David Lo

List of Publications by Citations

405 papers	8,805 citations	50 h-index	69 g-index
474 ext. papers	12,420 ext. citations	2.6 avg, IF	6.84 L-index

#	Paper	IF	Citations
405	Where should the bugs be fixed? More accurate information retrieval-based bug localization based on bug reports **2012**,		226
404	A discriminative model approach for accurate duplicate bug report retrieval **2010**,		190
403	Towards more accurate retrieval of duplicate bug reports **2011**,		163
402	HYDRA: Massively Compositional Model for Cross-Project Defect Prediction. *IEEE Transactions on Software Engineering*, **2016**, 42, 977-998	3.5	148
401	Deep Learning for Just-in-Time Defect Prediction **2015**,		133
400	Deep code comment generation **2018**,		115
399	Duplicate bug report detection with a combination of information retrieval and topic modeling **2012**,		113
398	Classification of software behaviors for failure detection **2009**,		101
397	TLEL: A two-layer ensemble learning approach for just-in-time defect prediction. *Information and Software Technology*, **2017**, 87, 206-220	3.4	94
396	SMArTIC **2006**,		93
395	Practitioners' expectations on automated fault localization **2016**,		92
394	History Driven Program Repair **2016**,		91
393	Version history, similar report, and structure: putting them together for improved bug localization **2014**,		90
392	Information Retrieval Based Nearest Neighbor Classification for Fine-Grained Bug Severity Prediction **2012**,		86
391	What are the characteristics of high-rated apps? A case study on free Android Applications **2015**,		83
390	**2013**,		77
389	S3: syntax- and semantic-guided repair synthesis via programming by examples **2017**,		76

388	. *IEEE Transactions on Software Engineering*, **2019**, 1-1	3.5	75
387	Improving Automated Bug Triaging with Specialized Topic Model. *IEEE Transactions on Software Engineering*, **2017**, 43, 272-297	3.5	72
386	What Security Questions Do Developers Ask? A Large-Scale Study of Stack Overflow Posts. *Journal of Computer Science and Technology*, **2016**, 31, 910-924	1.7	72
385	Identifying bug signatures using discriminative graph mining **2009**,		72
384	. *IEEE Transactions on Information Forensics and Security*, **2017**, 12, 1269-1284	8	71
383	Network Structure of Social Coding in GitHub **2013**,		71
382	Searching connected API subgraph via text phrases **2012**,		70
381	Information retrieval and spectrum based bug localization: better together **2015**,		67
380	Identifying self-admitted technical debt in open source projects using text mining. *Empirical Software Engineering*, **2018**, 23, 418-451	3.3	67
379	Data Mining for Software Engineering. *Computer*, **2009**, 42, 55-62	1.6	67
378	ELBlocker: Predicting blocking bugs with ensemble imbalance learning. *Information and Software Technology*, **2015**, 61, 93-106	3.4	65
377	Improved Duplicate Bug Report Identification **2012**,		64
376	Efficient mining of iterative patterns for software specification discovery **2007**,		63
375	Automatic Defect Categorization **2012**,		61
374	An empirical study on developer interactions in StackOverflow **2013**,		61
373	Automatic steering of behavioral model inference **2009**,		61
372	EnTagRec: An Enhanced Tag Recommendation System for Software Information Sites **2014**,		59
371	DRONE: Predicting Priority of Reported Bugs by Multi-factor Analysis **2013**,		59

370	A learning-to-rank based fault localization approach using likely invariants **2016**,		58
369	How practitioners perceive the relevance of software engineering research **2015**,		57
368	Predicting response in mobile advertising with hierarchical importance-aware factorization machine **2014**,		57
367	Automated prediction of bug report priority using multi-factor analysis. *Empirical Software Engineering*, **2015**, 20, 1354-1383	3.3	56
366	Identifying Linux bug fixing patches **2012**,		56
365	Got issues? Who cares about it? A large scale investigation of issue trackers from GitHub **2013**,		55
364	Efficient Mining of Closed Repetitive Gapped Subsequences from a Sequence Database. *Proceedings - International Conference on Data Engineering*, **2009**,	2	55
363	What do developers search for on the web?. *Empirical Software Engineering*, **2017**, 22, 3149-3185	3.3	54
362	Automated library recommendation **2013**,		54
361	**2017**,		54
360	Automatic recommendation of API methods from feature requests **2013**,		54
359	Multi-Factor Duplicate Question Detection in Stack Overflow. *Journal of Computer Science and Technology*, **2015**, 30, 981-997	1.7	53
358	**2006**,		53
357	Understanding the Test Automation Culture of App Developers **2015**,		52
356	Accurate developer recommendation for bug resolution **2013**,		52
355	API method recommendation without worrying about the task-API knowledge gap **2018**,		50
354	An Empirical Study of Classifier Combination for Cross-Project Defect Prediction **2015**,		49
353	Finding relevant answers in software forums **2011**,		49

352	Neural-machine-translation-based commit message generation: how far are we? **2018**,		49
351	Why and how developers fork what from whom in GitHub. *Empirical Software Engineering*, **2017**, 22, 547-578	3.3	48
350	Non-redundant sequential rules—Theory and algorithm. *Information Systems*, **2009**, 34, 438-453	2.7	47
349	Learning extended FSA from software: An empirical assessment. *Journal of Systems and Software*, **2012**, 85, 2063-2076	3.3	45
348	Extended comprehensive study of association measures for fault localization. *Journal of Software: Evolution and Process*, **2014**, 26, 172-219	1	45
347	Inferring semantically related software terms and their taxonomy by leveraging collaborative tagging **2012**,		44
346	RACK: Automatic API Recommendation Using Crowdsourced Knowledge **2016**,		43
345	Compositional Vector Space Models for Improved Bug Localization **2014**,		43
344	Measuring Program Comprehension: A Large-Scale Field Study with Professionals. *IEEE Transactions on Software Engineering*, **2018**, 44, 951-976	3.5	43
343	Popularity, Interoperability, and Impact of Programming Languages in 100,000 Open Source Projects **2013**,		42
342	Automatic, high accuracy prediction of reopened bugs. *Automated Software Engineering*, **2015**, 22, 75-109	2.5	41
341	An Empirical Study of Bugs in Machine Learning Systems **2012**,		41
340	Summarizing Source Code with Transferred API Knowledge **2018**,		40
339	Who should review this change?: Putting text and file location analyses together for more accurate recommendations **2015**,		38
338	Combining Word Embedding with Information Retrieval to Recommend Similar Bug Reports **2016**,		38
337	Perceptions, Expectations, and Challenges in Defect Prediction. *IEEE Transactions on Software Engineering*, **2020**, 46, 1241-1266	3.5	37
336	Neural Network-based Detection of Self-Admitted Technical Debt. *ACM Transactions on Software Engineering and Methodology*, **2019**, 28, 1-45	3.3	36
335	Revisiting supervised and unsupervised models for effort-aware just-in-time defect prediction. *Empirical Software Engineering*, **2019**, 24, 2823-2862	3.3	36

334	EnTagRec ++: An enhanced tag recommendation system for software information sites. *Empirical Software Engineering*, **2018**, 23, 800-832	3.3	35
333	Bug Characteristics in Blockchain Systems: A Large-Scale Empirical Study **2017**,		35
332	Automatic classification of software related microblogs **2012**,		35
331	Recommending People in Developers' Collaboration Network **2011**,		35
330	Dual analysis for recommending developers to resolve bugs. *Journal of Software: Evolution and Process*, **2015**, 27, 195-220	1	34
329	Concern Localization using Information Retrieval: An Empirical Study on Linux Kernel **2011**,		34
328	Mining modal scenario-based specifications from execution traces of reactive systems **2007**,		34
327	Automated construction of a software-specific word similarity database **2014**,		33
326	Empirical Evaluation of Bug Linking **2013**,		33
325	DeepJIT: An End-to-End Deep Learning Framework for Just-in-Time Defect Prediction **2019**,		32
324	**2017**,		32
323	Matching dependence-related queries in the system dependence graph **2010**,		32
322	Automated Configuration Bug Report Prediction Using Text Mining **2014**,		31
321	Mining and Ranking Generators of Sequential Patterns **2008**,		31
320	Automated Debugging Considered Harmful Considered Harmful: A User Study Revisiting the Usefulness of Spectra-Based Fault Localization Techniques with Professionals Using Real Bugs from Large Systems **2016**,		31
319	JFIX: semantics-based repair of Java programs via symbolic PathFinder **2017**,		30
318	Scenario-based and value-based specification mining **2010**,		30
317	Deep code comment generation with hybrid lexical and syntactical information. *Empirical Software Engineering*, **2020**, 25, 2179-2217	3.3	30

316	Active Semi-supervised Defect Categorization **2015**,		29
315	Combining Software Metrics and Text Features for Vulnerable File Prediction **2015**,		29
314	High-Impact Bug Report Identification with Imbalanced Learning Strategies. *Journal of Computer Science and Technology*, **2017**, 32, 181-198	1.7	28
313	Active code search **2014**,		28
312	**2008**,		28
311	How does Machine Learning Change Software Development Practices?. *IEEE Transactions on Software Engineering*, **2020**, 1-1	3.5	28
310	Should I follow this fault localization tool's output?. *Empirical Software Engineering*, **2015**, 20, 1237-1274	3.3	27
309	Code coverage and test suite effectiveness: Empirical study with real bugs in large systems **2015**,		27
308	Overfitting in semantics-based automated program repair. *Empirical Software Engineering*, **2018**, 23, 3007-3033	3.3	27
307	An Empirical Study of Adoption of Software Testing in Open Source Projects **2013**,		27
306	Potential biases in bug localization **2014**,		27
305	Comprehensive evaluation of association measures for fault localization **2010**,		27
304	Mining top-K large structural patterns in a massive network. *Proceedings of the VLDB Endowment*, **2011**, 4, 807-818	3.1	27
303	. *IEEE Transactions on Reliability*, **2016**, 65, 1810-1829	4.6	27
302	Defining Smart Contract Defects on Ethereum. *IEEE Transactions on Software Engineering*, **2020**, 1-1	3.5	26
301	Towards more accurate multi-label software behavior learning **2014**,		26
300	Mining Collaboration Patterns from a Large Developer Network **2010**,		26
299	Mining interesting link formation rules in social networks **2010**,		26

298	Mining temporal rules for software maintenance. *Journal of Software: Evolution and Process*, **2008**, 20, 227-247		26
297	Detecting similar repositories on GitHub **2017**,		25
296	On Reliability of Patch Correctness Assessment **2019**,		25
295	Modeling the evolution of development topics using Dynamic Topic Models **2015**,		25
294	Scenario-based and value-based specification mining: better together. *Automated Software Engineering*, **2012**, 19, 423-458	1.5	25
293	Detecting similar applications with collaborative tagging **2012**,		25
292	Automating Change-Level Self-Admitted Technical Debt Determination. *IEEE Transactions on Software Engineering*, **2019**, 45, 1211-1229	3.5	25
291	Mining indirect antagonistic communities from social interactions. *Knowledge and Information Systems*, **2013**, 35, 553-583	2.4	24
290	A Critical Evaluation of Spectrum-Based Fault Localization Techniques on a Large-Scale Software System **2017**,		24
289	Synergizing Specification Miners through Model Fissions and Fusions (T) **2015**,		24
288	Fusion fault localizers **2014**,		24
287	Theory and Practice, Do They Match? A Case with Spectrum-Based Fault Localization **2013**,		24
286	Chaff from the Wheat: Characterizing and Determining Valid Bug Reports. *IEEE Transactions on Software Engineering*, **2020**, 46, 495-525	3.5	24
285	Cross-project build co-change prediction **2015**,		23
284	Active Semi-supervised Approach for Checking App Behavior against Its Description **2015**,		23
283	A More Accurate Model for Finding Tutorial Segments Explaining APIs **2016**,		23
282	A comparative study on the effectiveness of part-of-speech tagging techniques on bug reports **2015**,		23
281	Will Fault Localization Work for These Failures? An Automated Approach to Predict Effectiveness of Fault Localization Tools **2013**,		23

280	Predicting Crashing Releases of Mobile Applications **2016**,		23
279	. *Empirical Software Engineering*, **2019**, 24, 1296-1327	3.3	23
278	Early prediction of merged code changes to prioritize reviewing tasks. *Empirical Software Engineering*, **2018**, 23, 3346-3393	3.3	22
277	Improving reusability of software libraries through usage pattern mining. *Journal of Systems and Software*, **2018**, 145, 164-179	3.3	22
276	Mining quantified temporal rules: Formalism, algorithms, and evaluation. *Science of Computer Programming*, **2012**, 77, 743-759	1.1	22
275	File-Level Defect Prediction: Unsupervised vs. Supervised Models **2017**,		22
274	Mining branching-time scenarios **2013**,		22
273	To what extent could we detect field defects? an empirical study of false negatives in static bug finding tools **2012**,		22
272	Interactive fault localization leveraging simple user feedback **2012**,		22
271	CDRep **2016**,		22
270	Automatic Generation of Pull Request Descriptions **2019**,		22
269	Data Quality Matters: A Case Study on Data Label Correctness for Security Bug Report Prediction. *IEEE Transactions on Software Engineering*, **2021**, 1-1	3.5	22
268	Automatic Fine-Grained Issue Report Reclassification **2014**,		21
267	Empirical Study of Usage and Performance of Java Collections **2017**,		21
266	A Comparative Study of Supervised Learning Algorithms for Re-opened Bug Prediction **2013**,		21
265	Mining message sequence graphs **2011**,		21
264	Combined classifier for cross-project defect prediction: an extended empirical study. *Frontiers of Computer Science*, **2018**, 12, 280-296	2.2	20
263	Learning to rank for bug report assignee recommendation **2016**,		20

262	WebAPIRec: Recommending Web APIs to Software Projects via Personalized Ranking. *IEEE Transactions on Emerging Topics in Computational Intelligence*, **2017**, 1, 145-156	4.1	19
261	A Large Scale Study of Multiple Programming Languages and Code Quality **2016**,		19
260	An empirical study of bug report field reassignment **2014**,		19
259	Condensing class diagrams by analyzing design and network metrics using optimistic classification **2014**,		19
258	AmaLgam+: Composing Rich Information Sources for Accurate Bug Localization. *Journal of Software: Evolution and Process*, **2016**, 28, 921-942	1	19
257	Checking Smart Contracts with Structural Code Embedding. *IEEE Transactions on Software Engineering*, **2020**, 1-1	3.5	18
256	Diversity maximization speedup for localizing faults in single-fault and multi-fault programs. *Automated Software Engineering*, **2016**, 23, 43-75	1.5	18
255	Automated Bug Report Field Reassignment and Refinement Prediction. *IEEE Transactions on Reliability*, **2016**, 65, 1094-1113	4.6	18
254	Hierarchical Parallel Algorithm for Modularity-Based Community Detection Using GPUs. *Lecture Notes in Computer Science*, **2013**, 775-787	0.9	18
253	Adoption of Software Testing in Open Source Projects--A Preliminary Study on 50,000 Projects **2013**,		18
252	Who Will Leave the Company?: A Large-Scale Industry Study of Developer Turnover by Mining Monthly Work Report **2017**,		18
251	SEWordSim: software-specific word similarity database **2014**,		18
250	Diversity maximization speedup for fault localization **2012**,		18
249	Mining past-time temporal rules from execution traces **2008**,		18
248	Extracting paraphrases of technical terms from noisy parallel software corpora **2009**,		18
247	Enhancing Automated Program Repair with Deductive Verification **2016**,		18
246	On the unreliability of bug severity data. *Empirical Software Engineering*, **2016**, 21, 2298-2323	3.3	18
245	Will this localization tool be effective for this bug? Mitigating the impact of unreliability of information retrieval based bug localization tools. *Empirical Software Engineering*, **2017**, 22, 2237-2279	3.3	17

244	NIRMAL: Automatic identification of software relevant tweets leveraging language model **2015**,		17
243	Evaluating defect prediction approaches using a massive set of metrics **2015**,		17
242	DupFinder **2014**,		17
241	Cross-language bug localization **2014**,		17
240	Collective Churn Prediction in Social Network **2012**,		17
239	**2009**,		17
238	Mining direct antagonistic communities in explicit trust networks **2011**,		17
237	A deeper look into bug fixes **2016**,		17
236	SmartEmbed: A Tool for Clone and Bug Detection in Smart Contracts through Structural Code Embedding **2019**,		17
235	PerfLearner: learning from bug reports to understand and generate performance test frames **2018**,		17
234	To the attention of mobile software developers: guess what, test your app!. *Empirical Software Engineering*, **2019**, 24, 2438-2468	3.3	16
233	Beyond support and confidence: Exploring interestingness measures for rule-based specification mining **2015**,		16
232	Augmenting and structuring user queries to support efficient free-form code search. *Empirical Software Engineering*, **2018**, 23, 2622-2654	3.3	16
231	The Impact of Mislabeled Changes by SZZ on Just-in-Time Defect Prediction. *IEEE Transactions on Software Engineering*, **2019**, 1-1	3.5	16
230	An effective change recommendation approach for supplementary bug fixes. *Automated Software Engineering*, **2017**, 24, 455-498	1.5	16
229	CNL: Collective Network Linkage Across Heterogeneous Social Platforms **2015**,		16
228	Semantic patch inference **2012**,		16
227	Observatory of trends in software related microblogs **2012**,		16

226	Revisiting Supervised and Unsupervised Methods for Effort-Aware Cross-Project Defect Prediction. *IEEE Transactions on Software Engineering*, **2020**, 1-1	3.5	16
225	APIBot: Question answering bot for API documentation **2017**,		15
224	On Locating Malicious Code in Piggybacked Android Apps. *Journal of Computer Science and Technology*, **2017**, 32, 1108-1124	1.7	15
223	RCLinker: Automated Linking of Issue Reports and Commits Leveraging Rich Contextual Information **2015**,		15
222	Mining direct antagonistic communities in signed social networks. *Information Processing and Management*, **2013**, 49, 773-791	6.3	15
221	**2011**,		15
220	Automating Intention Mining. *IEEE Transactions on Software Engineering*, **2020**, 46, 1098-1119	3.5	15
219	Characterizing malicious Android apps by mining topic-specific data flow signatures. *Information and Software Technology*, **2017**, 90, 27-39	3.4	14
218	SATD detector **2018**,		14
217	Deep Transfer Bug Localization. *IEEE Transactions on Software Engineering*, **2019**, 1-1	3.5	14
216	An Empirical Study of Bugs in Software Build Systems **2013**,		14
215	VuRLE: Automatic Vulnerability Detection and Repair by Learning from Examples. *Lecture Notes in Computer Science*, **2017**, 229-246	0.9	14
214	Should fixing these failures be delegated to automated program repair? **2015**,		14
213	**2015**,		14
212	Towards more accurate content categorization of API discussions **2014**,		14
211	CC2Vec **2020**,		14
210	How Practitioners Perceive the Relevance of ESEM Research **2016**,		14
209	Assessing the Generalizability of Code2vec Token Embeddings **2019**,		14

208	Network-Clustered Multi-Modal Bug Localization. *IEEE Transactions on Software Engineering*, **2019**, 45, 1002-1023	3.5	14
207	DEFECTCHECKER: Automated Smart Contract Defect Detection by Analyzing EVM Bytecode. *IEEE Transactions on Software Engineering*, **2021**, 1-1	3.5	14
206	Domain-specific cross-language relevant question retrieval **2016**,		13
205	Understanding inactive yet available assignees in GitHub. *Information and Software Technology*, **2017**, 91, 44-55	3.4	13
204	Mining Quantified Temporal Rules: Formalism, Algorithms, and Evaluation **2009**,		13
203	Efficient Mining of Recurrent Rules from a Sequence Database **2008**, 67-83		13
202	Empirical Study on Synthesis Engines for Semantics-Based Program Repair **2016**,		13
201	How Practitioners Perceive Automated Bug Report Management Techniques. *IEEE Transactions on Software Engineering*, **2020**, 46, 836-862	3.5	13
200	Cataloging GitHub Repositories **2017**,		12
199	Automatic Defect Categorization Based on Fault Triggering Conditions **2014**,		12
198	Code Coverage and Postrelease Defects: A Large-Scale Study on Open Source Projects. *IEEE Transactions on Reliability*, **2017**, 66, 1213-1228	4.6	12
197	Automatically Locating Malicious Packages in Piggybacked Android Apps **2017**,		12
196	TagCombine: Recommending Tags to Contents in Software Information Sites. *Journal of Computer Science and Technology*, **2015**, 30, 1017-1035	1.7	12
195	An Exploratory Study on Software Microblogger Behaviors **2014**,		12
194	Mining closed discriminative dyadic sequential patterns **2011**,		12
193	Specification mining of symbolic scenario-based models **2008**,		12
192	A Large Scale Study of Long-Time Contributor Prediction for GitHub Projects. *IEEE Transactions on Software Engineering*, **2021**, 47, 1277-1298	3.5	12
191	On the usefulness of ownership metrics in open-source software projects. *Information and Software Technology*, **2015**, 64, 102-112	3.4	11

190	Just-In-Time Defect Identification and Localization: A Two-Phase Framework. *IEEE Transactions on Software Engineering*, **2020**, 1-1	3.5	11
189	Predicting Effectiveness of IR-Based Bug Localization Techniques **2014**,		11
188	Inferring class level specifications for distributed systems **2012**,		11
187	Are faults localizable? **2012**,		11
186	kb-anonymity **2011**,		11
185	Predicting Best Answerers for New Questions: An Approach Leveraging Topic Modeling and Collaborative Voting. *Lecture Notes in Computer Science*, **2014**, 55-68	0.9	11
184	Automating App Review Response Generation **2019**,		11
183	Which Variables Should I Log?. *IEEE Transactions on Software Engineering*, **2019**, 1-1	3.5	11
182	Emerging App Issue Identification from User Feedback: Experience on WeChat **2019**,		10
181	Automatic query reformulation for code search using crowdsourced knowledge. *Empirical Software Engineering*, **2019**, 24, 1869-1924	3.3	10
180	Characterizing and identifying reverted commits. *Empirical Software Engineering*, **2019**, 24, 2171-2208	3.3	10
179	RACK: Code Search in the IDE Using Crowdsourced Knowledge **2017**,		10
178	Inference of development activities from interaction with uninstrumented applications. *Empirical Software Engineering*, **2018**, 23, 1313-1351	3.3	10
177	Deep specification mining **2018**,		10
176	Analyzing requirements and traceability information to improve bug localization **2018**,		10
175	Automatic recovery of root causes from bug-fixing changes **2013**,		10
174	**2012**,		10
173	Smart Contract Repair. *ACM Transactions on Software Engineering and Methodology*, **2020**, 29, 1-32	3.3	10

172	Inferring Links between Concerns and Methods with Multi-abstraction Vector Space Model **2016**,		10
171	How android app developers manage power consumption? **2016**,		10
170	BIKER: a tool for Bi-information source based API method recommendation **2019**,		9
169	An evaluation of pure spectrum-based fault localization techniques for large-scale software systems. *Software - Practice and Experience*, **2019**, 49, 1197-1224	2.5	9
168	Rule-based specification mining leveraging learning to rank. *Automated Software Engineering*, **2018**, 25, 501-530	1.5	9
167	It Takes Two to Tango: Deleted Stack Overflow Question Prediction with Text and Meta Features **2016**,		9
166	Augmenting and structuring user queries to support efficient free-form code search **2018**,		9
165	Automatic Code Review by Learning the Revision of Source Code. *Proceedings of the AAAI Conference on Artificial Intelligence*, **2019**, 33, 4910-4917	5	9
164	Recommending New Features from Mobile App Descriptions. *ACM Transactions on Software Engineering and Methodology*, **2019**, 28, 1-29	3.3	9
163	Automated Android application permission recommendation. *Science China Information Sciences*, **2017**, 60, 1	3.4	9
162	Query expansion via WordNet for effective code search **2015**,		9
161	An Empirical Study on the Adequacy of Testing in Open Source Projects **2014**,		9
160	What does software engineering community microblog about? **2012**,		9
159	HuMan: Creating memorable fingerprints of mobile users **2012**,		9
158	Multi-abstraction Concern Localization **2013**,		9
157	Search-based fault localization **2011**,		9
156	Sentiment Analysis for Software Engineering: How Far Can Pre-trained Transformer Models Go? **2020**,		9
155	How Practitioners Perceive Coding Proficiency **2019**,		8

154	Software Internationalization and Localization: An Industrial Experience **2013**,		8
153	Predicting Project Outcome Leveraging Socio-Technical Network Patterns **2013**,		8
152	Understanding Android App Piggybacking **2017**,		8
151	An empirical assessment of Bellon's clone benchmark **2015**,		8
150	Understanding task-driven information flow in collaborative networks **2012**,		8
149	Saffron. *Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM*, **2019**, 44, 14-14	0.4	8
148	BugsInPy: a database of existing bugs in Python programs to enable controlled testing and debugging studies **2020**,		8
147	F-Trail: Finding Patterns in Taxi Trajectories. *Lecture Notes in Computer Science*, **2013**, 86-98	0.9	8
146	Condensing Class Diagrams With Minimal Manual Labeling Cost **2016**,		8
145	Automated Identification of High Impact Bug Reports Leveraging Imbalanced Learning Strategies **2016**,		8
144	VT-Revolution: Interactive Programming Video Tutorial Authoring and Watching System. *IEEE Transactions on Software Engineering*, **2019**, 45, 823-838	3.5	8
143	Using finite-state models for log differencing **2018**,		8
142	Mining Sandboxes for Linux Containers **2017**,		7
141	Measuring the Declared SDK Versions and Their Consistency with API Calls in Android Apps. *Lecture Notes in Computer Science*, **2017**, 678-690	0.9	7
140	Build Predictor: More Accurate Missed Dependency Prediction in Build Configuration Files **2014**,		7
139	EFSPredictor: Predicting Configuration Bugs with Ensemble Feature Selection **2015**,		7
138	To what extent could we detect field defects? An extended empirical study of false negatives in static bug-finding tools. *Automated Software Engineering*, **2015**, 22, 561-602	1.5	7
137	Constrained feature selection for localizing faults **2015**,		7

136	It's not a bug, it's a feature: does misclassification affect bug localization? **2014**,		7
135	Build system analysis with link prediction **2014**,		7
134	NORT: Runtime Anomaly-Based Monitoring of Malicious Behavior for Windows. *Lecture Notes in Computer Science*, **2012**, 115-130	0.9	7
133	Orion: A Software Project Search Engine with Integrated Diverse Software Artifacts **2013**,		7
132	Mining Antagonistic Communities from Social Networks. *Lecture Notes in Computer Science*, **2010**, 68-80	0.9	7
131	What Permissions Should This Android App Request? **2016**,		7
130	A first look at unfollowing behavior on GitHub. *Information and Software Technology*, **2019**, 105, 150-160	3.4	7
129	Measuring program comprehension **2018**,		7
128	Prediction of relatedness in stack overflow: deep learning vs. SVM **2018**,		7
127	On detecting maximal quasi antagonistic communities in signed graphs. *Data Mining and Knowledge Discovery*, **2016**, 30, 99-146	5.6	6
126	Harnessing Twitter to support serendipitous learning of developers **2017**,		6
125	Revisiting Assert Use in GitHub Projects **2017**,		6
124	Who should make decision on this pull request? Analyzing time-decaying relationships and file similarities for integrator prediction. *Journal of Systems and Software*, **2019**, 154, 196-210	3.3	6
123	Active refinement of clone anomaly reports **2012**,		6
122	Mining Iterative Generators and Representative Rules for Software Specification Discovery. *IEEE Transactions on Knowledge and Data Engineering*, **2011**, 23, 282-296	4.2	6
121	Effort-aware just-in-time defect identification in practice: a case study at Alibaba **2020**,		6
120	Instance-Based Parameter Tuning via Search Trajectory Similarity Clustering. *Lecture Notes in Computer Science*, **2011**, 131-145	0.9	6
119	ORPLocator: Identifying Read Points of Configuration Options via Static Analysis **2016**,		6

118	Why reinventing the wheels? An empirical study on library reuse and re-implementation. *Empirical Software Engineering*, **2020**, 25, 755-789	3.3	6
117	INFAR: insight extraction from app reviews **2018**,		6
116	Practitioners' Views on Good Software Testing Practices **2019**,		5
115	Automatic, highly accurate app permission recommendation. *Automated Software Engineering*, **2019**, 26, 241-274	1.5	5
114	Automated Deprecated-API Usage Update for Android Apps: How Far are We? **2020**,		5
113	**2018**,		5
112	Finding needles in a haystack: Leveraging co-change dependencies to recommend refactorings. *Journal of Systems and Software*, **2019**, 158, 110420	3.3	5
111	XSearch: a domain-specific cross-language relevant question retrieval tool **2017**,		5
110	The Impact of Coverage on Bug Density in a Large Industrial Software Project **2017**,		5
109	Clustering Classes in Packages for Program Comprehension. *Scientific Programming*, **2017**, 2017, 1-15	1.4	5
108	BugLocalizer: integrated tool support for bug localization **2014**,		5
107	Dynamic Inference of Change Contracts **2014**,		5
106	An empirical study of bugs in build process **2014**,		5
105	Generating Question Titles for Stack Overflow from Mined Code Snippets. *ACM Transactions on Software Engineering and Methodology*, **2020**, 29, 1-37	3.3	5
104	AUSearch: Accurate API Usage Search in GitHub Repositories with Type Resolution **2020**,		5
103	Smart Contract Security: A Practitioners' Perspective **2021**,		5
102	Code Comment Quality Analysis and Improvement Recommendation: An Automated Approach. *International Journal of Software Engineering and Knowledge Engineering*, **2016**, 26, 981-1000	1	5
101	Statistical Log Differencing **2019**,		5

100	Overfitting in semantics-based automated program repair **2018**,		5
99	AnswerBot: an answer summary generation tool based on stack overflow **2019**,		4
98	Why is my code change abandoned?. *Information and Software Technology*, **2019**, 110, 108-120	3.4	4
97	AutoQuery: automatic construction of dependency queries for code search. *Automated Software Engineering*, **2016**, 23, 393-425	1.5	4
96	Domain-specific cross-language relevant question retrieval. *Empirical Software Engineering*, **2018**, 23, 1084-1122	3.3	4
95	Understanding Widespread Changes: A Taxonomic Study **2013**,		4
94	Personality and Project Success: Insights from a Large-Scale Study with Professionals **2017**,		4
93	An Empirical Study of Bug Fixing Rate **2015**,		4
92	AppMoD **2019**, 3, 1-22		4
91	psc2code. *ACM Transactions on Software Engineering and Methodology*, **2020**, 29, 1-38	3.3	4
90	Recommending Code Changes for Automatic Backporting of Linux Device Drivers **2016**,		4
89	Memory and resource leak defects and their repairs in Java projects. *Empirical Software Engineering* , **2020**, 25, 678-718	3.3	4
88	DeepReview: Automatic Code Review Using Deep Multi-instance Learning. *Lecture Notes in Computer Science*, **2019**, 318-330	0.9	3
87	Customer satisfaction feedback in an IT outsourcing company **2015**,		3
86	Prevalence, Contents and Automatic Detection of KL-SATD **2020**,		3
85	Practical and effective sandboxing for Linux containers. *Empirical Software Engineering*, **2019**, 24, 4034-4070	3.3	3
84	GitHub and Stack Overflow: Analyzing Developer Interests Across Multiple Social Collaborative Platforms. *Lecture Notes in Computer Science*, **2017**, 245-256	0.9	3
83	R-energy for evaluating robustness of dynamic networks **2013**,		3

82	LM **2010**,		3
81	An automated approach for finding variable-constant pairing bugs **2010**,		3
80	Towards Succinctness in Mining Scenario-Based Specifications **2011**,		3
79	Bug Signature Minimization and Fusion **2011**,		3
78	In-game action list segmentation and labeling in real-time strategy games **2012**,		3
77	Mining patterns and rules for software specification discovery. *Proceedings of the VLDB Endowment* , **2008**, 1, 1609-1616	3.1	3
76	BiasFinder: Metamorphic Test Generation to Uncover Bias for Sentiment Analysis Systems. *IEEE Transactions on Software Engineering*, **2021**, 1-1	3.5	3
75	PatchNet: Hierarchical Deep Learning-Based Stable Patch Identification for the Linux Kernel. *IEEE Transactions on Software Engineering*, **2020**, 1-1	3.5	3
74	. *IEEE Transactions on Software Engineering*, **2020**, 1-1	3.5	3
73	Why My Code Summarization Model Does Not Work. *ACM Transactions on Software Engineering and Methodology*, **2021**, 30, 1-29	3.3	3
72	CareerMapper: An automated resume evaluation tool **2016**,		3
71	. *IEEE Transactions on Software Engineering*, **2021**, 1-1	3.5	3
70	Modeling Functional Similarity in Source Code with Graph-Based Siamese Networks. *IEEE Transactions on Software Engineering*, **2021**, 1-1	3.5	3
69	A Deep Dive into the Impact of COVID-19 on Software Development. *IEEE Transactions on Software Engineering*, **2021**, 1-1	3.5	3
68	Wisdom in Sum of Parts **2018**,		3
67	DSM: a specification mining tool using recurrent neural network based language model **2018**,		3
66	Recommending Who to Follow in the Software Engineering Twitter Space. *ACM Transactions on Software Engineering and Methodology*, **2018**, 27, 1-33	3.3	3
65	Deep-Learning-Based App Sensitive Behavior Surveillance for Android Powered Cyber-Physical Systems. *IEEE Transactions on Industrial Informatics*, **2021**, 17, 5840-5850	11.9	3

64	BiasRV: uncovering biased sentiment predictions at runtime **2021**,		3
63	Technical Q8A Site Answer Recommendation via Question Boosting. *ACM Transactions on Software Engineering and Methodology*, **2021**, 30, 1-34	3.3	3
62	PatchNet: A Tool for Deep Patch Classification **2019**,		2
61	Scalable Parallelization of Specification Mining Using Distributed Computing **2015**, 623-648		2
60	Fusing multi-abstraction vector space models for concern localization. *Empirical Software Engineering*, **2018**, 23, 2279-2322	3.3	2
59	An Exploratory Study of Functionality and Learning Resources of Web APIs on ProgrammableWeb **2017**,		2
58	Experience report: An industrial experience report on test outsourcing practices **2015**,		2
57	Mining scenario-based specifications with value-based invariants **2009**,		2
56	kbe-anonymity: test data anonymization for evolving programs **2012**,		2
55	**2020**,		2
54	kb -anonymity. *ACM SIGPLAN Notices*, **2011**, 46, 447-457	0.2	2
53	A Machine Learning Approach for Vulnerability Curation **2020**,		2
52	An Empirical Study of Release Note Production and Usage in Practice. *IEEE Transactions on Software Engineering*, **2020**, 1-1	3.5	2
51	Automatic Android Deprecated-API Usage Update by Learning from Single Updated Example **2020**,		2
50	JITO: a tool for just-in-time defect identification and localization **2020**,		2
49	Why Do Smart Contracts Self-Destruct? Investigating the Selfdestruct Function on Ethereum. *ACM Transactions on Software Engineering and Methodology*, **2022**, 31, 1-37	3.3	2
48	Scalable online vetting of Android apps for measuring declared SDK versions and their consistency with API calls. *Empirical Software Engineering*, **2021**, 26, 1	3.3	2
47	A Survey on Deep Learning for Software Engineering. *ACM Computing Surveys*,	13.4	2

46	Distinguishing Similar Design Pattern Instances through Temporal Behavior Analysis **2020**,		1
45	Recommending frequently encountered bugs **2018**,		1
44	Inferring Spread of Readers Emotion Affected by Online News. *Lecture Notes in Computer Science*, **2017**, 426-439	0.9	1
43	BOAT: an experimental platform for researchers to comparatively and reproducibly evaluate bug localization techniques **2014**,		1
42	Clustering of search trajectory and its application to parameter tuning. *Journal of the Operational Research Society*, **2013**, 64, 1742-1752	2	1
41	Bidirectional mining of non-redundant recurrent rules from a sequence database **2011**,		1
40	Automated identification of libraries from vulnerability data **2020**,		1
39	An exploratory study on the repeatedly shared external links on Stack Overflow. *Empirical Software Engineering*, **2022**, 27, 1	3.3	1
38	Memory and resource leak defects and their repairs in Java projects **2020**, 25, 678		1
37	Spiteful, One-Off, and Kind: Predicting Customer Feedback Behavior on Twitter. *Lecture Notes in Computer Science*, **2016**, 368-381	0.9	1
36	Mining Past-Time Temporal Rules. *Advances in Computational Intelligence and Robotics Book Series*, **2010**, 259-277	0.4	1
35	Out of sight, out of mind? How vulnerable dependencies affect open-source projects. *Empirical Software Engineering*, **2021**, 26, 1	3.3	1
34	Unveiling the Mystery of API Evolution in Deep Learning Frameworks: A Case Study of Tensorflow 2 **2021**,		1
33	SIEVE: Helping developers sift wheat from chaff via cross-platform analysis. *Empirical Software Engineering*, **2020**, 25, 996-1030	3.3	1
32	Broken External Links on Stack Overflow. *IEEE Transactions on Software Engineering*, **2021**, 1-1	3.5	1
31	Emerging App Issue Identification via Online Joint Sentiment-Topic Tracing. *IEEE Transactions on Software Engineering*, **2021**, 1-1	3.5	1
30	Active Learning of Discriminative Subgraph Patterns for API Misuse Detection. *IEEE Transactions on Software Engineering*, **2021**, 1-1	3.5	1
29	Including Everyone, Everywhere: Understanding Opportunities and Challenges of Geographic Gender-Inclusion in OSS. *IEEE Transactions on Software Engineering*, **2021**, 1-1	3.5	1

28	**2018**,		1
27	Characterizing Common and Domain-Specific Package Bugs: A Case Study on Ubuntu **2018**,		1
26	Context-aware Retrieval-based Deep Commit Message Generation. *ACM Transactions on Software Engineering and Methodology*, **2021**, 30, 1-30	3.3	1
25	Maintenance-related concerns for post-deployed Ethereum smart contract development: issues, techniques, and future challenges. *Empirical Software Engineering*, **2021**, 26, 1	3.3	1
24	What makes a popular academic AI repository?. *Empirical Software Engineering*, **2021**, 26, 1	3.3	1
23	An Empirical Study of Bugs in Software Build System. *IEICE Transactions on Information and Systems*, **2014**, E97.D, 1769-1780	0.6	0
22	Just-In-Time Obsolete Comment Detection and Update. *IEEE Transactions on Software Engineering*, **2021**, 1-1	3.5	0
21	Understanding in-app advertising issues based on large scale app review analysis. *Information and Software Technology*, **2022**, 142, 106741	3.4	0
20	Automating App Review Response Generation Based on Contextual Knowledge. *ACM Transactions on Software Engineering and Methodology*, **2022**, 31, 1-36	3.3	0
19	On the Reproducibility and Replicability of Deep Learning in Software Engineering. *ACM Transactions on Software Engineering and Methodology*, **2022**, 31, 1-46	3.3	0
18	Opportunities and Challenges in Code Search Tools. *ACM Computing Surveys*, **2022**, 54, 1-40	13.4	0
17	Improving Software Quality and Productivity Leveraging Mining Techniques. *Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM*, **2015**, 40, 1-2	0.4	0
16	Adversarial Specification Mining. *ACM Transactions on Software Engineering and Methodology*, **2021**, 30, 1-40	3.3	0
15	Watch out for this commit! A study of influential software changes. *Journal of Software: Evolution and Process*, **2019**, 31, e2181	1	0
14	Legion: Massively Composing Rankers for Improved Bug Localization at Adobe. *IEEE Transactions on Software Engineering*, **2021**, 1-1	3.5	0
13	An exploratory study on the introduction and removal of different types of technical debt in deep learning frameworks. *Empirical Software Engineering*, **2021**, 26, 1	3.3	0
12	CodeMatcher: Searching Code Based on Sequential Semantics of Important Query Words. *ACM Transactions on Software Engineering and Methodology*, **2022**, 31, 1-37	3.3	0
11	Web APIs: Features, Issues, and Expectations -- A Large-Scale Empirical Study of Web APIs from Two Publicly Accessible Registries Using Stack Overflow and A User Survey. *IEEE Transactions on Software Engineering*, **2022**, 1-1	3.5	0

10	Leveraging machine learning and information retrieval techniques in software evolution tasks. *Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM*, **2014**, 39, 1-2	0.4
9	The Java Pathfinder Workshop 2019. *Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM*, **2020**, 45, 20-22	0.4
8	Deep Just-In-Time Defect Localization. *IEEE Transactions on Software Engineering*, **2021**, 1-1	3.5
7	Social Collaborative Media in Software Development **2017**, 1-7	
6	Mining Software Specifications **2009**, 1303-1309	
5	Specification Mining. *Chapman & Hall/CRC Data Mining and Knowledge Discovery Series*, **2011**, 1-27	
4	Sequence-to-Sequence Learning for Automated Software Artifact Generation **2021**, 111-140	
3	Preface to the Special Issue on Program Comprehension. *Empirical Software Engineering*, **2019**, 24, 208-210	3.9
2	Orchestration or Automation: Authentication Flaw Detection in Android Apps. *IEEE Transactions on Dependable and Secure Computing*, **2021**, 1-1	3.9
1	AndroEvolve: automated Android API update with data flow analysis and variable denormalization. *Empirical Software Engineering*, **2022**, 27, 1	3.3