Applications of Machine Learning in Requirements Traceability: A Systematic Mapping Study

Xingfu Li[1], Bangchao Wang [1,2,*], Hongyan Wan[1,2], Yang Deng[1], Zihan Wang[3]
[1]*School of Computer Science and Artificial Intelligence, Wuhan Textile University*, Wuhan, China
[2]*Engineering Research Center of Hubei Province for Clothing Information, Wuhan Textile University*, Wuhan, China
[3]*School of Mathematical & Physical Science, Wuhan Textile University*, Wuhan, China
lixingfu1999@163.com, wangbc@whu.edu.cn, hywan@wtu.edu.cn, dd0028y@163.com, wangzh011031@163.com

***Abstract*—Requirements traceability (RT) is crucial for requirement management and impact analysis of requirement change in software development. The applications of machine learning (ML) technologies to RT have received much attention. In this paper, we aim to provide the state-of-the-art progress of the studies on the intersection of ML and RT. A systematic mapping study (SMS) is conducted and 26 studies have been identified as primary studies. The results present 32 ML technologies and 7 enhancement strategies for establishing trace links. Besides, 46 datasets are utilized for validating the performance of these ML technologies. Additionally, the overall quality of these primary studies is at a good level. This study indicates that numerous studies have proved the potential of utilizing ML technologies for predicting emerging trace links in RT by utilizing existing traceability information. Moreover, open-source datasets are the most popular, which greatly improves the reproducibility of studies. However, there is still a gap between academia and industrial application because of the lack of industrial practice and guidance from practitioners.**

Keywords—requirements traceability, machine learning, quality assessment, systematic mapping study.

I. Introduction

Requirements traceability (RT) is “ the ability to describe and follow the life of a requirement, in both a forwards and backwards direction” [1]. It is one of the vital activities in Requirements engineering (RE). It is beneficial for software development that RT helps to identify the origin of requirements, analyze the impact of requirements change and trace the relationships between requirements and other artifacts [2]. This ensures the needs of stakeholders are always met in development process, and promotes the transparency and traceability of the software development process.

There are various technologies such as information retrieval (IR) and machine learning (ML) approaches. However, it cannot use historical information for prediction, which leads to poor universality. Especially some informal products still require a lot of manual intervention. After 2017, increasing ML technologies are developed and applied to RT to obtain complete requirements trace links. It can learn from existing traceability information to obtain characteristics. The applications of ML in RT approaches have been emerging as a hot topic with the efforts of researchers and practitioners. However, there has been no systematic literature review about the status of ML-based RT approaches over the past ten years. It prompts us to summarize the progress of ML-based RT.

In this paper, we provide evidence-based insights of the intersection of ML and RT. We hope that it can help researchers and practitioners better understand ML-based tracing approaches and extend them through the novel study.

This paper is structured into six main sections: Section I provides an introduction to the background. Section II introduces the related work and research questions (RQs). Section III and Section IV provide the execution process of the SMS method and the findings, respectively. Section V discusses the validity threat and potential research directions. At last, the conclusion is summarized in Section VI.

II. Related Work

As far as we know, there are some relevant reviews about RT. Wang et al. provide a review of the intersection of RT and IR. They have summarized 21 enhancement strategies that can improve the performance of four kinds of IR technologies [3]. Torkar et al. have concluded available tools and galore techniques for RT [4]. However, they don't pay close attention to the state-of-the-art progress of ML-based RT approaches. Tufail et al. have performed a systematic review and identified seven kinds of RT models, ten challenges, and fourteen tools [2]. However, they don't summarize the specific technologies to guide researchers.

Wang et al. have conducted a systematic literature review and discussed challenges associated with RT activities [5]. They also summarize RT technologies that include ML technologies and evaluate the overall quality of primary studies. It is noted that the time interval of their work is 2006–2016, which indicates that more novel technologies haven't been covered. The ML-based SE approaches have been reviewed by Mezouar et al. [6]. However, their work doesn't focus on the RT field.

However, there has been no review about the application of ML in RT field in the past decade. It is desired to comprehensively analyze these ML techniques to understand emerging study directions in RT. The aim of this paper is to systematically investigate ML-based tracing approaches over the last decade. Besides, the overall degree of quality is quantified to provide a reference for researchers. In summary, this study has addressed four RQs to provide the current progress of the studies on the intersection of ML and RT:

RQ1: What are the publication times and venues of primary studies?

RQ2: Which ML approaches and strategies are applied to RT?

RQ3: Which datasets are utilized for ML-based RT approaches?

RQ4: What is the overall quality of primary studies?

III. Research Method

This SMS has followed the guidance provided by [7]. In this section, a detailed search strategy, classification process and quality assessment method are defined to conduct SMS.

* Corresponding author: Bangchao Wang.

DOI reference number: 10.18293/SEKE2023-135

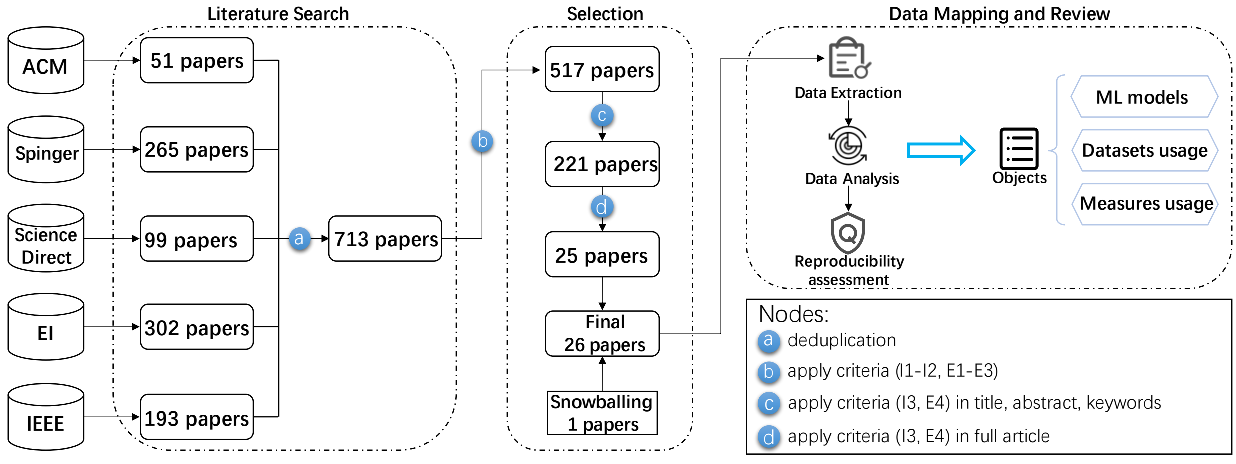


Fig. 1. Search Process and Results

A. *Search Strategy*

In this section, the search terms and databases are described for searching. Besides, selection criteria are proposed to screen primary studies. The whole search process is shown in Fig. 1.

1) Search Scope and Terms

Five common databases are selected as follows: ACM Digital Library, IEEE Xplorer, Science Direct, SpringerLink and EI Compendex. These databases cover a wide range of studies related to ML and RT. In addition, this SMS is conducted based on relevant literature from the past ten years, with a search period from January 2013 to December 2022.

According to the four RQs of this study, the based search terms are defined as "requirements traceability" and "machine learning". In order to refine these search terms, alternative and related terms in the RT field are used to iteratively retrieve 5 databases according to the PICO standard [3]. The finalized search terms are as follows:

Population: requirements traceability, requirements trace, requirements tracing, requirements traceability recovery.

Intervention: machine learning, ML, supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning.

After determining search terms, logical operators (i.e., **OR** and **AND**) can be used to connect strings to build more complex search queries. The search string is formulated as follows: ("requirements traceability" OR "requirements trace" OR "requirements tracing" OR "requirements traceability recovery") AND ("machine learning" OR "ML" OR "supervised learning" OR "unsupervised learning" OR "semi-supervised learning" OR "reinforcement learning").

2) Study selection criteria

In this step, inclusion and exclusion criteria are designed to select primary studies that are related to this research. The following are the inclusion and exclusion selection criteria:

- **Inclusion selection criteria:**

I1: The study is published from January 2013 to December 2022.

I2: The study is written in English.

I3: The study with the more detailed description is selected when multiple authors have the same study.

- **Exclusion selection criteria:**

E1: The study is a review or gray literature.

E2: The study with less than 3 pages.

E3: Duplicate studies with the same authors.

E4: The study is not related to the application of ML in RT.

3) Study selection procedure

The Zotero tool is used to perform the search and selection process. Firstly, we utilize search strings constructed in III.A to retrieve 910 literatures from five digital libraries. Among them, EI Compendex contains nearly one-third of literature (302/910). Secondly, the Zotero tool is employed to remove duplicates resulting in 713 unique studies. Next, inclusion and exclusion criteria (I1-I2, E1-E3) are applied to perform a coarse-grained filter on these 713 studies. Subsequently, I3 and E4 criteria are employed to exclude irrelevant literatures based on their titles, abstract and keywords. A total of 221 relevant papers are obtained after this round of filtering. Finally, 25 literatures are selected by reading the full text. In addition, snowballing is performed to prevent the omission of related studies. Ultimately, 26 literatures are strictly selected as the primary studies.

B. *Data Classification and Analysis*

After determining 26 primary studies, the data items that need to be extracted from each primary study have been confirmed by discussing according to our goals. Afterwards, the first two authors extracted data from all the primary studies separately. We have compared these two extraction reports to identify controversial points. Finally, all authors have discussed and determined the final results through the seminar. The extraction report for consultation is generated in a Word file. Additionally, the technical report and detailed search process are available on the website (https://github.com/WTU-intelligent-software-development/ML-based-RT-SMS).

C. *Quality Assessment*

During the quality assessment phase, the quality of all primary studies has been evaluated. Table I presents that these primary studies are rigorously assessed based on four dimensions selected according to the technology transfer model [8]. Each evaluation dimension is divided into different levels of evidence for precise quantification, and Table I provides a detailed description of these evidence levels and their corresponding scores. Then, the scores obtained from these evaluations are utilized to conduct a comprehensive analysis to reflect the extent to which these studies support technology transfer.

TABLE I QUALITY ASSESSMENT METHOD

Items	Question	Description
Research Method	What research method is used in the evaluation?	Level 1: Evidence obtained from demonstration or toy examples (0.2).
		Level 2: Evidence obtained from expert opinions or observations (0.4).
		Level 3: Evidence obtained from academic studies, such as controlled experiments (0.6).
		Level 4: Evidence obtained from industrial studies, such as case study (0.8).
		Level 5: Evidence obtained from industrial practice (1.0).
Context	In what context is the tracing technology validated?	Level 1: Evaluation conducted in academic context (0.6).
		Level 2: Evaluation conducted in industry context (1.0).
Subjects	What subjects are used in the evaluation?	Level 1: Evaluation conducted by student (0.6).
		Level 2: Evaluation conducted by researcher (0.8).
		Level 3: Evaluation conducted by practitioner (1.0).
Dataset Source	Whether or not the primary study provides the dataset source?	Level 1: None of dataset sources are provided (0.6).
		Level 2: A portion of dataset sources are provided (0.8).
		Level 3: All dataset sources are provided (1.0).

The first evaluation dimension used for quality assessment (Research Method) is proposed by Alves et al. [9]. It is one of the key factors to ensure the credibility and effectiveness of research results. The correct selection and appropriate interpretation of research methods are one of the key factors to ensure the credibility and effectiveness of research results.

Context and Subjects proposed by Ivarsson et al. [10] are adopted as the second and third dimensions. Context includes industrial context and academic context. Validation in the academic environment provides support for industrial applications. The subject usually includes different roles such as students, researchers, and practitioners. Various subjects represent different research abilities and levels of experience.

In addition, providing a dataset source can increase the credibility of study and provide more opportunities for other researchers to expand their research. Therefore, the source of the dataset has been added as the final evaluation dimension. Finally, the evaluation method proposed by Wang et al. [5] has been utilized to assign scores to each evidence level.

IV. Results

A. *RQ1: What are the publication times and venues of primary studies?*

Fig. 2 shows the number of studies published per year in this field and the distribution of venues from 2013 to 2022. A total of 26 primary studies are published, covering 7 types of journals and 10 types of conferences. We can see that studies have become plentiful since 2017. Requirement Engineering Conference (RE Conference) and International Conference on Software Engineering (ICSE) are the venues with the most publications. It should be noted that the underlined venues are conferences and their names are abbreviations due to the space limit. The full names of venues are recorded in the technical reports if readers are interested.

B. *RQ2: Which ML approaches and strategies are applied to RT?*

The illustration of the general process of ML-based RT approaches is shown in Fig. 3. It is usually divided into three stages: preprocessing stage, link generation stage, and link refinement stage. The preprocessing stage includes the data preprocessing process and the generation of the feature vector process. The main task of the link generation stage is to filter out candidate links. The link refinement stage optimizes the candidate links generated in the link generation stage.

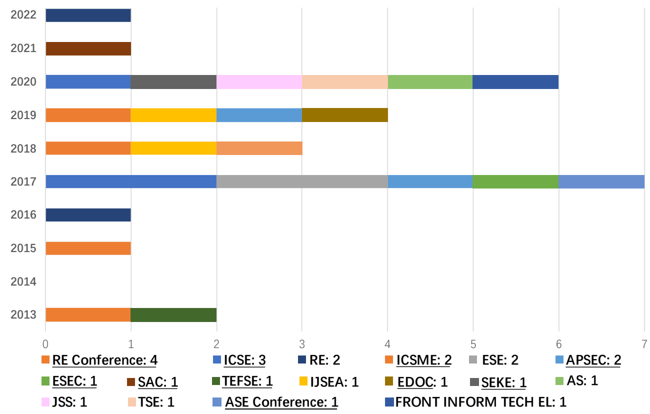


Fig. 2. Number of Studies and Distribution of Venues Per Year

Table II shows the stages in which ML technologies are applied and their frequency of utilization. 32 ML technologies have been used 76 times. Fig. 3 shows the enhancement strategies applied in each stage. Researchers focus on link classification and representation learning strategies in 7 enhancement strategies.

Random Forest, Decision Tree, and Naive Bayes are the most frequently used ML models in the link generation stage. On the one hand, the time cost of calculating the similarity between each pair of artifacts is reduced. On the other hand, it abandons the similarity calculation and transforms it into classification tasks to obtain more complete trace links. Word2vec and doc2vec are also utilized frequently in the preprocessing stage. They are trained by context to bridge the semantic gap. It is noted that three-fourths of ML models have been used less than three times. Many ML technologies, such as deep learning models, are limited by the scale of the dataset. Besides, the increased complexity of the applications of some technologies for RT is a potential reason.

C. *RQ3: Which datasets are utilized for ML-based RT approaches?*

The datasets used in each primary study are summarized to analyze datasets used to verify ML-based RT approaches. Table III shows the specific information, source links, frequency of use, and primary studies of these datasets. There are 46 datasets that have been used 100 times in total. Datasets that have a source link and are used more than four times are chosen to display because of space limitations. If you need more detailed information, please see the technical report

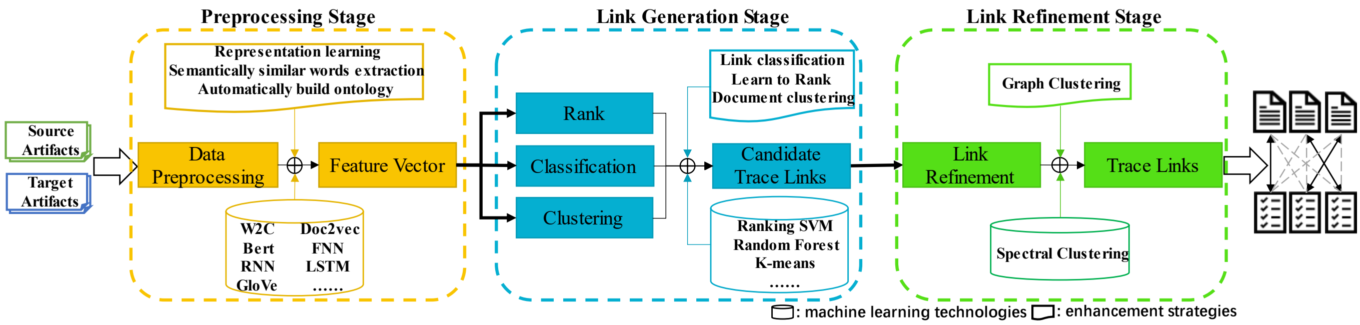


Fig. 3. The General Process of ML-based Requirements Traceability

TABLE II. LIST OF ML TECHNOLOGIES AND APPLYING STAGE FOR REQUIREMENTS TRACEABILITY

ML Technologies	Strategies	Applying Stage	Freq.
Random Forest	Link Classification [11][13][14][15][16][17][18][19]	G	8
Decision Tree	Link Classification [11][14][16][17][19][20][21][22]	G	8
Naive Bayes	Link Classification [13][14][16][17][19][20][21][22]	G	8
K Nearest Neighbor (KNN)	Link Classification [11][13][14][17][19][21]	G	6
Word2vec	Representation Learning [12][23][24][25][26]	P	5
Paragraph Vector (Doc2vec)	Representation Learning [11][25][27]	P	3
Support Vector Machine (SVM)	Link Classification [13][20][28]	G	3
Feedforward Neural Network (FNN)	Semantically Similar Words Extraction[24], Link Classification [29]	P, G	2
Gradient Boosting Decision Tree (GBDT)	Link Classification [11][20]	G	2
Logistic Regression	Link Classification [13][16]	G	2
Hierarchical Agglomerative Clustering (HAC)	Semantically Similar Words Extraction [30][31]	P	2
Recurrent Neural Networks (RNN)	Representation Learning [26], Link Classification [29]	P, G	2
Active Learning (AL)	Link Classification [18][32]	G	2
Bagging	Link Classification [16][21]	G	2
Bidirectional Encoder Representation from Transformers (Bert)	Representation Learning [27]	P	1
Long Short-Term Memory (LSTM)	Representation Learning [26]	P	1
Bi-directional Gated Recurrent Unit (Bi-GRU)	Representation Learning [26]	P	1
Fasttext	Representation Learning [27]	P	1
Global Vectors (GloVe)	Representation Learning [23]	P	1
K-medoids	Semantically Similar Words Extraction [31]	P	1
Single Link Clustering	Automatically Build Ontology [28]	P	1
Bi-directional Long Short-Term Memory (Bi-LSTM)	Representation Learning [26]	P	1
Gated Recurrent Unit (GRU)	Representation Learning [26]	P	1
Universal Sentence Encoder (USE)	Representation Learning [27]	P	1
K-means	Documents Clustering [33]	G	1
Logit Boost	Link Classification [21]	G	1
Hierarchical Bayesian Network	Link Classification [34]	G	1
Label Spreading	Link Classification [23]	G	1
Ranking SVM	Learn to Rank [12]	G	1
RankBoost	Learn to Rank [29]	G	1
Reinforcement Learning (RL)	Link Classification [35]	G	1
Spectral Clustering	Graph Clustering [36]	R	1

Note: "P" represents Preprocessing Stage, "G" represents Links Generation Stage, "R" represents Links Refinement Stage.

TABLE III. DATASETS AND INFORMATION FROM PRIMARY STUDIES

Dataset Name	Source Artifacts	Target Artifacts	True link	Source Link	Freq.	Primary studies
eTour	Use Case	Code	366	http://www.coest.org	12	[11][12][13][14][15][18][19][20][21][25][32][34]
EasyClinic	Use Case	Code	93		9	[12][13][14][18][19][21][25][32][36]
		Interaction Diagram	26			
		Use Case	53			
	Test Case	Use Case	93			
iTrust	Use Case	Code	534		9	[12][13][14][17][18][19][22][25][34]
	Requirement	Code	535			
SMOS	Use Case	Code	1045		8	[11][13][14][15][18][19][32][34]
CM-1	High-level requirement	Low-level design document	Unclear		7	[12][14][19][24][25][35][36]
	High-level requirement	Low-level requirement	45			
	Requirement	Design	46			
	Requirement	Use Case	Unclear			
eAnci	Use Case	Code	567		7	[11][13][14][15][18][19][32]
MODIS	High-level requirement	Low-level requirement	41	http://promise.site.uottawa.ca/SERepository	4	[12][18][24][32]

Note1: There are 23 open-source datasets with only one usage frequency, i.e., Albergate, GANNT, CCHIT, EBT, LibEST, Selex SI, Chess, Gantt, JHotDraw, ARC-IT, DASHBUILDER, JBTM, Accumulo, Ignite, Isis, Tika, Care2x, ClearHealth, Physician, Trial Implementations, PatientOS, PracticeOne, WorldVistA.

Note2: There are 16 datasets without the source link, i.e., Pine, Drools, Lucene, PTC, Waterloo, AIRFLOW, ANY23, IMMUTANT, CAF and 7 unnamed datasets.

mentioned in Section III.B.

Nearly 65% (30/46) of datasets that are used in primary studies are open-source datasets. These open-source datasets have been used 81 times (81%, 81/100). Besides, less than 27% (7/26) of primary studies don't fully use open-source datasets. More importantly, the majority (59/81) of all open-source datasets are provided by CoEST. Moreover, eTour, EasyClinic, and iTrust are the three most popular open-source datasets.

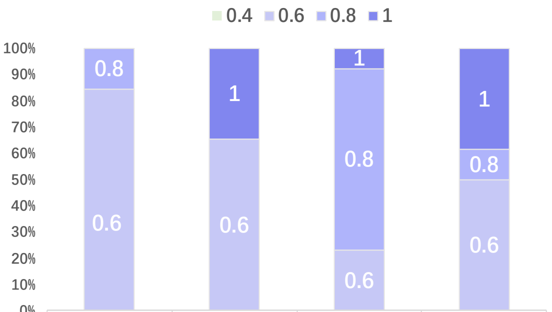
D. RQ4: What is the overall quality of primary studies?

Fig. 4 has summarized the overall quality of each assessment dimension. Fig. 5 uses box diagrams to analyze the overall quality of all primary studies comprehensively. The following information could be obtained by analyzing the data listed in Fig. 4 and Fig. 5:

a) The vast majority of studies (around 85%) use the experiment research method, while only 15% (4/26) of primary studies the use case study research method. It indicates that most studies lack validation in real industrial environments, which hinders researchers from understanding more complex phenomena.

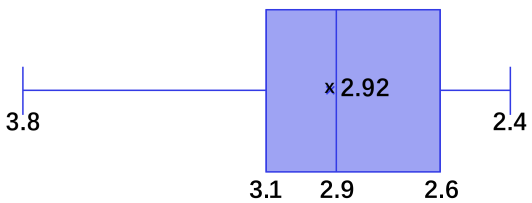
b) More than half of primary studies (17/26, around 65%) are validated in the academic context. According to the technology transfer model [8], these studies have a large gap from actual industrial applications, and industrial validation (such as static and dynamic validation) is still needed before the final solution can be released.

c) Only two primary studies subjects are practitioners. It indicates that most studies lack guidance from practitioners' practical and professional industry knowledge. Therefore, it is suggested that more practitioners can participate in evaluation and validation.



	Research Method	Context	Subject	Dataset Source
1	0	9	2	10
0.8	4	0	18	3
0.6	22	17	6	13
0.4	0	0	0	0
Avg.	0.63	0.74	0.78	0.78

Fig. 4. The Distribution of Quality Score of Each Assessment Dimensions



Poor (0-1.99)	Middle (2.0-2.59)	Good (2.6-3.19)	Excellent (3.2-4.0)
0	2	18	6
0%	8%	69%	23%

Fig. 5. Overall Quality Assessment Scores of All Primary Studies

d) Half of the studies don't provide a dataset source. The lack of datasets source not only affects reproducibility but also reduces the transparency and credibility of the study. It is noted that the four studies [11][19][24][30] not provide dataset sources because of confidentiality.

We have calculated the sum of the quality scores for each primary study. It should be noted that each quality assessment criterion has a maximum score of 1, and the full quality score for each study is 4. The quality levels are divided into four categories (poor: 0-1.99, middle: 2.0-2.59, good: 2.6-3.19, excellent: 3.2-4) [5][10]. From Fig. 5, the quality scores are mainly concentrated between 2.6 and 3.1. Moreover, the median and mean scores suggest that the overall quality of primary studies is at a "good" level. This proves that this SMS is trustworthy.

V. DISCUSSION

A. Validity Threats Discussion

The threats that influence processes and the findings of this SMS are introduced in this section. They are divided into internal validity, external validity, conclusion validity, and construction validity. Four threats are introduced as follows:

Internal validity: To avoid the risk of internal threats, data extraction and classification should be carefully conducted to ensure their accuracy. Therefore, in this study, the first two authors collaborated closely and conducted repeated reviews to minimize potential errors.

External validity: External validity concerns the generality of the conclusions of our SMS. For instance, whether the primary studies can represent the research questions of the review topic. In order to mitigate the threat, multiple search processes have been conducted and the search terms were constantly adjusted to improve the coverage.

Conclusion validity: The literature may be excluded wrongly. Therefore, the first two authors simultaneously conducted the screening process and determined the final primary study through discussion among the entire authors.

Construction validity: Construction validity is concerned with the deviation of the studied concept from the studied topic. The based search terms are "requirements traceability" and "machine learning" To reduce the impact, synonyms or alternative terms are used for searching to maximize coverage.

B. Future Research Directions

This SMS has illuminated several future research directions with great potential for further exploration in RT.

Firstly, the mainstream ML models applied to RT are traditional models such as Random Forest, Decision Tree, and Naive Bayes at present. With the development of ML or deep learning, the applications of novel technologies to RT are a potential future research direction.

Secondly, due to the ability of different technologies to handle different features, mining a combination of multiple machine learning techniques may improve the accuracy of feature based link recognition.

Finally, the utilization of many types of features may lead to feature redundancy and even have a counterproductive effect. Exploring a suitable feature selection approach to improve the performance of models is worth study.

VI. CONCLUSION

This SMS presents that ML techniques are playing a role in RT and researchers pay more attention to applying ML to

RT to obtain accurate and complete trace links. The following conclusions can be drawn:

1) Seven enhancement strategies have been conducted to support the establishment of trace links. ML models can learn from existing traceability information to predict new links. Moreover, link classification has attracted much attention to distinguish whether artifacts have trace links.

2) Open-source datasets are more popular than closed-source datasets. Besides, CoEST is the most popular source of datasets. In order to make the study more reproducible, researchers are suggested to use open-source datasets.

3) The overall quality is at a good level. This indicates that the research level in this field is relatively good, and the selected literature is representative and reliable. However, it is also noticed that the research method lowers the overall quality. It is suggested that researchers can verify and evaluate in the realistic industrial environment.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China Project (No. 62102291), and the Opening Foundation of Engineering Research Center of Hubei Province for Clothing Information (No. 2022HBCI02, No. 2022HBCI05).

REFERENCES

[1] O. C. Z. Gotel, C. W. Finkelstein, “An Analysis of the Requirements Traceability Problem,” in Proceedings of IEEE International Conference on Requirements Engineering, 1994, pp. 94-101.

[2] R. Torkar, T. Gorschek, R. Feldt, M. Svahnberg, U. A. Raja, K. Kamran, “Requirements traceability: a systematic review and industry case study,” International Journal of Software Engineering and Knowledge Engineering, vol. 22, no. 3, pp. 385-433, 2012.

[3] B. Wang, H. Wang, R. Luo, et al., “A Systematic Mapping Study of Information Retrieval Approaches Applied to Requirements Trace Recovery,” in 34th International Conference on Software Engineering and Knowledge Engineering, 2022, pp. 1–6.

[4] R. Torkar, T. Gorschek, R. Feldt, et al., “Requirements Traceability: a Systematic Review and Industry Case Study,” Int. J. Softw. Eng. Knowl. Eng., vol. 22, no. 3, 2012.

[5] B. Wang, R. Peng, Y. Li, et al., “Requirements traceability technologies and technology transfer decision support: A systematic review,” J. Syst. Softw., vol. 146, pp. 59–79, 2018.

[6] H. Mezouar and A. E. Afia, “A Systematic Literature Review of Machine Learning Applications in Software Engineering,” in Proceedings of the 5th International Conference on Big Data and Internet of Things, 2022, pp. 317–331.

[7] K. Petersen, S. Vakkalanka, and L. Kuzniarz, “Guidelines for conducting systematic mapping studies in software engineering: An update,” Inf. Softw. Technol., vol. 64, pp. 1–18, 2015.

[8] T. Gorschek, P. Garre, S. Larsson, and C. Wohlin, “A model for technology transfer in practice,” IEEE Softw., vol. 23, no. 6, pp. 88–95, 2006.

[9] V. Alves, N. Niu, C. Alves, and G. Valença, “Requirements engineering for software product lines: A systematic literature review,” Inf. Softw. Technol., vol. 52, no. 8, pp. 806–820, 2010.

[10] M. Ivarsson and T. Gorschek, “Technology transfer decision support in requirements engineering research: a systematic review of REj,” Requir. Eng., vol. 14, no. 3, pp. 155–175, 2009.

[11] T. Li, S. Wang, D. Lillis, and Z. Yang, “Combining Machine Learning and Logical Reasoning to Improve Requirements Traceability Recovery,” Applied Sciences, vol. 10, no. 20, pp. 7253, 2020.

[12] T. Zhao, Q. Cao, and Q. Sun, “An Improved Approach to Traceability Recovery Based on Word Embeddings,” in 2017 24th Asia-Pacific Software Engineering Conference (APSEC), 2017, pp. 81–89.

[13] C. Mills, J. Escobar-Avila, and S. Haiduc, "Automatic traceability maintenance via machine learning classification." in 2018 IEEE International Conference on Software Maintenance and Evolution (ICSME), 2018, pp. 369-380.

[14] C. Mills, “Automating traceability link recovery through classification,” in Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering, 2017, pp. 1068–1070.

[15] C. Mills and S. Haiduc, “A Machine Learning Approach for Determining the Validity of Traceability Links,” in 2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C), 2017, pp. 121–123.

[16] D. Falessi, J. Roll, J. Guo, and J. Cleland-Huang, "Leveraging historical associations between requirements and source code to identify impacted classes." IEEE Trans. Software Eng., vol. 46, no. 4, pp. 420-441, 2018.

[17] M. Hammoudi, C. Mayr-Dorn, A. Mashkoor, and A. Egyed, “On the effect of incompleteness to check requirement-to-method traces,” in Proceedings of the 36th Annual ACM Symposium on Applied Computing, 2021, pp. 1465–1474.

[18] C. Mills, J. Escobar-Avila, A. Bhattacharya, et al., “Tracing with Less Data: Active Learning for Classification-Based Traceability Link Recovery,” in 2019 IEEE International Conference on Software Maintenance and Evolution (ICSME), 2019, pp. 103–113.

[19] C. Mills, “Towards the automatic classification of traceability links,” in 2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE), 2017, pp. 1018–1021.

[20] G. Zhao, T. Li, and Z. Yang, “An Extended Knowledge Representation Learning Approach for Context-based Traceability Link Recovery,” in 32nd International Conference on Software Engineering and Knowledge Engineering, 2020, pp. 77–82.

[21] D. Falessi, M. Di Penta, G. Canfora, and G. Cantone, “Estimating the number of remaining links in traceability recovery,” Empir. Software Eng., vol. 22, no. 3, pp. 996–1027, 2017.

[22] J. Guo, M. Gibiec, and J. Cleland-Huang, “Tackling the term-mismatch problem in automated trace retrieval,” Empir. Software Eng., vol. 22, no. 3, pp. 1103–1142, 2017.

[23] E. Effa Bella, S. Creff, M. P. Gervais, and R. Bendraou, “ATLaS: A Framework for Traceability Links Recovery Combining Information Retrieval and Semi-Supervised Techniques,” in 2019 IEEE 23rd International Enterprise Distributed Object Computing Conference (EDOC), 2019, pp. 161–170.

[24] W. Wang, N. Niu, H. Liu, and Z. Niu, “Enhancing Automated Requirements Traceability by Resolving Polysemy,” in 2018 IEEE 26th International Requirements Engineering Conference (RE), 2018, pp. 40–51.

[25] L. Chen, D. Wang, J. Wang, and Q. Wang, “Enhancing Unsupervised Requirements Traceability with Sequential Semantics,” in 2019 26th Asia-Pacific Software Engineering Conference, 2019, pp. 23–30.

[26] J. Guo, J. Cheng, and J. Cleland-Huang, “Semantically Enhanced Software Traceability Using Deep Learning Techniques,” in 2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE), 2017, pp. 3–14.

[27] M. Abbas, A. Ferrari, A. Shatnawi, et al., “On the relationship between similar requirements and similar software: A case study in the railway domain,” Requir. Eng., pp. 1-25, 2022.

[28] Z. Li and L. Huang, “Tracing Requirements as a Problem of Machine Learning,” IJSEA, vol. 9, no. 4, pp. 21–36, 2018.

[29] A. C. Marcen, R. Lapena, Ó. Pastor, and C. Cetina, “Traceability Link Recovery between Requirements and Models using an Evolutionary Algorithm Guided by a Learning to Rank Algorithm: Train control and management case,” J. Syst. Softw., vol. 163, pp. 110519, 2020.

[30] A. Mahmoud, “An information theoretic approach for extracting and tracing non-functional requirements,” in 2015 IEEE 23rd International Requirements Engineering Conference (RE), 2015, pp. 36–45.

[31] A. Mahmoud and G. Williams, “Detecting, classifying, and tracing non-functional software requirements,” Requir. Eng., vol. 21, no. 3, pp. 357–381, 2016.

[32] T. Du, G. Shen, Z. Huang, et al.,“Automatic traceability link recovery via active learning,” Frontiers Inf. Technol. Electron. Eng., vol. 21, no. 8, pp. 1217–1225, 2020.

[33] M. Mezghani, J. Kang, E. B. Kang, and F. Sedes, “Clustering for Traceability Managing in System Specifications,” in 2019 IEEE 27th International Requirements Engineering Conference, 2019.

[34] K. Moran, David N. Palacio, Carlos Bernal-Cardenas, et al., “Improving the effectiveness of traceability link recovery using hierarchical bayesian networks,” in 2020 42nd International Conference on Software Engineering, 2020, pp. 873–885.

[35] H. Sultanov and J. H. Hayes, “Application of reinforcement learning to requirements engineering: requirements tracing,” in 2013 21st IEEE International Requirements Engineering Conference, 2013, pp. 52–61.

[36] P. Rempel, P. Mader, and T. Kuschke, “Towards feature-aware retrieval of refinement traces,” in 2013 7th International Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE), 2013, pp. 100–104.