A Systematic Literature Review on Machine Learning for Automated Requirements Classification

J. Manuel Pérez-Verdejo, Ángel Juan Sánchez-García,
Jorge Octavio Ocharán-Hernández
School of Statistics and Informatics
Universidad Veracruzana
Xalapa, Veracruza, México
manuel.vrdjo@gmail.com, angesanchez@uv.mx, jocharan@uv.mx

***Abstract*—The development of quality software begins with the correct identification of the system needs. These requirements represent the basis of the subsequent activities in the software life cycle. The correct identification of these requirements in their different categories impacts on the actions taken to meet them. However, this classification can be often time-consuming or error-prone when it comes to large-scale systems, so different proposals have been made to assist in this process automatically. This systematic literature review identifies those applications of Machine Learning techniques in the classification of software requirements. In this regard, 13 articles were identified, from which relevant information on the applied algorithms, their training process, and their evaluation metrics are analyzed. From the results obtained, it is identified that the most recurrent classification algorithms featured on the identified studies are Naïve Bayes, Decision Trees, and Natural Language Processing algorithms. The most frequent training datasets are academic databases and collected user reviews.**

***Index Terms*—Requirements Engineering, Requirements Classification, Machine Learning, Classification, Systematic Literature Review**

I. Introduction

Requirements Engineering (RE), conceived as a subdiscipline dedicated to encompassing the capabilities and attributes that a system must fulfill [1], involves several of the most relevant activities within the software development process. Moreover, its elicitation, analysis, specification, and validation stages [2] represent a set of activities focused on understanding the customer's needs regarding the software to be developed.

In this regard, it is in the requirements analysis where the customer input results of the elicitation methods are refined into classified requirements for tracing. Thus, in this phase, the requirements analyst classifies the user statements into nine different categories of requirements [1]: business requirements, external interface requirements, quality attributes, data requirements, functional requirements, solution ideas, business rules, user requirements, and constraints.

However, this activity including the elicitation process is usually time-consuming and error-prone [3], therefore the application of new techniques that assist in this type of task means a field of opportunity. Thus, the use of machine learning becomes a viable alternative, by dedicating its work to the construction of programs able to automatically improve with the experience [4].

With the aim to provide initial insight into the application of machine learning strategies to automatically classify software requirements, the present study carries out a systematic literature review, to portrait the state of the art of this field.

This paper is organized as follows. In section II the related work is presented. Section III describes the research method followed to conduct the review in machine learning algorithms for requirements classification. In Section IV, the conducting process of the systematic review is described, as well as the primary studies found. Section V presents the results obtained from data extraction and synthesis performed in the review. In section VI the results found are discussed. Finally, the conclusions and the identified insights are summarized in section VII.

II. Related Work

Software Engineering has proven to be a suitable and successful field for automation, as techniques and approaches specific to the Machine Learning area are increasingly being implemented into several processes among the software lifecycle [5]. In particular, the activities involving the Requirements Engineering subdisciplines (elicitation, analysis, specification, and validation) [1] have turned into the focus of several tools and frameworks aiming to automatically assist them.

The systematic review conducted by Meth, Brhel, and Maedche [3] studies the state of the automated requirements elicitation tools, as of 2013. Categorizing the primary studies found by their scope and degree of automation (among others), their approach portraits the novel applications by the following scopes: "Abstraction Identification", "Requirements Model Generation", "Requirements Quality Analysis" and "Requirements Identification". Among their primary studies, 11 papers addressed the requirements identification. However, most of the studies in this category were semi-automatic implementations. This may be due to the fact, that the term "Machine Learning" was not one of the terms that lead the search, restraining the automated part only to results that included the term "automated".

Additionally, in [6] Binkhonain and Zhao carried out a systematic literature review on Machine Learning models

for identification and classification non-functional requirements(NFR). In this regard, they found several approaches that implement at least one machine learning algorithm. Among their contribution, they summarize the different approaches followed for the Natural Language Processing strategy, being techniques such as stemming, Stopwords removal, and Part-of-Speech the most common. On the other hand, and due to the focus on NFR, their study allows them to identify subcategories related to this kind of requirements, in particular, they highlight three: security, performance, and usability.

III. Research Method

The research method followed to conduct the review was based on the guidelines described by Kitchenham and Charters [7]. Hence, the procedure was divided into three stages: (1) planning, featured in this section, and (2) conducting, and (3) results, both shown in section V.

A. Planning

In this phase, the following activities that guided the review are described: the specification of the research questions, definition of the data sources and the search strategy, and the definition of the selection criteria. Those are further described in the following subsections.

1) Research Questions: The research questions elaborated to carry on the systematic review are:

- *RQ1: Which machine learning algorithms have been adopted to classify requirements?*
- *RQ2: What metrics have been used to measure the performance of those algorithms?*
- *RQ3: What kind of software projects have been used to train those algorithms?*
- *RQ4: Which requirements categories are the most frequent to train those algorithms?*

The motivation of RQ1 is to find evidence of the application of specific machine learning techniques in the RE, specifically in the classification of requirements. However, the classes to be identified are expected as belonging to one of the requirement categories identified by Wiegers and Beatty [1]. Regarding RQ2, it establishes the goal of identifying how to evaluate the proposed models, for this way of comparing results between them.

The RQ3 is determined to understand the domain in which the model proposals are focused. Therefore, it is possible to know the necessary means to adopt general models to specific domains. On the other hand, RQ4 is focused to know the categories of requirements that have been considered for the application of those classifiers.

2) Search Strategy: The search for studies was carried out in four academic databases, access to which is provided by the National Consortium of Scientific and Technological Information Resources (abbreviated CONRICyt in Spanish). Table I shows their names and pages.

Additionally, the search process was carried out guided by the three following search strings:

- *"Requirements Engineering" AND "Machine Learning"*
- *"Software Requirements Elicitation" AND ("Machine Learning" OR "Machine Learning Model")*
- *("Machine Learning" OR "Machine Learning Model") AND "Software Requirements Classification"*

TABLE I
Selected Databases

Database	URL
IEEEXplore	https://ieeexplore.ieee.org
ACM Digital Library	https://dl.acm.org
Science Direct	https://www.sciencedirect.com
SpringerLink	https://link.springer.com

The first is related to identifying applications belonging to the machine learning area in Requirements Engineering, this in order to find a more general approach for these implementations. The second search string is established as the analysis phase is considered to be the phase immediately following the elicitation [1]. Finally, the third search string is meant to directly identify requirements classification papers.

3) Advanced Search: The first stage of inclusion and exclusion criteria was performed directly to the search engines of the academic databases, in order to reduce results beyond the review scope. With this regard, publications that met the following requirements were collected:

- Published between 2010 and 2019
- Is either a journal or a conference paper
- Is accessible
- Features the search terms either in the title or the abstract

4) Primary Studies Selection: On this matter, inclusion and exclusion criteria were established for the selection of primary studies. The process of applying these criteria was carried out in two stages. The criteria within the stages are described in Table II.

TABLE II
Inclusion and Exclusion Criteria

Stage	Inclusion Criteria	Exclusion Criteria
Second Stage	1. The paper is published in English. 2. Either the title or the abstract leads to answer at least one research question	1. The paper is duplicated. 2. Neither the title nor the abstract features at least one of the search terms
Third Stage	1. The domain of the study is focused on the application of machine learning in Software Requirements Engineering.	1. The domain of the study does not consist in the application of some technique of requirements classification in any of the categories defined in [1].

5) Data Extraction: From each identified article once all the criteria have been applied, the following information is collected: title, authors, publication year, source, type of publication, reference, keywords, abstract, and the answer to the research questions addressed. Additionally, topic modeling

analysis is performed to identify the most frequent terms and trends throughout the identified studies. For this procedure, the transcribed version of the articles featured on their web pages was required. If it wasn't available, the content was extracted directly from the file.

6) Data Synthesis: In order to identify the evidence to answer the research questions, the narrative synthesis method was selected. In this regard, it was possible to extract the information of interest by detailed reading [8].

7) Quality Assessment: A seven-question checklist was designed to provide a quality assessment of the primary studies, inspired by the example shown in Kitchenham and Charters' guidelines [7]. The questions established for the criteria are presented in table III. Articles that score more than 4 of the questions in the criteria are included.

TABLE III
QUALITY ASSESSMENT CRITERIA

ID	Criteria
Q1	Is the aim of the study clearly described?
Q2	Does the study follow any method?
Q3	Is it clear what projects were used to construct each model?
Q4	Is it clear how the model metrics were measured?
Q5	Were all model construction methods fully defined?
Q6	Are the categories to classify clearly defined?
Q7	Are the links between data, interpretation and conclusions clearly expressed?

IV. CONDUCTING

Once the searches were made with the predefined criteria in the academic search engines, 325 articles were obtained. From these publications, a superficial review was made regarding the titles and abstracts, to identify potential answers to the research questions. As a result of this preliminary analysis, 55 papers were identified. Moreover, those documents that had already been identified, either in another database or with another search string were excluded, the last to avoid repeated articles. Finally, a detailed reading of the remaining 38 studies was made, until all those that met the criteria were identified, resulting in a total of 13 articles. Figure 1 describes this process and Table IV shows the studies included in the review, their publication year and source. Additionally, the quality assortment score of the studies is shown in Table V.

V. RESULTS

This section summarizes the results of the review, discusses the findings made, and answers the research questions.

A. Data Extraction

Besides the papers' metadata provided in their academic database entry (titles, abstracts, publication year, and venue), their full-text content and references were extracted. With these data, it was possible to identify the information regarding publication trends on the topic. Specifically, the years of publication, conferences or journals where the studies were presented, and their shared references were explored.

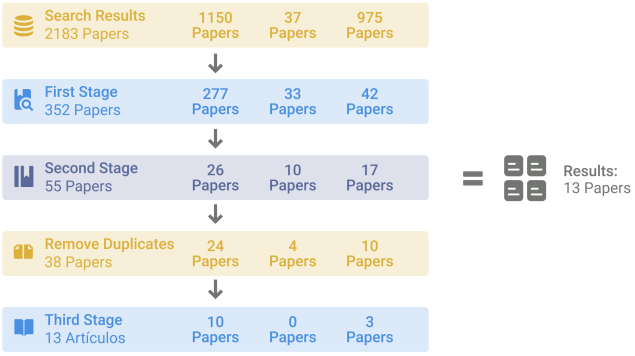

Fig. 1. Number of papers included during the study selection process

TABLE IV
SYSTEMATIC REVIEW STUDIES

ID	Author	Date	Source
S1	Abad et al. [9]	2017	International Requirements Engineering Conference
S2	Baker et al. [10]	2019	Computer Software and Applications Conference
S3	Dekhtyar and Fong [11]	2017	International Requirements Engineering Conference
S4	Iqbal, Elahidoost and Lucio [12]	2019	Asia-Pacific Software Engineering Conference
S5	Jindal, Malhotra and Jain [13]	2016	International Conference on Advances in Computing, Communications and Informatics
S6	Kurtanović and Maalej [14]	2017	International Requirements Engineering Conference
S7	Li et al. [15]	2018	Journal of Systems and Software
S8	Lu and Liang [16]	2017	International Conference on Evaluation and Assessment in Software Engineering
S9	Marinho et al. [17]	2018	International Conference on the Quality of Information and Communications Technology
S10	Riaz [18]	2014	International Requirements Engineering Conference
S11	Sharma et al. [19]	2014	International Advance Computing Conference
S12	Taj et al. [20]	2019	International Conference on Software and Information Engineering
S13	Wang et al. [21]	2018	International Symposium on Empirical Software Engineering and Measurement

1) Publication Year: Growth in interest in the application of machine learning for software requirements classification was distinguished. Specifically, 2017 is depicted as the year with the most published articles in this matter, as 4 studies were published. Figure 2 shows the trends in publications addressing automatic requirements classification.

2) Venue: The International Requirements Engineering Conference is identified as the main source for the publication of articles of this matter, as four articles were published in proceedings of this congress. Regarding the remaining studies, their publication was conducted in different conferences or journals as seen in Table IV.

TABLE V
PRIMARY STUDIES QUALITY ASSESSMENT

ID	Q1	Q2	Q3	Q4	Q5	Q6	Q7
S1	1	1	1	1	1	1	0
S2	1	1	1	1	1	1	1
S3	1	1	1	1	1	1	1
S4	1	1	1	0	0	1	1
S5	1	1	0	1	1	0	1
S6	1	1	1	1	1	1	1
S7	1	1	1	1	1	1	1
S8	1	1	1	1	1	0	1
S9	1	1	1	1	1	1	1
S10	1	1	1	1	1	1	1
S11	1	1	1	1	0	1	0
S12	1	1	0	1	1	0	1
S13	1	1	1	1	0	1	1

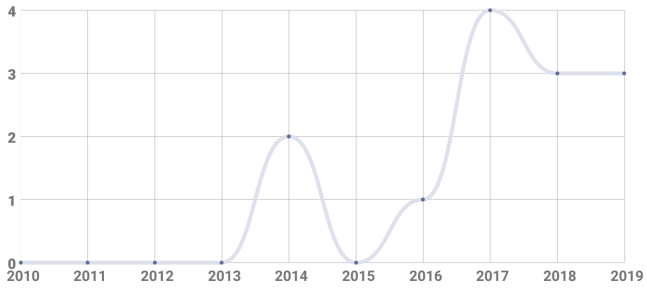

Fig. 2. Primary studies by year

3) Shared References: All the references made in the studies were collected and compared, thus it was possible to identify the ten most cited studies [9], [14], [22]–[29]. The most frequent reference was the study by Cleland-Huang et al. in 2006 [22]. Moreover, this article is identified by the studies that refer to it, as one of the earliest applications of machine learning for automated requirements classification. It was also found that more studies among the ten most frequent are authored by Cleland-Huang [23], [27]. It should be noted that [9] and [14], found in the systematic review process, are also among the most cited. Figure 3 shows the frequency with which these studies were cited.

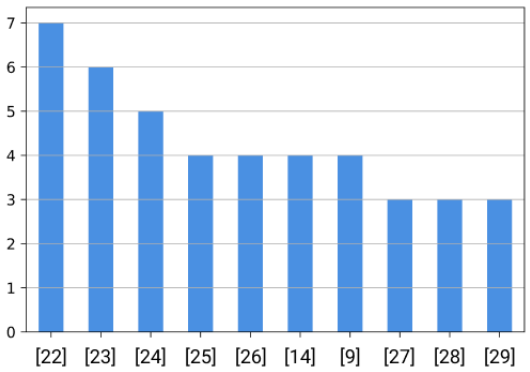

Fig. 3. Top 10 most cited studies

4) Language Processing and Topic Modeling: Following the data extraction, the content was analyzed using the Gensim library for text analysis and visualization [30]. For this process, the full-text was collected via the transcribed version on the studies' download page. In case it was not available, the extraction was done manually. For this analysis, images, tables, equations, as well as their corresponding captions were excluded.

Once gathered the textual content, it was possible to plot a word cloud with the most frequent terms and bigrams throughout all the studies analyzed. In this kind of visualization, the font size represents the frequency with which a bigram or individual term appeared [31]. In Figure 4 it is possible to visualize this representation.

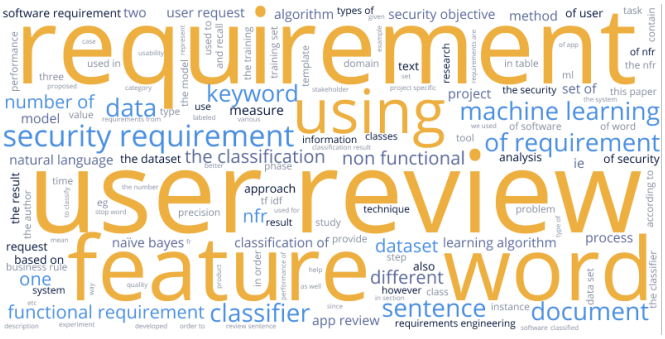

Fig. 4. Word cloud of the most frequent terms in the identified studies

As a result of this representation, it can be determined that the most frequent terms throughout the studies analyzed were "requirement", "user review", "feature" and "word" and to a lesser extent "security requirement", "machine learning" and "classifier".

In addition, a topic modeling process was carried out using the LDA algorithm (Latent Dirichlet Allocation), similar to the process conducted in the systematic review on [32]. The configuration was set to find four different topics. As a result, the figure 5 shows another word cloud, in this case, with the most frequent terms of each topic located. It is important to emphasize that it is possible for a term to intersect in more than one topic, especially with very frequent words such as "requirement".

With the generated LDA model, it is possible to identify the value or weight of a word concerning the topic in which it is found. Thus, Figure 6 presents the relationship between each term and the topic to which it belongs, showing its frequency and weight among each cluster. The textual analysis, the studies' data, and search results can be accessed for replication purposes in the project's repository.[1]

B. Data Synthesis

1) RQ1: Which machine learning algorithms have been adopted to classify requirements?: The study developed by Sharma, Bhatia, and Biswas [19] consists of the application of

[1]https://github.com/quality-attributes/systematic-review

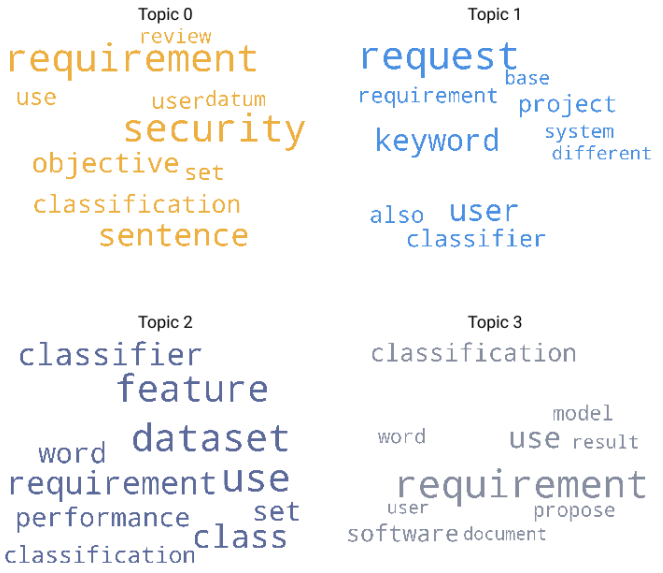

Fig. 5. Word cloud of the most frequent terms by topic

four machine learning algorithms to identify content related to business rules in different academic documents. In this regard, they implemented algorithms such as Minimum Sequential Optimization (SMO), Bayesian Networks, Random Forest, and Naïve Bayes. His approach required different experts to perform the manual classification of documents from different domains for the models' training. Later sections discuss training documents and evaluations.

On the other hand, Abad et al. [9] implement the J48 algorithm for decision trees in order to classify requirements sentences into Functional Requirements (FR) and Non-Functional Requirements (NFR). Plus, within the scope of non-functional requirements, they extended their classifications in 11 subcategories: Availability (A), Look & Feel (LF), Maintainability (MN), Operability (O), Performance (PE), Scalability (SC), Security (SE), Usability (US), Fault Tolerance (FT), and Portability (PO); (b) one constraint category: Legal & Licensing (L). It is important to note that these implementations required a natural language (NL) preprocessing procedure to fit the requirements' specific notation. The classification of NFR subcategories applied three classification approaches by topic modeling, clustering, and Naïve Bayes. In the classification by topic modeling, they implemented both the Latent Dirichlet Allocation (LDA) and Biterm Topic Model (BTM) algorithms, to establish relationships based on the occurrence of certain words.

Turning to the previous classifications, Baker et al. [10] apply machine learning techniques from a different perspective. Using the same database, with the addition of another one featured in a data challenge, a model was created to classify only five categories: maintainability, operability, performance, security, and usability. The approach implements two types of neural networks for this study, Artificial Neural Networks (ANN) with a hidden layer and a Convolutional Neural

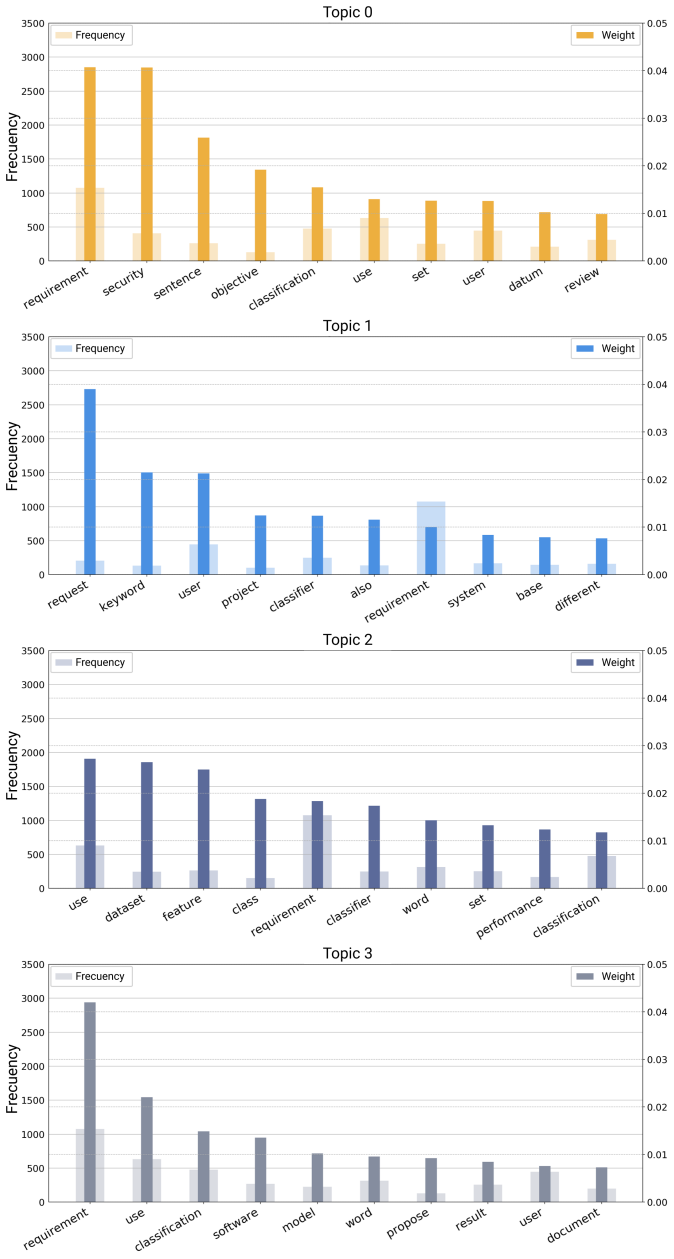

Fig. 6. Frequency and weight of the most common terms in the identified topics

Networks model trained through the bag-of-words natural language processing method.

Using the same dataset Dekhtyar and Fong [11] classified FR and NFR, as well as security-related requirements. However, different NL preprocessing methods were used for this study, such as Term Frequency-Inverse Document Frequency (TF-IDF) and word2vec. The models implemented were a Convolutional Neural Network and Naïve Bayes classifier.

With the same scope for security requirements, the study conducted by Riaz et al. [18] focused on identifying this kind of security-related documents. Within its classification, 6 security concerns are identified that are explored in require-

ments documentation: confidentiality, integrity, identification & authentication, availability, accountability, and privacy. Consequently, their approach implements NL processing by TF-IDF and classifies through a K-Nearest Neighbor model that is able to identify and suggest templates for the proper drafting of security requirements.

On the other hand, and related to classifiers applied to user reviews, Wang et al. [21] trained different machine learning models to test a hypothesis: if the implementation of app changelogs can improve the performance of the classifiers. Thus, their study focuses on classifying user reviews of different applications available in app stores, such as WhatsApp, iBooks, and TripAdvisor. They compared their classifiers' performance by training with and without this additional documentation. Given the nature of the type of comments that are published in reviews, their study only focuses on identifying 6 categories of requirements: Usability, Reliability, Portability, Performance, FR, and Others. Supported by NLP techniques such as stopwords removal, stemming and lemmatization, different classification models are trained: Naïve Bayes, Bagging, J48, and K Nearest Neighbor. Even though their experimental results do not show a considerable improvement in the classification, their study identifies the Naïve Bayes model as the most capable for future studies of this nature.

In general, Table VI shows the frequency of the ML algorithms identified in the bibliography found. In it, the Naïve Bayes, J48, and K Nearest Neighbor classifiers stand out as the most widely used for the classification of software requirements.

TABLE VI
FREQUENCY AND STUDIES OF THE IDENTIFIED MACHINE LEARNING ALGORITHMS

Model	Frequency	Studies
Naive Bayes	7	[9], [11], [15], [16], [19]–[21]
J48	4	[9], [13], [16], [21]
K Nearest Neighbor	3	[15], [18], [21]
Random Fores	2	[14], [19]
Convolutional Neural Networks	2	[10], [11]
Bagging	2	[16], [21]
SMO	1	[19]
Bayesian Network	1	[19]
K-means	1	[9]
LDA	1	[9]
BTM	1	[9]
Artificial Neural Networks	1	[10]
Adaptive Boost	1	[14]
Extra Tree	1	[14]
Gradient Boosting	1	[14]
Support Vector Machine	1	[15]
Stochastic Gradient Descent Classifier	1	[17]
Decision Tree	1	[20]

2) RQ2: What metrics have been used to measure the performance of those algorithms?: Regarding the means for the machine learning model evaluation, a consensus is identified on three specific metrics: Precision, Recall, and F-Score or F1. These metrics are related to model testing, comparing the actual result with the one predicted by the model. The Precision is described as:

$$Precision = \frac{TP}{TP + FP}$$

Being TP the True Positives, the times the model predicted the right answer, and FP the times that False Positives occurred, meaning a misclassification. However, this metric is calculated concerning a specific class. On the other hand, the Recall metric is expressed as:

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

Also known as *Sensitivity*, this metric shows the number of right classifications, compared with the misclassification in a different class. Finally, the F-Score describes the relation between Precision and Recall.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Although these metrics are calculated in all the studies identified, excluding the survey conducted in [12], the way they are compared to the original database differs. In some studies, a process of cross-validation is performed, where a series of subsamples or folds is extracted in order to split the dataset into training and test sets, the resulting metrics are the average metrics for each fold [4]. However, the split between training and test set can be done once and before the training process is performed, in some cases extracting a subsample from the original database, or comparing to a different source in others.

3) RQ3: What kind of software projects have been used to train those algorithms?: During the review, articles were identified that trained their classification models with related databases. Specifically, two databases were adopted by more than one study.

The case of the TERA-PROMISE database [33] consists of 625 functional (FR) and non-functional requirements (NFR) all of them written in requirement notation. Hence, each proposal that adopts it [9], [12], [13], [17], [20] implies text preprocessing. This database is also divided into subcategories of non-functional requirements: Availability (A), Look & Feel (LF), Maintainability (MN), Operability (O), Performance (PE), Scalability (SC), Security (SE), Usability (US), Fault Tolerance (FT), and Portability (PO); (b) one constraint category: Legal & Licensing (L). TERA-PROMISE is also referenced within the International Requirements Engineering Conference 2017 Data Challenge, among other databases. Thus different studies focused on their analysis, resulting in papers featured in the conference [9]–[11], [14].

Additionally, some studies carried out their training with information from available requirements documentation, such as Software Requirements Specifications (ERS) or domain-specific documents [18], [19]. On the other hand, the analysis of user reviews in search of requirements-related information

seemed to be a topic of interest, either from application stores [15], [20] or from crowdsourcing sites [15], [20]. In the case of these content extraction procedures, prior labeling was required to achieve the training of the classification model. It should be noted that studies were found that performed their training with more than one database.

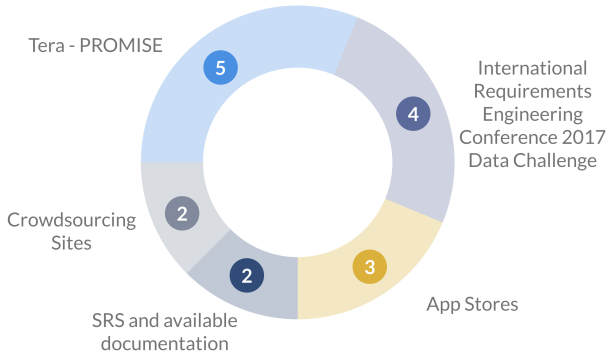

Fig. 7. Data sources used to train the models featured in the identified studies

4) RQ4: Which requirements categories are the most frequent to train those algorithms?: Once the target requirements in the studies have been identified, the comparison between them is shown in Figure 8. It is worth highlighting that the dichotomy between Functional and Non-Functional Requirements and the subcategories of the latter are the most recurring category-searches in the articles consulted. Something to note is that the search for this category of requirement is frequently focused on requirements associated to Quality Attributes. Although this classification was not carried out by searching all the attributes described in the ISO 25010 standard [34], papers were found that did so partially, or even only focused on classifying a specific quality attribute such as security [13], [18].

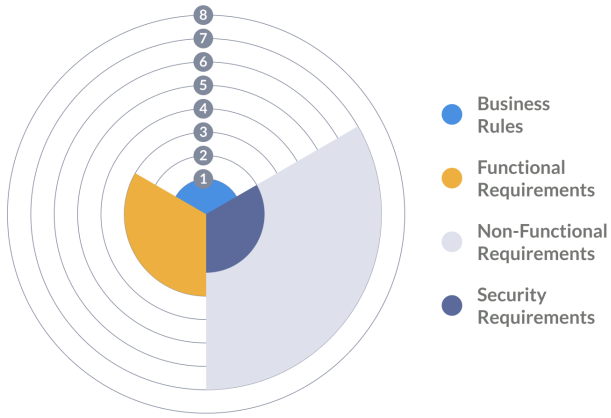

Fig. 8. Target requirements categories searched in the studied papers

In Figure 8, the entries counted as Functional Requirements are related to studies that focused only on classifying whether a document was associated to this category or not, without further analysis in the latter case.

VI. Discussion

Automatic software requirements classification represents an area of increasing interest for Requirement Engineering. Recent advances regarding the application of machine learning techniques to classify these requirements were identified in the present study. Based on the results obtained, areas of opportunity and topics of interest are identified concerning this matter. After reading the articles found, [22] is identified as one of the precursor studies on the subject, so it is proposed to extend the years of searching for future literature reviews, to increase the scope of studies. Concerning the spread of these applications, it was found that the most suitable venue for publishing these studies was the International Requirements Engineering Conference, where several studies were published.

Although several studies that address the use of machine learning in the Requirements Engineering area were found, this paper led to the realization that most of the studies regarding automated requirements classification are meant for academic purposes, despite the results found in practical implementations. Regarding the topics of interest, it was found that one of the applications of most comprehensive resonance is the classification of requirements applied to user reviews. Additionally, several Natural Language Processing strategies were found among the studies, this variation in approaches directly affects the performance of the classifiers. Another variable of resonance that affects the classification accuracy is whether the samples in the training are structured in requirements language or, as found in many other approaches, the samples are unstructured user reviews. Further analysis in the NLP strategies and syntaxis is needed to optimize this process.

VII. Conclusions

Motivated by the novel applications of machine learning in Requirements Engineering, this study aimed to identify the use of these techniques for classifying software requirements. It has been found that there are several proposals for this topic. Regarding the algorithms chosen, several of the studies addressed the use of models such as Naïve Bayes and J48 classifiers. Furthermore, a wide variety of machine learning algorithms were explored in the available literature for requirements classification, implying several preprocessing approaches for processing the text. The assessment of these models is focused on the relation between precision, recall, and F-Score. Nevertheless, the studies found are mostly focused on academic purposes, addressing this topic by using structured requirements databases, rather than natural language. As it was previously stated, an increasing interest in user reviews represents an area available for future research. Finally, it was concluded that most of the studies focus their efforts on classifying functional and non-functional requirements, especially requirements associated with quality attributes.

REFERENCES

[1] K. E. Wiegers and J. Beatty, *Software Requirements*, 3rd ed. Redmond, WA, USA: Microsoft Press, 2013.

[2] G. Kotonya and I. Sommerville, *Requirements Engineering: Processes and Techniques*, 1st ed. Wiley Publishing, 1998.

[3] H. Meth, M. Brhel, and A. Maedche, "The state of the art in automated requirements elicitation," *Information and Software Technology*, vol. 55, no. 10, pp. 1695–1709, oct 2013. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0950584913000827

[4] T. M. Mitchell, *Machine Learning*, 1st ed. New York, NY, USA: McGraw-Hill, Inc., 1997.

[5] "Machine Learning and Software Engineering," *Software Quality Journal*, vol. 11, no. 2, pp. 87–119, 2003. [Online]. Available: https://doi.org/10.1023/A:1023760326768

[6] M. Binkhonain and L. Zhao, "A review of machine learning algorithms for identification and classification of non-functional requirements," *Expert Systems with Applications: X*, vol. 1, p. 100001, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2590188519300010

[7] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," Department of Computer Science University of Durham, Durham, UK, Tech. Rep., 2007.

[8] M. Rodgers, A. Sowden, M. Petticrew, L. Arai, H. Roberts, N. Britten, and J. Popay, "Testing Methodological Guidance on the Conduct of Narrative Synthesis in Systematic Reviews: Effectiveness of Interventions to Promote Smoke Alarm Ownership and Function," *Evaluation*, vol. 15, no. 1, pp. 49–73, 2009. [Online]. Available: https://doi.org/10.1177/1356389008097871

[9] Z. S. H. Abad, O. Karras, P. Ghazi, M. Glinz, G. Ruhe, and K. Schneider, "What Works Better? A Study of Classifying Requirements," in *2017 IEEE 25th International Requirements Engineering Conference (RE)*, 2017, pp. 496–501.

[10] C. Baker, L. Deng, S. Chakraborty, and J. Dehlinger, "Automatic Multi-class Non-Functional Software Requirements Classification Using Neural Networks," in *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, vol. 2, jul 2019, pp. 610–615.

[11] A. Dekhtyar and V. Fong, "RE Data Challenge: Requirements Identification with Word2Vec and TensorFlow," in *2017 IEEE 25th International Requirements Engineering Conference (RE)*, 2017, pp. 484–489.

[12] T. Iqbal, P. Elahidoost, and L. Lucio, "A Bird's Eye View on Requirements Engineering and Machine Learning," in *Proceedings - Asia-Pacific Software Engineering Conference, APSEC*, vol. 2018-Decem, dec 2019, pp. 11–20.

[13] R. Jindal, R. Malhotra, and A. Jain, "Automated classification of security requirements," in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2016, pp. 2027–2033.

[14] Z. Kurtanović and W. Maalej, "Automatically Classifying Functional and Non-functional Requirements Using Supervised Machine Learning," in *2017 IEEE 25th International Requirements Engineering Conference (RE)*, 2017, pp. 490–495.

[15] C. Li, L. Huang, J. Ge, B. Luo, and V. Ng, "Automatically classifying user requests in crowdsourcing requirements engineering," *Journal of Systems and Software*, vol. 138, pp. 108–123, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0164121217303096

[16] M. Lu and P. Liang, "Automatic Classification of Non-Functional Requirements from Augmented App User Reviews," in *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*, ser. EASE'17. New York, NY, USA: ACM, 2017, pp. 344–353. [Online]. Available: http://doi.acm.org/10.1145/3084226.3084241

[17] M. Marinho, D. Arruda, F. Wanderley, and A. Lins, "A Systematic Approach of Dataset Definition for a Supervised Machine Learning Using NFR Framework," in *2018 11th International Conference on the Quality of Information and Communications Technology (QUATIC)*, 2018, pp. 110–118.

[18] M. Riaz, J. King, J. Slankas, and L. Williams, "Hidden in plain sight: Automatically identifying security requirements from natural language artifacts," in *2014 IEEE 22nd International Requirements Engineering Conference (RE)*, aug 2014, pp. 183–192.

[19] R. Sharma, J. Bhatia, and K. K. Biswas, "Automated identification of business rules in requirements documents," in *2014 IEEE International Advance Computing Conference (IACC)*, feb 2014, pp. 1442–1447.

[20] S. Taj, Q. Arain, I. Memon, and A. Zubedi, "To Apply Data Mining for Classification of Crowd Sourced Software Requirements," in *Proceedings of the 2019 8th International Conference on Software and Information Engineering*, ser. ICSIE '19. New York, NY, USA: ACM, 2019, pp. 42–46. [Online]. Available: http://doi.acm.org/10.1145/3328833.3328837

[21] C. Wang, F. Zhang, P. Liang, M. Daneva, and M. van Sinderen, "Can App Changelogs Improve Requirements Classification from App Reviews?: An Exploratory Study," in *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ser. ESEM '18. New York, NY, USA: ACM, 2018, pp. 43:1—-43:4. [Online]. Available: http://doi.acm.org/10.1145/3239235.3267428

[22] J. Cleland-Huang, R. Settimi, Xuchang Zou, and P. Solc, "The Detection and Classification of Non-Functional Requirements with Application to Early Aspects," in *14th IEEE International Requirements Engineering Conference (RE'06)*. IEEE, sep 2006, pp. 39–48. [Online]. Available: http://ieeexplore.ieee.org/document/1704047/

[23] J. Cleland-Huang, R. Settimi, X. Zou, and P. Solc, "Automated classification of non-functional requirements," *Requirements Engineering*, vol. 12, no. 2, pp. 103–120, may 2007. [Online]. Available: http://link.springer.com/10.1007/s00766-007-0045-1

[24] J. Slankas and L. Williams, "Automated extraction of non-functional requirements in available documentation," in *2013 1st International Workshop on Natural Language Analysis in Software Engineering (NaturaLiSE)*. IEEE, may 2013, pp. 9–16. [Online]. Available: http://ieeexplore.ieee.org/document/6611715/

[25] M. Glinz, "On Non-Functional Requirements," in *15th IEEE International Requirements Engineering Conference (RE 2007)*. IEEE, oct 2007, pp. 21–26. [Online]. Available: http://ieeexplore.ieee.org/document/4384163/

[26] W. Zhang, Y. Yang, Q. Wang, and F. Shu, "An Empirical Study on Classification of Non-Functional Requirements," 2011.

[27] E. Knauss, D. Damian, G. Poo-Caamano, and J. Cleland-Huang, "Detecting and classifying patterns of requirements clarifications," in *2012 20th IEEE International Requirements Engineering Conference (RE)*. IEEE, sep 2012, pp. 251–260. [Online]. Available: http://ieeexplore.ieee.org/document/6345811/

[28] W. Maalej, Z. Kurtanović, H. Nabil, and C. Stanik, "On the automatic classification of app reviews," *Requirements Engineering*, vol. 21, no. 3, pp. 311–331, sep 2016. [Online]. Available: http://link.springer.com/10.1007/s00766-016-0251-9

[29] H. Yang and P. Liang, "Identification and Classification of Requirements from App User Reviews," jul 2015, pp. 7–12. [Online]. Available: http://ksiresearchorg.ipage.com/seke/seke15paper/seke15paper_63.pdf

[30] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, may 2010, pp. 45–50.

[31] W. Cui, Y. Wu, S. Liu, F. Wei, M. X. Zhou, and H. Qu, "Context preserving dynamic word cloud visualization," in *2010 IEEE Pacific Visualization Symposium (PacificVis)*. IEEE, mar 2010, pp. 121–128. [Online]. Available: http://ieeexplore.ieee.org/document/5429600/

[32] L. Sun and Y. Yin, "Discovering themes and trends in transportation research using topic modeling," *Transportation Research Part C: Emerging Technologies*, vol. 77, pp. 49–66, apr 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0968090X17300207

[33] J. Sayyad Shirabad and T. J. Menzies, "The PROMISE Repository of Software Engineering Databases." School of Information Technology and Engineering, University of Ottawa, Canada, 2005. [Online]. Available: http://promise.site.uottawa.ca/SERepository

[34] International Organization for Standardization, "ISO 25010 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models," International Organization for Standardization, Standard, 2011.