engineering Conference (RE)*. IEEE, 386–391.

[10] Fuyuki Ishikawa and Nobukazu Yoshioka. 2019. How do engineers perceive difficulties in engineering of machine-learning systems?-questionnaire survey. In *2019 IEEE/ACM Joint 7th International Workshop on Conducting Empirical Studies in Industry (CESI) and 6th International Workshop on Software Engineering Research and Industrial Practice (SER&IP)*. IEEE, 2–9.

[11] Toshihiro Kamishima, Shotaro Akakamishima2011fairnessho, and Jun Sakuma. 2011. Fairness-aware learning through regularization approach. In *2011 IEEE 11th International Conference on Data Mining Workshops*. IEEE, 643–650.

[12] Staffs Keele et al. 2007. *Guidelines for performing systematic literature reviews in software engineering*. Technical Report. Citeseer.

[13] Barbara Kitchenham and Stuart Charters. 2007. Guidelines for performing systematic literature reviews in software engineering. (2007).

[14] Zijad Kurtanović and Walid Maalej. 2017. Automatically classifying functional and non-functional requirements using supervised machine learning. In *2017 IEEE 25th International Requirements Engineering Conference (RE)*. Ieee, 490–495.

[15] Jan Ladiges, Alexander Fay, Christopher Haubeck, and Winfried Lamersdorf. 2013. Operationalized definitions of non-functional requirements on automated production facilities to measure evolution effects with an automation system. In *2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA)*. IEEE, 1–6.

[16] J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics* (1977), 159–174.

[17] Bashar Nuseibeh and Steve Easterbrook. 2000. Requirements engineering: a roadmap. In *Proceedings of the Conference on the Future of Software Engineering*. 35–46.

[18] Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. 2019. Dissecting racial bias in an algorithm used to manage the health of populations. *Science* 366, 6464 (2019), 447–453.

[19] David Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Francois Crespo, and Dan Dennison. 2015. Hidden technical debt in machine learning systems. *Advances in neural information processing systems* 28 (2015), 2503–2511.

[20] Andreas Vogelsang and Markus Borg. 2019. Requirements engineering for machine learning: Perspectives from data scientists. In *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*. IEEE, 245–251.