Exploring the Advancements in Bug Traceability Over the Past Decade: A Systematic Mapping Study

1. Data Extraction for Research Questions

1.1RQ1: What are the times/venues of the studies?

Index	Title	Author	Year	Venue	Publication
S1 [8]	FRLink: Improving the recovery of missing issue-commit links by revisiting file relevance	Yan Sun Qing Wang Ye Yang	2017	Information and Software Technology （IST）	Journal
S2 [9]	Automatically Matching Bug Reports With Related App Reviews	Mario Haering Christoph Stanik Walid Maalej	2021	International Conference on Software Engineering （ICSE）	Conference
S3 [10]	Analyzing Requirements and Traceability Information to Improve Bug Localization	Michael Rath David Lo Patrick Mäder	2018	Mining Software Repositories （MSR）	Conference
S4 [11]	Automated Recovery of Issue-Commit Links Leveraging Both Textual and Non-textual Data	Pooya Rostami Mazrae Maliheh Izadi Abbas Heydarnoori	2021	International Conference on Software Maintenance and Evolution (ICSME)	Conference
S5 [12]	BTLink : automatic link recovery between issues and commits based on pre-trained BERT model	Jinpeng Lan Lina Gong Jingxuan Zhang Haoxiang Zhang	2023	Empirical Software Engineering	Journal
S6 [13]	Do Information Retrieval Algorithms for Automated Traceability Perform Effectively on Issue Tracking System Data?	Thorsten Merten Daniel Krämer Bastian Mager Paul Schell Simone Bürsner Barbara Paech	2016	Requirements Engineering: Foundation for Software Quality (REFSQ)	Conference
S7 [14]	Enhancing Model-based Fault Traceability by Using Similarity between Bug and Commit Information	Dongju Jung Kyeongsic Min Jung-Won Lee Byungjeong Lee	2019	JOURNAL OF INTERNET COMPUTING AND SERVICES (JICS)	Journal
S8 [7]	Enhancing Traceability Link Recovery with Unlabeled Data	Jianfei Zhu Guanping Xiao Zheng Zheng Yulei Sui	2022	IEEE International Symposium on Software Reliability Engineering （ISSRE）	Conference
S9 [15]	Eye movements in software traceability link recovery	Bonita Sharif John Meinken Timothy Shaffer Huzefa Kagdi	2017	Empirical Software Engineering (ESE)	Journal
S10 [16]	SpojitR: Intelligently Link Development Artifacts	Michael Rath Mihaela Todorova Tomova Patrick Mäder	2020	IEEE International Conference on Software Analysis, Evolution,	Conference

				and Reengineering （SANER）	
S11 [17]	Identifying Supplementary Bug-fix Commits	Tao Ji Jinkun Pan Liqian Chen Xiaoguang Mao	2018	International Computer Software and Applications Conference （COMPSAC）	Conference
S12 [18]	Influence of Structured Information in Bug Report Descriptions on IR-based Bug Localization	Michael Rath Patrick Mäder	2018	Euromicro Conference on Software Engineering and Advanced Applications （SEAA）	Conference
S13 [19]	Issue Link Label Recovery and Prediction for Open Source Software	Alexander Nicholson Jin L.C. Guo	2021	IEEE International Requirements Engineering Conference Workshops (REW)	Workshops
S14 [20]	Locating Bug IDs and Development Logs in Open Source Software (OSS) projects: An Experience Report	Bilyaminu Auwal Romo Andrea Capiluppi Ajaz Ali	2018	International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)	Conference
S15 [21]	On the effectiveness of automated tracing from model changes to project issues	Wouter van Oosten Randell Rasiman Fabiano Dalpiaz Toine Hurkmans	2023	Information and Software Technology (IST)	Journal
S16 [22]	RCLinker: Automated Linking of Issue Reports and Commits Leveraging Rich Contextual Information	Tien-Duy B. Le Mario Linares Vasquez David Lo Denys Poshyvanyk	2015	IEEE International Conference on Program Comprehension （ICPC）	Conference
S17 [23]	EALink: An Efficient and Accurate Pre-trained Framework for Issue-Commit Link Recovery	Chenyuan Zhang Yanlin Wang Zhao Wei Yong Xu Juhong Wang Hui Li Rongrong Ji	2023	International Conference on Automated Software Engineering （ASE）	Conference
S18 [24]	Traceability in the Wild: Automatically Augmenting Incomplete Trace Links	Michael Rath Jacob Rendall Jin L.C. Guo Jane Cleland-Huang Patrick Mäder	2018	International Conference on Software Engineering （ICSE）	Conference
S19 [25]	Traceability recovery between bug reports and test cases-a Mozilla Firefox case study	Guilherme Gadelha Franklin Ramalho Tiago Massoni	2021	Automated Software Engineering （ASE）	Journal
S20 [26]	AmaLgam+: Composing rich information sources for accurate bug localization	Shaowei Wang David Lo	2016	Journal of Software: Evolution and Process	Journal
S21 [27]	Bug Localization Based on Code Change Histories and Bug Reports	Klaus Changsun Youm June Ahn Jeongho Kim Eunseok Lee	2015	Asia-Pacific Software Engineering Conference （APSEC）	Conference
S22 [28]	Discovering Loners and Phantoms in Commit and Issue Data	Gerald Schermann Martin Brandtner Sebastiano Panichella Philipp Leitner	2015	IEEE International Conference on Program Comprehension （ICPC）	Conference

		Harald Gall			
S23 [29]	Improving Missing Issue-Commit Link Recovery using Positive and Unlabeled Data	Yan Sun Celia Chen Qing Wang Barry Boehm	2017	International Conference on Automated Software Engineering (ASE)	Conference
S24 [30]	RAT: A Refactoring-Aware Traceability Model for Bug Localization	Feifei Niu Wesley K. G.Assunção LiGuo Huang Christoph Mayr-Dorn Jidong Ge Bin Luo Alexander Egyed	2023	International Conference on Software Engineering (ICSE)	Conference

1.2 Extracted data for RQ2: What is the definition of BT?

RQ3: What types of bug trace links are recovered from primary studies?

Index	**Source Artifact**	**Target Artifact**	**Datasets**
S1 [8]	Bug Report	Commit	CLI Collections CSV IO Lang Math
S2 [9]	Bug Report	Problem Report	Firefox Browser VLC Media Player VLC Media Player Nextcloud
S3 [10]	Bug Report	Source Code	Axis2 Derby Drools Hadoop HornetQ Infinispan Izpack Keycloak Log4J2 Pig Railo Seam2 Teiid Weld Wildfly
S4 [11]	Bug Report	Commit	Beam Flink Freemarker Airflow Arrow Netbeans Ignite Isis Groovy Cassandra Ambari

			Calcite
S5 [12]	Bug Report	Commit	Isis Beam Tika Tez Avro Nutch OODT Ivy Giraph Buildr Keras Log4net
S6 [13]	Bug Report	Feature Bug Report	c:geo Lighttpd Radiant Redmine
S7 [14]	Bug Report	Commit	Zxing
S8 [7]	Bug Report	Commit	**For train** Albergate CCHIT CMI eANCI EasyClinic EBT eTOUR GANNT HIPAA Ice Breaker Infused Pump iTrust Kiosk SMOS WARC **For evaluation** Flask Pgcli Keras
S9 [15]	Bug Report	Source Code	JabRef (a graphical application for managing bibliographic databases)
S10 [16]	Bug Report	Commit	CRUNCH FALCON AVRO PIG KAFKA
S11 [17]	Bug Report	Commit	WordPress-Android Atom Moby OpenCV Kubernetes Swift
S12 [18]	Bug Report	Source Code	Derby Drools Groovy

			Infinispan Maven Pig Seam2
S13 [19]	Bug Report	Bug Report	Hive Ambari Flex
S14 [20]	Bug Report	Commit	Brackets Leaflet Reddit CocoaPods Puma AutoMapper MonoDevelop CodeHub Manos puppet
S15 [21]	Commit	Bug Report	Company Control Data Learn Portfolio Service Store
S16 [22]	Bug Report	Commit	CLI Collections CSV IO Lang Math
S17 [23]	Bug Report	Commit	Ambari Calcite Groovy Ignite Isis Netbeans
S18 [24]	Bug Report	Commit	Derby Drools Groovy Infinispan Maven Pig
S19 [25]	Bug Report	Test Case	Mozilla Firefox
S20 [26]	Bug Report	Source Code	AspectJ Eclipse SWT ZXing
S21 [27]	Bug Report	Source Code	AspectJ SWT ZXing
S22 [28]	Bug Report	Commit	ActiveMQ Ambari Camel CXF Felix Hadoop

			HBase Hive Jackrabbit Oak Karaf PDFBox Sling Spark Stanbol Tika
S23 [29]	Bug Report	Commit	Avro Buildr Chukwa Falcon Giraph Ivy Knox Log4net Nutch OODT Tez Tika
S24 [30]	Bug Report	Source Code	Derby Drools Hornetq Izpack Keycloak Log4j2 Railo Seam2 Teiid Weld Wildfly

The datasets used to generate trace links for each type, along with the datasets frequently utilized in the primary studies:

Source Artifact	Target Artifacts	Datasets	Reference	Datasets (Freq. > 2)
Bug Report	Commit	CLI, Collections, CSV, IO, Lang, Math, Beam, Flink, Freemarker, Airflow, Netbeans, Ignite, Isis, Groovy, Cassandra , Ambari , Calcite, Tika, Tez, Avro, Nutch, OODT, Ivy, Giraph, Buildr, Keras, Log4net, Zxing, Flask, Pgcli, Keras, Crunch, Falcon, Pig, Kafka, WordPress-Android, Atom, Moby, OpenCV, Kubernetes, Swift, Brackets, Leaflet, Reddit, CocoaPods, Puma, AutoMapper, MonoDevelop, CodeHub, Manos, puppet, Derby, Drools, Infinispan, Maven, ActiveMQ, Camel, CXF, Felix, Hadoop, HBase, Hive, Jackrabbit Oak, Karaf, PDFBox, Sling, Spark, Stanbol, Chukwa, Knox, Company, Control, Data, Learn, Portfolio, Service, Store	S1, S4, S5, S7, S8, S10, S11, S14, S15, S16, S17, S18, S22, S23	Isis, Groovy, Ambari, Tika, Avro, Zxing, Pig, Derby, Drools, Infinispan, Seam2
	Source Code	Axis2, Derby, Drools, Hadoop, HornetQ, Infinispan, Izpack, Keycloak, Log4J2, Pig, Railo, Seam2, Teiid, Weld, Wildfly, JabRef, Groovy, Maven, Seam2, AspectJ, Eclipse, SWT, ZXing	S3, S9, S12, S20, S21, S24	
	Problem Report	Firefox Browser,VLC Media Player,VLC Media Player,Nextcloud	S2	
	Bug Report	c:geo,Lighttpd,Radiant,Redmine,Hive,Ambari,Flex	S6, S13	
	Feature	c:geo,Lighttpd,Radiant,Redmine	S6	
	Test Case	Mozilla Firefox	S19	

1.3RQ4：Which techniques are used in the process of BTR?

Index	Title	Model	Categories of techniques used
S1 [8]	FRLink: Improving the recovery of missing issue-commit links by revisiting file relevance	FRLink	IR
S2 [9]	Automatically Matching Bug Reports with Related App Reviews	DeepMatcher	DL
S3 [10]	Analyzing Requirements and Traceability Information to Improve Bug Localization	TraceScore	IR+ML+DL+Other
S4 [11]	Automated Recovery of Issue-Commit Links Leveraging Both Textual and Non-textual Data	Hybrid-Linker	ML+IR+Other
S5 [12]	BTLink : automatic link recovery between issues and commitsbased on pre-trained BERT model	BTLink	DL
S6 [13]	Do Information Retrieval Algorithms for Automated Traceability Perform Effectively on Issue Tracking System Data?	OpenTrace	IR
S7 [14]	Enhancing Model-based Fault Traceability by Using Similarity between Bug and Commit Information	Fault Traceability Enhancement Technique	IR +Heuristic+Other
S8 [7]	Enhancing Traceability Link Recovery with Unlabeled Data	TRACEFUN	IR+DL
S9 [15]	Eye movements in software traceability link recovery	iTrace	eye-tracking based
S10 [16]	SpojitR: Intelligently Link Development Artifacts	SpojitR	ML+IR
S11 [17]	Identifying Supplementary Bug-fix Commits	SupBCFinder	ML + Heuristic
S12 [18]	Influence of Structured Information in Bug Report Descriptions on IR-based Bug Localization	-	IR + Heuristic
S13 [19]	Issue Link Label Recovery and Prediction for Open Source Software	-	IR+ ML+DL+Other
S14 [20]	Locating Bug IDs and Development Logs in Open Source Software (OSS) projects:An Experience Report	-	SZZ
S15 [21]	On the effectiveness of automated tracing from model changes to project issues	LCDTrace	ML+IR

S16 [22]	RCLinker: Automated Linking of Issue Reports and Commits Leveraging Rich Contextual Information	RCLinker	ML+Other
S17 [23]	EALink: An Efficient and Accurate Pre-trained Framework for Issue-Commit Link Recovery	EALink	DL
S18 [24]	Traceability in the Wild: Automatically Augmenting Incomplete Trace Links	-	IR+ML+Other
S19 [25]	Traceability recovery between bug reports and test cases-a Mozilla Firefox case study	-	IR+DL
S20 [26]	AmaLgam+:Composing rich information sources for accurate bug localization	AmaLgam+	IR+Other
S21 [27]	Improved bug localization based on code change histories and bug reports	BLIA	IR
S22 [28]	Discovering Loners and Phantoms in Commit and Issue Data	PaLiMod	heuristics
S23 [29]	Improving Missing Issue-Commit Link Recovery using Positive and Unlabeled Data	PULink	ML
S24 [30]	RAT: A Refactoring-Aware Traceability Model for Bug Localization	RAT	IR+ML+Other

The distribution of techniques (ML, DL and IR) and the corresponding states:

Index	ML	Stage	DL	Stage	IR	Stage
S1 [8]					1) VSM 2) TF-IDF	1) Link generation 2) Link preparation
S2 [9]			DistilBERT	Link preparation		
S3 [10]	Decision Tree	Link generation			1) VSM 2) TF-IDF	Link generation
S4 [11]	1) Decision Tree 2) Gradient Boosting 3) Logistic Regression 4) Stochastic Gradient Descent 5) Naïve Bayes 6) Generalized Linear 7) Random Forest	Link generation	1) Word2Vec 2) Doc2Vec	Link preparation	TF-IDF	Link preparation

	8) XGBoost Model					
S5 [12]			1) RoBERTa 2) CodeBERT	Link preparation		
S6 [13]					1) VSM 2) LSI 3) BM25 4) BM25+ 5) BM25L	Link generation
S7 [14]					1) VSM 2) TF-IDF	1) Link generation 2) Link preparation
S8 [7]			1) Glove 2) LSTM	1) Link preparation 2) Link generation	1) VSM 2) TF-IDF	1) Link generation 2) Link preparation
S9 [15]						
S10 [16]	Random Rorest	Link generation			1)VSM-nGram 2) TF-IDF	Link generation Link preparation
S11 [17]	SVM	Link preparation				
S12 [18]					1) BLUiR 2) AmaLgam	Link generation
S13 [19]	1) Logistic Regression 2) Random Forest 3) fastText	1) 2) Link generation 3) Link preparation	Neural Network	Link generation	TF-IDF	Link preparation
S14 [20]						
S15 [21]	1) XGBoost 2) LightGBM 3) Random Forests	Link generation			1) VSM 2) TF-IDF	1) Link generation 2) Link preparation
S16 [22]	Random forest	Link generation				
S17 [23]			1) RoBERTa、 2) CodeBERT 3) Contrastive Learning	Link preparation		
S18 [24]	1) Naive Bayes 2) Decision Tree 3) Random Forest	Link generation			VSM-nGram	Link generation
S19 [25]			Glove	preprocessing stage	1) LSI 2) LDA 3) BM25	Link generation
S20 [26]					1) BugLocator 2) BLUiR 3) TF-IDF	1) 2) Link generation 3) preparation
S21 [27]				tr	rVSM	Link generation
S22 [28]						
S23 [29]	Random Forest	Link generation				

S24 [30]	SVM	Link generation			1) VSM 2) TF-IDF	1) Link generation 2) Link preparation

The distribution of techniques (Heuristic, Eye-tracking and Other) and the corresponding states:

Index	Heuristic	Stage	Eye-tracking	Stage	Other	Stage
S1 [8]						
S2 [9]						
S3 [10]					Random Undersampling	Link preparation
S4 [11]					1) one-hot transformation 2) Random Undersampling	Link preparation
S5 [12]						
S6 [13]						
S7 [14]	keyword extraction heuristic	Link preparation			Behavior Model	Link refinement
S8 [7]						
S9 [15]			Itrace	Link generation		Link generation
S10 [16]						
S11 [17]	identify commit Heuristics	Link preparation				
S12 [18]	text tagging heuristic	Link preparation				
S13 [19]					SMOTE	
S14 [20]					SZZ	Link preparation
S15 [21]					1) Random Undersampling 2) SMOTE	Link preparation
S16 [22]					1) Near-Miss 2)ChangeScribe	Link preparation
S17 [23]						
S18 [24]					Random Undersampling	Link preparatio
S19 [25]						
S20 [26]					Genetic Algorithm	Link generation
S21 [27]						
S22 [28]	(1) Loner heuristic (2) Phantom heuristic.	Link generation				

S23 [29]						
S24 [30]					1) SimiScore 2) TraceScore	Link generation

The distribution of techniques across different strategies:

Strategies	Stage	Techniques						Strategy Characteristics
		IR-base	ML-based	DL-based	Heuristic-based	Eye-trace	Other	
Representation Learning	P	TF-IDF [7][8][10][11] [14][16] [19] [21][26]	SVM, fastText [19][17]	DistilBERT, Word2Vec, Doc2Vec, RoBERTa, CodeBERT, Contrastive learning, Glove [9][11][7][25] [12][23]			One-Hot Transformation [11]	TF-IDF, Word Embeddings, BERT,etc., are used to generate vector representations of artifacts
Data Balance	P						Random-UnderSampling, SMOTE, Near-Miss [10][11][19] [21][22][24]	Utilizing techniques to address the imbalance in the dataset, aiming to achieve a balanced distribution of samples
Link Classification	G		Decision Tree, Gradient Boosting, Logistic Regression , Stochastic Gradient Descen, Random Forest, Naïve Bayes, Generalized Linear, XGBoost, LightGBM [10][11][16][19] [21][22][24][29]	LSTM, Neural Network [7][19]				Utilizing the classifier to distinguish whether artifacts have trace link
Similarity Calculation	G	VSM, rVSM, VSM-nGram, LSI, LDA, BM25, BM25+, BM25L [8][10][13] [14][7][16] [21][24][25] [8]					SimiScore, TraceScore [30]	Calculate the similarity between source artifacts and target artifacts
Traceability Enhancement	P,R				Keyword Extraction, commit Identification, Text Tagging, Loner, Phantom [14][17] [18][28]		SZZ, ChangeScribe, Behavior Model [20][22][14]	Utilizing techniques during the creation of trace link to achieve better traceability results
Bug Location	G	BLUiR, AmaLgam, BugLocator [18][26]				Itrace [15]		Using specialized bug localization techniques to generate trace link between bug report and source code
Optimization Algorithm	G						Genetic Algorithm [26]	Using optimization algorithms such as genetic algorithms to assign optimal weights to text similarity suspicious scores

1.4 RQ5：Which metrics are used to evaluate the performance of BTR techniques?

Index	Title	Metrics
S1 [8]	FRLink: Improving the recovery of missing issue-commit links by revisiting file relevance	Precision Recall F-measure MCC
S2 [9]	Automatically Matching Bug Reports With Related App Reviews	MAP Hit Ratio
S3 [10]	Analyzing Requirements and Traceability Information to Improve Bug Localization	MAP MRR Top@n
S4 [11]	Automated Recovery of Issue-Commit Links Leveraging Both Textual and Non-textual Data	Precision Recall F1
S5 [12]	BTLink : automatic link recovery between issues and commits based on pre-trained BERT model	F1 MCC ACC PF AUC Precision Recall
S6 [13]	Do Information Retrieval Algorithms for Automated Traceability Perform Effectively on Issue Tracking System Data?	Precision Recall F1 F2
S7 [14]	Enhancing Model-based Fault Traceability by Using Similarity between Bug and Commit Information	Accuracy
S8 [7]	Enhancing Traceability Link Recovery with Unlabeled Data	MAP F1 F2
S9 [15]	Eye movements in software traceability link recovery	Precision Recall
S10 [16]	SpojitR: Intelligently Link Development Artifacts	Precision Recall Accuracy
S11 [17]	Identifying Supplementary Bug-fix Commits	Precision Recall F-measure
S12 [18]	Influence of Structured Information in Bug Report Descriptions on IR-based Bug Localization	Top@n MAP MRR
S13 [19]	Issue Link Label Recovery and Prediction for Open Source Software	F1
S14 [20]	Locating Bug IDs and Development Logs in Open Source Software (OSS) projects: An Experience Report	Precision Recall F-measure
S15 [21]	On the effectiveness of automated tracing from model changes to project issues	F2 F0.5 Precision Recall
S16 [22]	RCLinker: Automated Linking of Issue Reports and Commits Leveraging Rich Contextual	Precision Recall F-measure

	Information	
S17 [23]	EALink: An Efficient and Accurate Pre-trained Framework for Issue-Commit Link Recovery	Precision NDGG@k MRR Hit@n
S18 [24]	Traceability in the Wild: Automatically Augmenting Incomplete Trace Links	Precision Recall F2 F0.5
S19 [25]	Traceability recovery between bug reports and test cases-a Mozilla Firefox case study	Precision Recall F2 REI
S20 [26]	AmaLgam+: Composing rich information sources for accurate bug localization	MAP MRR Hit@n
S21 [27]	Improved bug localization based on code change histories and bug reports	Top@n MAP MRR
S22 [28]	Discovering Loners and Phantoms in Commit and Issue Data	Precision Recall F-measure
S23 [29]	Improving Missing Issue-Commit Link Recovery using Positive and Unlabeled Data	Precision Recall F-measure
S24 [30]	RAT: A Refactoring-Aware Traceability Model for Bug Localization	Top@n MAP MRR

The metrics used in each primary study along with their usage frequency:

Reference	Primary Metrics								Secondary Metrics								Total
	F-measures			Recall	Precision	Accuracy	Top @n	Hit @n	MCC	MAP	MRR	ACC	PF	AUC	NDGG @K	REI	
	F1	F2	F 0.5														
S1	✓			✓	✓				✓								4
S2								✓		✓							2
S3							✓			✓	✓						3
S4	✓			✓	✓												3
S5	✓			✓	✓				✓			✓	✓	✓			7
S6	✓	✓		✓	✓												4
S7						✓											1
S8	✓	✓								✓							3
S9				✓	✓												2
S10				✓	✓	✓											3
S11	✓			✓	✓												3
S12							✓			✓	✓						3
S13	✓																1
S14	✓			✓	✓												3
S15		✓	✓	✓	✓												4
S16	✓			✓	✓												3
S17					✓			✓			✓				✓		4
S18		✓	✓	✓	✓												4
S19		✓		✓	✓											✓	4
S20								✓		✓	✓						3
S21							✓			✓	✓						3
S22	✓			✓	✓												3
S23	✓			✓	✓												3
S24							✓			✓	✓						3

1.5RQ5：What is the overall quality of primary studies?

1.5.1 RQ5.1: What is the degree of decision support for technology transfer?

Index	Q1-Q4				Q5-Q8			
	Research method	Context	Subject	Degree of automation	Context described	Study Design	Validity discussed	Measures Used
S1	Lab Experience	Academic	Researcher	strong	Strong	Strong	Strong	0.8
S2	Case study	Academic	practitioner	strong	Strong	Strong	Medium	0.6
S3	Lab Experience	Academic	Researcher	Medium	Medium	Strong	Strong	0.6
S4	Lab Experience	Academic	Researcher	strong	Strong	Strong	Strong	0.6
S5	Lab Experience	Academic	Researcher	strong	Strong	Strong	Medium	1.0
S6	Lab Experience	Academic	Student	strong	Strong	Strong	Medium	0.8
S7	Lab Experience	Academic	Student	Medium	Medium	Medium	Weak	0.2
S8	Lab Experience	Academic	Researcher	Medium	Strong	Strong	Strong	0.6
S9	Case study	Academic	practitioner	strong	Strong	Strong	Strong	0.6
S10	Lab Experience	Academic	Researcher	Medium	Medium	Strong	Weak	0.6
S11	Lab Experience	Academic	Researcher	Medium	Strong	Strong	Strong	0.6
S12	Lab Experience	Academic	Researcher	strong	Medium	Strong	Medium	0.6
S13	Case study	Academic	Student	strong	Strong	Strong	Medium	0.2
S14	Lab Experience	Academic	Student	Medium	Medium	Medium	Strong	0.6
S15	Case study	Industry	Student	strong	Strong	Strong	Strong	0.8
S16	Lab Experience	Academic	Student	strong	Strong	Strong	Strong	0.6
S17	Case study	Academic	practitioner	strong	Strong	Strong	Medium	0.8
S18	Lab Experience	Academic	Researcher	strong	Strong	Strong	Medium	0.8
S19	Case study	Academic	Researcher	strong	Strong	Strong	Strong	0.8
S20	Lab Experience	Academic	Researcher	strong	Strong	Strong	Strong	0.6
S21	Lab Experience	Academic	Student	strong	Strong	Strong	Strong	0.6
S22	Lab Experience	Academic	Researcher	Strong	Strong	Strong	Medium	0.6
S23	Lab Experience	Academic	Researcher	Strong	Medium	Strong	Weak	0.6
S24	Lab Experience	Academic	Student	Strong	Strong	Strong	Strong	0.6

1.5.2 RQ5.2: What is the degree of reproducibility for techniques investigated in primary studies?

Index	Method					Data				Experiment				
	Proble m	Objective /goal	Research method	Research questions	Pseudo code	Training data	Validation data	Test data	Results	Hypothesis and Prediction	Source code	Hardware Specificat -ions	Software dependen-cies	Experi-ment setup
S1	1	1	1	1	1	1	0	1	1	0	1	1	0	1
S2	1	1	1	1	0	1	0	0	1	0	1	0	0	1
S3	1	1	1	1	0	1	0	1	1	1	0	0	1	1
S4	1	1	1	1	0	1	0	1	1	0	1	1	1	1
S5	1	1	1	1	0	1	1	1	1	0	1	1	0	1
S6	1	0	0	1	0	**0**	**0**	**0**	1	0	1	0	1	1
S7	0	0	1	0	1	0	0	0	1	0	0	0	0	1
S8	0	1	1	1	0	1	1	1	1	1	1	1	0	1
S9	1	1	1	1	0	**0**	**0**	**0**	1	1	1	1	1	1
S10	1	1	1	0	0	1	0	1	1	0	1	0	1	1
S11	1	1	1	1	1	1	0	1	1	1	0	0	1	1
S12	0	0	1	0	0	**0**	**0**	**0**	1	0	0	0	0	1
S13	1	1	1	1	0	1	1	1	1	1	0	0	1	1
S14	1	1	1	0	0	**0**	**0**	**0**	1	1	0	0	1	1
S15	0	1	1	1	0	1	0	1	1	0	1	0	1	1
S16	1	1	1	1	1	1	0	1	1	0	0	1	1	1
S17	1	1	1	1	0	1	1	1	1	0	1	1	1	1
S18	1	1	1	1	0	1	0	1	1	0	0	0	0	1
S19	1	1	1	1	0	1	0	0	1	0	1	1	1	1
S20	1	1	1	1	0	1	0	1	1	1	0	1	1	1
S21	1	1	1	1	0	**0**	**0**	**0**	1	1	1	0	1	1
S22	1	1	1	1	1	**0**	**0**	**0**	1	0	0	0	0	1
S23	1	1	1	0	0	1	1	1	1	0	0	0	0	1
S24	1	1	1	1	0	**0**	**0**	**0**	1	0	1	0	1	1

Note： The values in the data factor are bolded to indicate a weight of 0.

2. Search process record

2.1Digital libraries

Database	Website
Google Scholar	https://scholar.google.com/
ScienceDirect	https://www.sciencedirect.com/
EI	https://www.engineeringvillage.com/
IEEE	https://ieeexplore.ieee.org/
Wiley	https://onlinelibrary.wiley.com
Springer	https://www.springer.com/
ACM	https://dl.acm.org/

2.2Inclusion/Exclusion criteria

Inclusion selection criteria	
I1	The time span of the study is from January 2014 to December 2023, and the study must be published as a journal paper, conference paper or workshop.
I2	The dataset primarily consists of bug report artifacts.
I3	The research topic must be techniques used in the BTR process.
I4	When presented with two papers by the same authors with the same technology and topic, we select the more complete one.
Exclusion selection criteria	
E1	The study is a review paper or grey literature.
E2	This study is not written in English.
E3	This study is not a complete full-text or is less than 4 pages.

2.3Search terms

PICO	Search terms
Population(P)	traceability recovery, traceability maintenance, traceability assessment, trace links, traceability link, bug trace, bug tracing, bug traceability, bug links
Intervention(I)	bug, issue, defect

Search statements:

("bug" OR "issue" OR "defect") AND ("traceability recovery" OR "traceability maintenance" OR "traceability assessment" OR "trace links" OR "traceability link" OR "bug trace" OR "bug tracing" OR "bug traceability" OR "bug links")

2.4Search results

Database	Number of search	After filtering duplicate studies and preliminary filtering (I1, E1-E3)	Filtering by title, abstract, and keyword (I2-I3)	Filtering by full-text (I2-I4)	Snowballing	Total
Google scholar	3370	121	59	19	5	24
ScienceDirect	539					
EI	90					
IEEE	35					
Wiley	74					
Springer	310					
ACM	256					

(1) Google scholar

Advanced search:

("bug" OR "issue" OR "defect") AND ("traceability recovery" OR "traceability maintenance" OR "traceability assessment" OR "trace links" OR "traceability link" OR "bug trace" OR "bug tracing" OR "bug traceability" OR "bug links")

Screenshot of search process in Google scholar:

(2) ScienceDirect

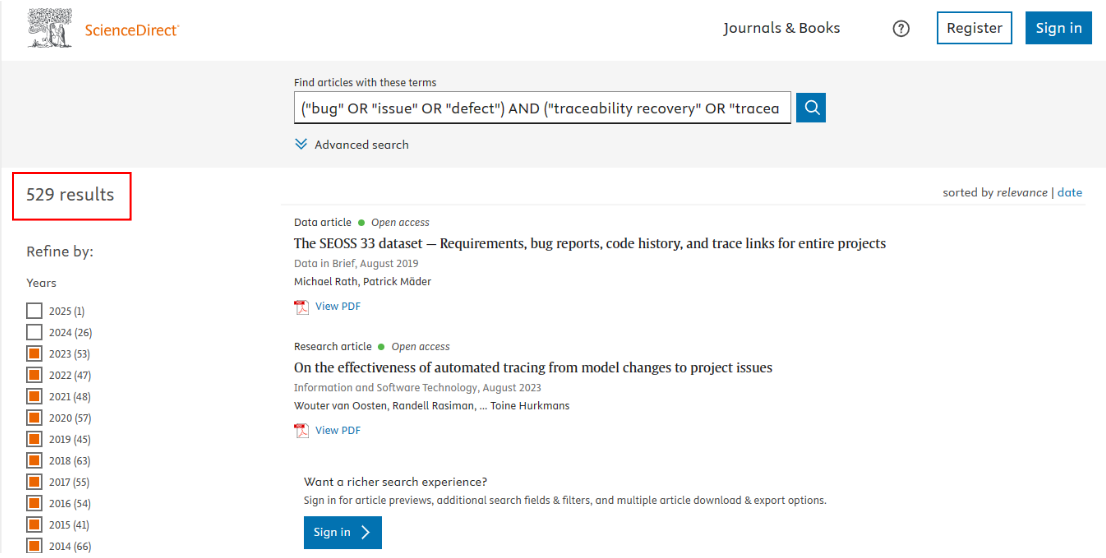
Advanced search:

("bug" OR "issue" OR "defect") AND ("traceability recovery" OR "traceability maintenance" OR "traceability assessment" OR "trace links" OR "traceability link" OR "bug trace" OR "bug tracing" OR "bug traceability" OR "bug links")

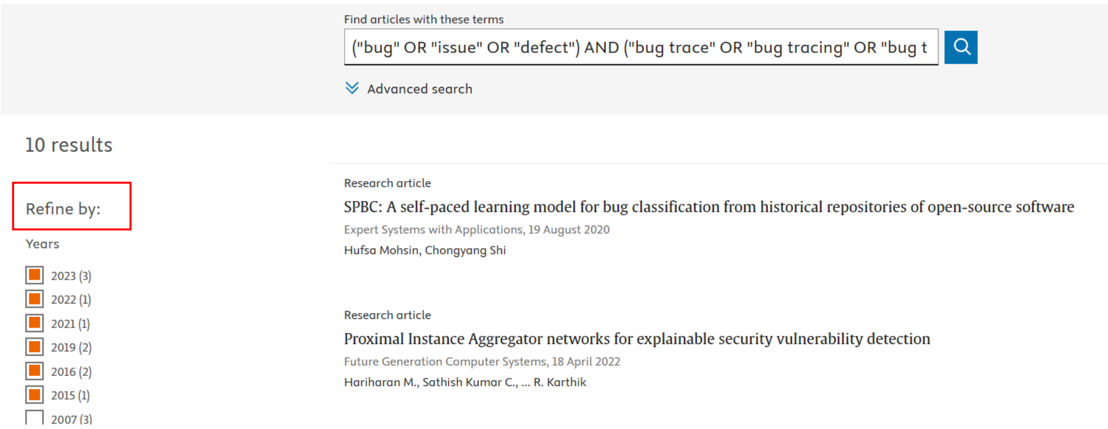
Due to the limitation of Boolean connectors (up to 8 per field), we split the search term into two parts: a) and b).

Screenshot of search process in ScienceDirect:

a) ("bug" OR "issue" OR "defect") AND ("traceability recovery" OR "traceability maintenance" OR "traceability assessment" OR "trace links" OR "traceability link")



b) ("bug" OR "issue" OR "defect") AND ("bug trace" OR "bug tracing" OR "bug traceability" OR "bug links")

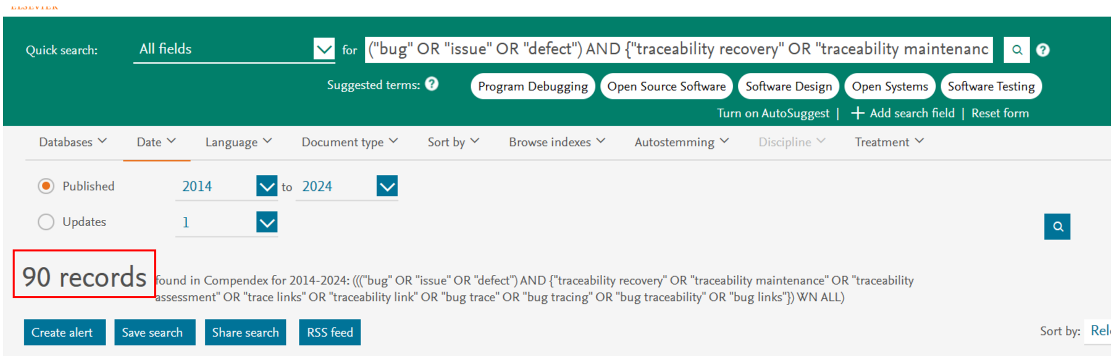


(3) EI

Advanced search:

("bug" OR "issue" OR "defect") AND {"traceability recovery" OR "traceability maintenance" OR "traceability assessment" OR "trace links" OR "traceability link" OR "bug trace" OR "bug tracing" OR "bug traceability" OR "bug links"}

Screenshot of search process in EI:

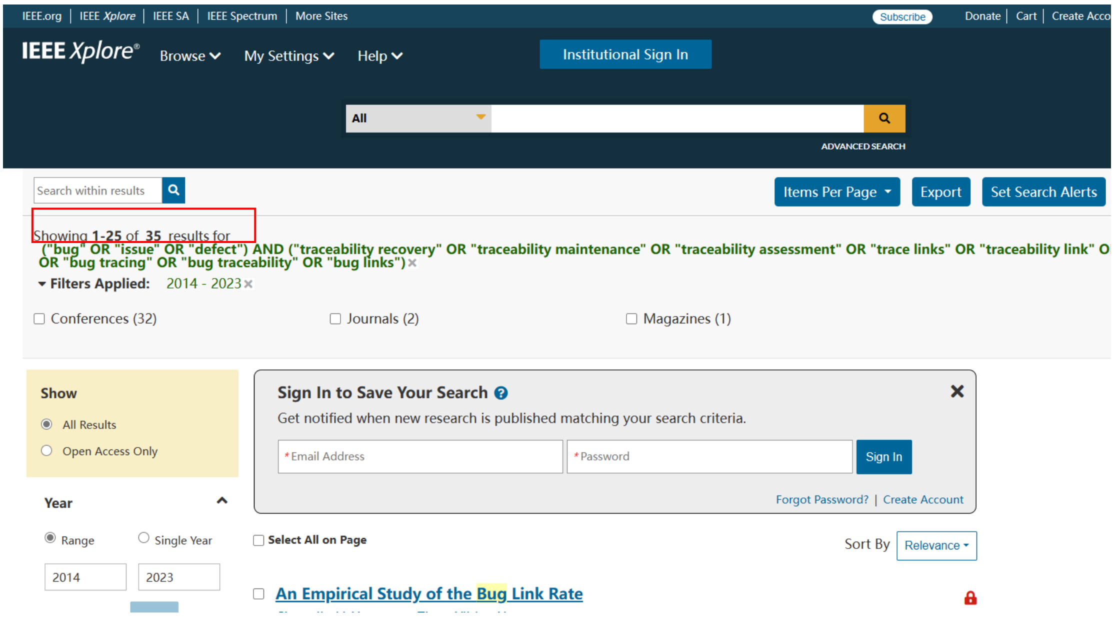


(4) IEEE

Advanced search:

("bug" OR "issue" OR "defect") AND ("traceability recovery" OR "traceability maintenance" OR "traceability assessment" OR "trace links" OR "traceability link" OR "bug trace" OR "bug tracing" OR "bug traceability" OR "bug links")

Screenshot of search process in IEEE:

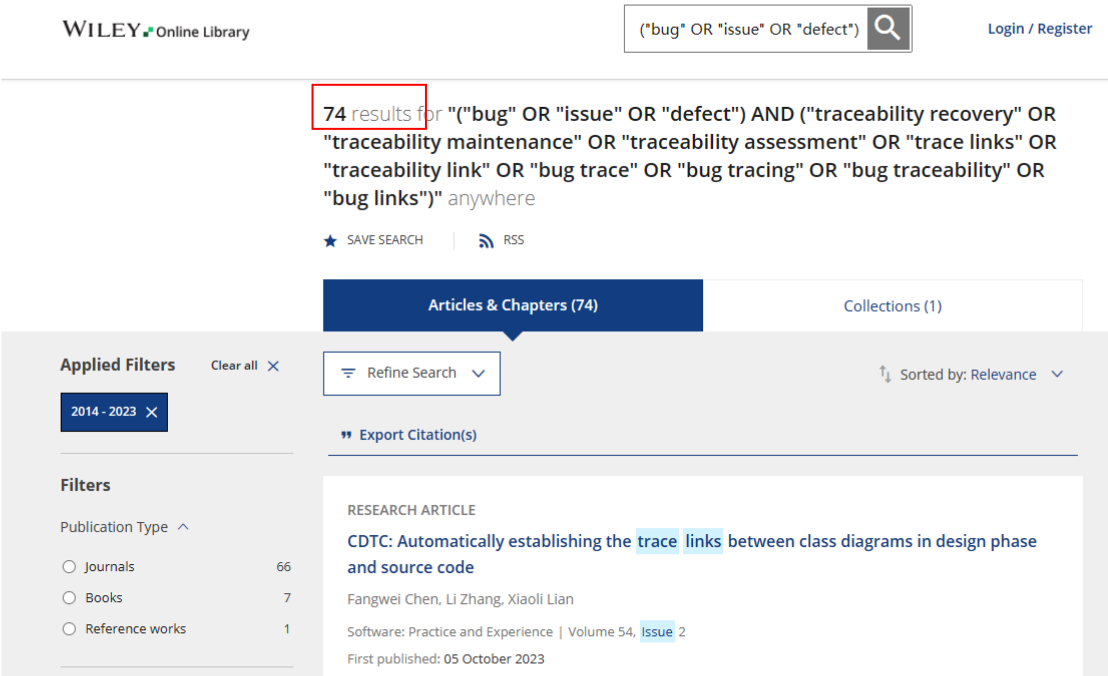


(5) Wiley

Advanced search:

("bug" OR "issue" OR "defect") AND ("traceability recovery" OR "traceability maintenance" OR "traceability assessment" OR "trace links" OR "traceability link" OR "bug trace" OR "bug tracing" OR "bug traceability" OR "bug links")

Screenshot of search process in Wiley:

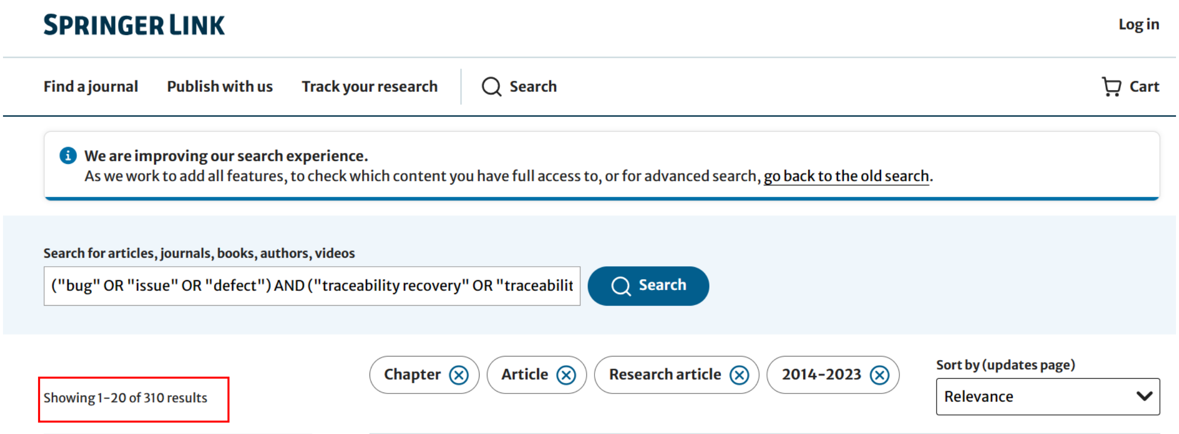


(6) Springer

Advanced search:

("bug" OR "issue" OR "defect") AND ("traceability recovery" OR "traceability maintenance" OR "traceability assessment" OR "trace links" OR "traceability link" OR "bug trace" OR "bug tracing" OR "bug traceability" OR "bug links")

Screenshot of search process in Springer:



(7) ACM

Advanced search:

("bug" OR "issue" OR "defect") AND ("traceability recovery" OR "traceability maintenance" OR "traceability assessment" OR "trace links" OR "traceability link" OR "bug trace" OR "bug tracing" OR "bug traceability" OR "bug links")

Screenshot of search process in ACM:

ACM DIGITAL LIBRARY | Association for Computing Machinery

Browse | About | Sign in | Register

Journals | Magazines | Proceedings | Books | SIGs | Conferences | People

("bug" OR "issue" OR "defec...

Search Results

("bug" OR "issue" OR "defect") AND ("traceability recovery" OR ...

Advanced Search

Applied Filters

2014 - 2023

Clear All

People

256 Results for: [[All: "bug"] OR [All: "issue"] OR [All: "defect"]] AND [[All: "traceability recovery"] OR [All: "traceability maintenance"] OR [All: "traceability assessment"] OR [All: "trace links"] OR [All: "traceability link"] OR [All: "bug trace"] OR [All: "bug tracing"] OR [All: "bug traceability"] OR [All: "bug links"]] AND [E-Publication Date: (01/01/2014 TO 12/31/2023)]

Edit Search | Save Search | RSS

Feedback

Searched The ACM Full-Text Collection (734,550 records) | Expand your search to The ACM Guide to Computing Literature (3,679,918 records)