Xin Xia

List of Publications by Citations

188 papers	3,541 citations	33 h-index	50 g-index
217 ext. papers	5,621 ext. citations	2.9 avg, IF	6.09 L-index

#	Paper	IF	Citations
188	HYDRA: Massively Compositional Model for Cross-Project Defect Prediction. *IEEE Transactions on Software Engineering*, **2016**, 42, 977-998	3.5	148
187	Deep Learning for Just-in-Time Defect Prediction **2015**,		133
186	Deep code comment generation **2018**,		115
185	TLEL: A two-layer ensemble learning approach for just-in-time defect prediction. *Information and Software Technology*, **2017**, 87, 206-220	3.4	94
184	Practitioners' expectations on automated fault localization **2016**,		92
183	Predicting semantically linkable knowledge in developer online forums via convolutional neural network **2016**,		80
182	**2013**,		77
181	. *IEEE Transactions on Software Engineering*, **2019**, 1-1	3.5	75
180	Improving Automated Bug Triaging with Specialized Topic Model. *IEEE Transactions on Software Engineering*, **2017**, 43, 272-297	3.5	72
179	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
178	Identifying self-admitted technical debt in open source projects using text mining. *Empirical Software Engineering*, **2018**, 23, 418-451	3.3	67
177	ELBlocker: Predicting blocking bugs with ensemble imbalance learning. *Information and Software Technology*, **2015**, 61, 93-106	3.4	65
176	Automated prediction of bug report priority using multi-factor analysis. *Empirical Software Engineering*, **2015**, 20, 1354-1383	3.3	56
175	What do developers search for on the web?. *Empirical Software Engineering*, **2017**, 22, 3149-3185	3.3	54
174	**2017**,		54
173	Multi-Factor Duplicate Question Detection in Stack Overflow. *Journal of Computer Science and Technology*, **2015**, 30, 981-997	1.7	53
172	Accurate developer recommendation for bug resolution **2013**,		52

171	API method recommendation without worrying about the task-API knowledge gap **2018**,		50
170	An Empirical Study of Classifier Combination for Cross-Project Defect Prediction **2015**,		49
169	Neural-machine-translation-based commit message generation: how far are we? **2018**,		49
168	Why and how developers fork what from whom in GitHub. *Empirical Software Engineering*, **2017**, 22, 547-578	3.3	48
167	Measuring Program Comprehension: A Large-Scale Field Study with Professionals. *IEEE Transactions on Software Engineering*, **2018**, 44, 951-976	3.5	43
166	Automatic, high accuracy prediction of reopened bugs. *Automated Software Engineering*, **2015**, 22, 75-109	2.5	41
165	A two-phase transfer learning model for cross-project defect prediction. *Information and Software Technology*, **2019**, 107, 125-136	3.4	41
164	Summarizing Source Code with Transferred API Knowledge **2018**,		40
163	Who should review this change?: Putting text and file location analyses together for more accurate recommendations **2015**,		38
162	Combining Word Embedding with Information Retrieval to Recommend Similar Bug Reports **2016**,		38
161	Perceptions, Expectations, and Challenges in Defect Prediction. *IEEE Transactions on Software Engineering*, **2020**, 46, 1241-1266	3.5	37
160	Neural Network-based Detection of Self-Admitted Technical Debt. *ACM Transactions on Software Engineering and Methodology*, **2019**, 28, 1-45	3.3	36
159	Revisiting supervised and unsupervised models for effort-aware just-in-time defect prediction. *Empirical Software Engineering*, **2019**, 24, 2823-2862	3.3	36
158	Improving defect prediction with deep forest. *Information and Software Technology*, **2019**, 114, 204-216	3.4	35
157	Bug Characteristics in Blockchain Systems: A Large-Scale Empirical Study **2017**,		35
156	Dual analysis for recommending developers to resolve bugs. *Journal of Software: Evolution and Process*, **2015**, 27, 195-220	1	34
155	**2017**,		32
154	Automated Configuration Bug Report Prediction Using Text Mining **2014**,		31

153	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
152	Deep code comment generation with hybrid lexical and syntactical information. *Empirical Software Engineering*, **2020**, 25, 2179-2217	3.3	30
151	Combining Software Metrics and Text Features for Vulnerable File Prediction **2015**,		29
150	High-Impact Bug Report Identification with Imbalanced Learning Strategies. *Journal of Computer Science and Technology*, **2017**, 32, 181-198	1.7	28
149	How does Machine Learning Change Software Development Practices?. *IEEE Transactions on Software Engineering*, **2020**, 1-1	3.5	28
148	. *IEEE Transactions on Software Engineering*, **2019**, 1-1	3.5	27
147	. *IEEE Transactions on Reliability*, **2016**, 65, 1810-1829	4.6	27
146	Defining Smart Contract Defects on Ethereum. *IEEE Transactions on Software Engineering*, **2020**, 1-1	3.5	26
145	Towards more accurate multi-label software behavior learning **2014**,		26
144	Detecting similar repositories on GitHub **2017**,		25
143	Automating Change-Level Self-Admitted Technical Debt Determination. *IEEE Transactions on Software Engineering*, **2019**, 45, 1211-1229	3.5	25
142	Enhancing developer recommendation with supplementary information via mining historical commits. *Journal of Systems and Software*, **2017**, 134, 355-368	3.3	24
141	Fusion fault localizers **2014**,		24
140	Chaff from the Wheat: Characterizing and Determining Valid Bug Reports. *IEEE Transactions on Software Engineering*, **2020**, 46, 495-525	3.5	24
139	Cross-project build co-change prediction **2015**,		23
138	Predicting Crashing Releases of Mobile Applications **2016**,		23
137	Characterising deprecated Android APIs **2018**,		23
136	Early prediction of merged code changes to prioritize reviewing tasks. *Empirical Software Engineering*, **2018**, 23, 3346-3393	3.3	22

135	File-Level Defect Prediction: Unsupervised vs. Supervised Models **2017**,		22
134	Automatic Generation of Pull Request Descriptions **2019**,		22
133	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
132	A Comparative Study of Supervised Learning Algorithms for Re-opened Bug Prediction **2013**,		21
131	Information Credibility on Twitter in Emergency Situation. *Lecture Notes in Computer Science*, **2012**, 45-59	0.9	21
130	Combined classifier for cross-project defect prediction: an extended empirical study. *Frontiers of Computer Science*, **2018**, 12, 280-296	2.2	20
129	An empirical study of bug report field reassignment **2014**,		19
128	Checking Smart Contracts with Structural Code Embedding. *IEEE Transactions on Software Engineering*, **2020**, 1-1	3.5	18
127	Diversity maximization speedup for localizing faults in single-fault and multi-fault programs. *Automated Software Engineering*, **2016**, 23, 43-75	1.5	18
126	Automated Bug Report Field Reassignment and Refinement Prediction. *IEEE Transactions on Reliability*, **2016**, 65, 1094-1113	4.6	18
125	Who Will Leave the Company?: A Large-Scale Industry Study of Developer Turnover by Mining Monthly Work Report **2017**,		18
124	Evaluating defect prediction approaches using a massive set of metrics **2015**,		17
123	Diversified Third-party Library Prediction for Mobile App Development. *IEEE Transactions on Software Engineering*, **2020**, 1-1	3.5	17
122	Cross-language bug localization **2014**,		17
121	SmartEmbed: A Tool for Clone and Bug Detection in Smart Contracts through Structural Code Embedding **2019**,		17
120	Extracting and analyzing time-series HCI data from screen-captured task videos. *Empirical Software Engineering*, **2017**, 22, 134-174	3.3	16
119	Personalized project recommendation on GitHub. *Science China Information Sciences*, **2018**, 61, 1	3.4	16
118	The Impact of Mislabeled Changes by SZZ on Just-in-Time Defect Prediction. *IEEE Transactions on Software Engineering*, **2019**, 1-1	3.5	16

117	. *IEEE Transactions on Software Engineering*, **2019**, 1-1	3.5	16
116	An effective change recommendation approach for supplementary bug fixes. *Automated Software Engineering*, **2017**, 24, 455-498	1.5	16
115	What do Programmers Discuss about Deep Learning Frameworks. *Empirical Software Engineering*, **2020**, 25, 2694-2747	3.3	16
114	Revisiting Supervised and Unsupervised Methods for Effort-Aware Cross-Project Defect Prediction. *IEEE Transactions on Software Engineering*, **2020**, 1-1	3.5	16
113	Bug Report Enrichment with Application of Automated Fixer Recommendation **2017**,		15
112	Automating Intention Mining. *IEEE Transactions on Software Engineering*, **2020**, 46, 1098-1119	3.5	15
111	Characterizing malicious Android apps by mining topic-specific data flow signatures. *Information and Software Technology*, **2017**, 90, 27-39	3.4	14
110	ActionNet: Vision-Based Workflow Action Recognition From Programming Screencasts **2019**,		14
109	SATD detector **2018**,		14
108	An Empirical Study of Bugs in Software Build Systems **2013**,		14
107	Towards more accurate content categorization of API discussions **2014**,		14
106	DEFECTCHECKER: Automated Smart Contract Defect Detection by Analyzing EVM Bytecode. *IEEE Transactions on Software Engineering*, **2021**, 1-1	3.5	14
105	Domain-specific cross-language relevant question retrieval **2016**,		13
104	How Practitioners Perceive Automated Bug Report Management Techniques. *IEEE Transactions on Software Engineering*, **2020**, 46, 836-862	3.5	13
103	Automatic Defect Categorization Based on Fault Triggering Conditions **2014**,		12
102	TagCombine: Recommending Tags to Contents in Software Information Sites. *Journal of Computer Science and Technology*, **2015**, 30, 1017-1035	1.7	12
101	A Large Scale Study of Long-Time Contributor Prediction for GitHub Projects. *IEEE Transactions on Software Engineering*, **2021**, 47, 1277-1298	3.5	12
100	Just-In-Time Defect Identification and Localization: A Two-Phase Framework. *IEEE Transactions on Software Engineering*, **2020**, 1-1	3.5	11

99	Automating App Review Response Generation **2019**,		11
98	Which Variables Should I Log?. *IEEE Transactions on Software Engineering*, **2019**, 1-1	3.5	11
97	Characterizing and identifying reverted commits. *Empirical Software Engineering*, **2019**, 24, 2171-2208	3.3	10
96	Inference of development activities from interaction with uninstrumented applications. *Empirical Software Engineering*, **2018**, 23, 1313-1351	3.3	10
95	Inferring Links between Concerns and Methods with Multi-abstraction Vector Space Model **2016**,		10
94	How android app developers manage power consumption? **2016**,		10
93	Discovering, Explaining and Summarizing Controversial Discussions in Community Q&A Sites **2019**,		10
92	BIKER: a tool for Bi-information source based API method recommendation **2019**,		9
91	It Takes Two to Tango: Deleted Stack Overflow Question Prediction with Text and Meta Features **2016**,		9
90	Automated Android application permission recommendation. *Science China Information Sciences*, **2017**, 60, 1	3.4	9
89	ActivitySpace: A Remembrance Framework to Support Interapplication Information Needs **2015**,		9
88	Combining Collaborative Filtering and Topic Modeling for More Accurate Android Mobile App Library Recommendation **2017**,		8
87	How Practitioners Perceive Coding Proficiency **2019**,		8
86	Characterization and Prediction of Popular Projects on GitHub **2019**,		8
85	Software Internationalization and Localization: An Industrial Experience **2013**,		8
84	Wireframe-based UI Design Search through Image Autoencoder. *ACM Transactions on Software Engineering and Methodology*, **2020**, 29, 1-31	3.3	8
83	Condensing Class Diagrams With Minimal Manual Labeling Cost **2016**,		8
82	Automated Identification of High Impact Bug Reports Leveraging Imbalanced Learning Strategies **2016**,		8

81	**2019**,		8
80	VT-Revolution: Interactive Programming Video Tutorial Authoring and Watching System. *IEEE Transactions on Software Engineering*, **2019**, 45, 823-838	3.5	8
79	Mining Sandboxes for Linux Containers **2017**,		7
78	Build Predictor: More Accurate Missed Dependency Prediction in Build Configuration Files **2014**,		7
77	EFSPredictor: Predicting Configuration Bugs with Ensemble Feature Selection **2015**,		7
76	Build system analysis with link prediction **2014**,		7
75	What Permissions Should This Android App Request? **2016**,		7
74	Measuring program comprehension **2018**,		7
73	What design topics do developers discuss? **2018**,		7
72	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
71	RW.KNN **2011**,		6
70	Effort-aware just-in-time defect identification in practice: a case study at Alibaba **2020**,		6
69	**2018**,		6
68	Automatic, highly accurate app permission recommendation. *Automated Software Engineering*, **2019**, 26, 241-274	1.5	5
67	Locating Latent Design Information in Developer Discussions: A Study on Pull Requests. *IEEE Transactions on Software Engineering*, **2019**, 1-1	3.5	5
66	XSearch: a domain-specific cross-language relevant question retrieval tool **2017**,		5
65	An empirical study of bugs in build process **2014**,		5
64	Generating Question Titles for Stack Overflow from Mined Code Snippets. *ACM Transactions on Software Engineering and Methodology*, **2020**, 29, 1-37	3.3	5

63	Smart Contract Security: A Practitioners' Perspective **2021**,		5
62	How Should I Improve the UI of My App?. *ACM Transactions on Software Engineering and Methodology*, **2021**, 30, 1-38	3.3	5
61	AnswerBot: an answer summary generation tool based on stack overflow **2019**,		4
60	Software quality assessment model: a systematic mapping study. *Science China Information Sciences* , **2019**, 62, 1	3.4	4
59	Why is my code change abandoned?. *Information and Software Technology*, **2019**, 110, 108-120	3.4	4
58	CDA: Characterising Deprecated Android APIs. *Empirical Software Engineering*, **2020**, 25, 2058-2098	3.3	4
57	Domain-specific cross-language relevant question retrieval. *Empirical Software Engineering*, **2018**, 23, 1084-1122	3.3	4
56	Multitask defect prediction. *Journal of Software: Evolution and Process*, **2019**, 31, e2203	1	4
55	Personality and Project Success: Insights from a Large-Scale Study with Professionals **2017**,		4
54	A Systematic Mapping Study of Quality Assessment Models for Software Products **2017**,		4
53	An Empirical Study of Bug Fixing Rate **2015**,		4
52	psc2code. *ACM Transactions on Software Engineering and Methodology*, **2020**, 29, 1-38	3.3	4
51	A Self-Attentional Neural Architecture for Code Completion with Multi-Task Learning **2020**,		4
50	Duplicate Pull Request Detection **2019**,		4
49	Customer satisfaction feedback in an IT outsourcing company **2015**,		3
48	Practical and effective sandboxing for Linux containers. *Empirical Software Engineering*, **2019**, 24, 4034-4070	3.3	3
47	API-misuse detection driven by fine-grained API-constraint knowledge graph **2020**,		3
46	Modular Tree Network for Source Code Representation Learning. *ACM Transactions on Software Engineering and Methodology*, **2020**, 29, 1-23	3.3	3

45	Demystify official API usage directives with crowdsourced API misuse scenarios, erroneous code examples and patches **2020**,		3
44	Retrieve and refine **2020**,		3
43	Automating just-in-time comment updating **2020**,		3
42	. *IEEE Transactions on Software Engineering*, **2020**, 1-1	3.5	3
41	Why My Code Summarization Model Does Not Work. *ACM Transactions on Software Engineering and Methodology*, **2021**, 30, 1-29	3.3	3
40	**2018**,		3
39	Technical Q8A Site Answer Recommendation via Question Boosting. *ACM Transactions on Software Engineering and Methodology*, **2021**, 30, 1-34	3.3	3
38	Scalable Relevant Project Recommendation on GitHub **2017**,		2
37	Fusing multi-abstraction vector space models for concern localization. *Empirical Software Engineering*, **2018**, 23, 2279-2322	3.3	2
36	Automating Aggregation for Software Quality Modeling **2017**,		2
35	Experience report: An industrial experience report on test outsourcing practices **2015**,		2
34	An Empirical Study of Release Note Production and Usage in Practice. *IEEE Transactions on Software Engineering*, **2020**, 1-1	3.5	2
33	JITO: a tool for just-in-time defect identification and localization **2020**,		2
32	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
31	Recommending tags for pull requests in GitHub. *Information and Software Technology*, **2021**, 129, 106394	3.4	2
30	Embedding app-library graph for neural third party library recommendation **2021**,		2
29	How does working from home affect developer productivity? A case study of Baidu during the COVID-19 pandemic. *Science China Information Sciences*, **2022**, 65,	3.4	2
28	A Survey on Deep Learning for Software Engineering. *ACM Computing Surveys*,	13.4	2

27	Recommending frequently encountered bugs **2018**,		1
26	Learning to Aggregate: An Automated Aggregation Method for Software Quality Model **2017**,		1
25	BOAT: an experimental platform for researchers to comparatively and reproducibly evaluate bug localization techniques **2014**,		1
24	An exploratory study on the repeatedly shared external links on Stack Overflow. *Empirical Software Engineering*, **2022**, 27, 1	3.3	1
23	Instance-Ranking: A New Perspective to Consider the Instance Dependency for Classification. *Lecture Notes in Computer Science*, **2013**, 112-123	0.9	1
22	Plot2API: Recommending Graphic API from Plot via Semantic Parsing Guided Neural Network **2021**,		1
21	Unveiling the Mystery of API Evolution in Deep Learning Frameworks: A Case Study of Tensorflow 2 **2021**,		1
20	**2019**,		1
19	Broken External Links on Stack Overflow. *IEEE Transactions on Software Engineering*, **2021**, 1-1	3.5	1
18	Emerging App Issue Identification via Online Joint Sentiment-Topic Tracing. *IEEE Transactions on Software Engineering*, **2021**, 1-1	3.5	1
17	Helping or not helping? Why and how trivial packages impact the npm ecosystem. *Empirical Software Engineering*, **2021**, 26, 1	3.3	1
16	Characterizing Common and Domain-Specific Package Bugs: A Case Study on Ubuntu **2018**,		1
15	Context-aware Retrieval-based Deep Commit Message Generation. *ACM Transactions on Software Engineering and Methodology*, **2021**, 30, 1-30	3.3	1
14	Maintenance-related concerns for post-deployed Ethereum smart contract development: issues, techniques, and future challenges. *Empirical Software Engineering*, **2021**, 26, 1	3.3	1
13	What makes a popular academic AI repository?. *Empirical Software Engineering*, **2021**, 26, 1	3.3	1
12	A unified multi-task learning model for AST-level and token-level code completion. *Empirical Software Engineering*, **2022**, 27, 1	3.3	1
11	An Empirical Study of Bugs in Software Build System. *IEICE Transactions on Information and Systems*, **2014**, E97.D, 1769-1780	0.6	0
10	Just-In-Time Obsolete Comment Detection and Update. *IEEE Transactions on Software Engineering*, **2021**, 1-1	3.5	0

9	Understanding in-app advertising issues based on large scale app review analysis. *Information and Software Technology*, **2022**, 142, 106741	3.4	0
8	Automating App Review Response Generation Based on Contextual Knowledge. *ACM Transactions on Software Engineering and Methodology*, **2022**, 31, 1-36	3.3	0
7	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
6	Analysis of Trending Topics and Text-based Channels of Information Delivery in Cybersecurity. *ACM Transactions on Internet Technology*, **2022**, 22, 1-27	3.8	0
5	Opportunities and Challenges in Code Search Tools. *ACM Computing Surveys*, **2022**, 54, 1-40	13.4	0
4	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
3	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
2	Deep Just-In-Time Defect Localization. *IEEE Transactions on Software Engineering*, **2021**, 1-1	3.5	
1	How does Visualisation Help App Practitioners Analyse Android Apps?. *IEEE Transactions on Dependable and Secure Computing*, **2022**, 1-1	3.9	