Tse-Hsun Chen

List of Publications by Year
in descending order

38 papers	979 citations	840776 11 h-index	610901 24 g-index
39 all docs	39 docs citations	39 times ranked	501 citing authors

#	Article	IF	Citations
1	A survey on the use of topic models when mining software repositories. Empirical Software Engineering, 2016, 21, 1843-1919.	3.9	132
2	Detecting performance anti-patterns for applications developed using object-relational mapping. , 2014, , .		91
3	Logram: Efficient Log Parsing Using n-Gram Dictionaries. IEEE Transactions on Software Engineering, 2020, , 1-1.	5.6	63
4	An empirical study of dormant bugs. , 2014, , .		59
5	Studying software logging using topic models. Empirical Software Engineering, 2018, 23, 2655-2694.	3.9	52
6	CacheOptimizer: helping developers configure caching frameworks for hibernate-based database-centric web applications. , 2016, , .		40
7	Finding and Evaluating the Performance Impact of Redundant Data Access for Applications that are Developed Using Object-Relational Mapping Frameworks. IEEE Transactions on Software Engineering, 2016, 42, 1148-1161.	5.6	38
8	Understanding the factors for fast answers in technical Q&A websites. Empirical Software Engineering, 2018, 23, 1552-1593.	3.9	38
9	Studying the effectiveness of application performance management (APM) tools for detecting performance regressions for web applications. , 2016, , .		37
10	DLFinder: Characterizing and Detecting Duplicate Logging Code Smells. , 2019, , .		36
11	Studying the characteristics of logging practices in mobile apps: a case study on F-Droid. Empirical Software Engineering, 2019, 24, 3394-3434.	3.9	33
12	An Empirical Study of Obsolete Answers on Stack Overflow. IEEE Transactions on Software Engineering, 2021, 47, 850-862.	5.6	33
13	Analytics-Driven Load Testing: An Industrial Experience Report on Load Testing of Large-Scale Systems. , 2017, , .		27
14	DeepLV: Suggesting Log Levels Using Ordinal Based Neural Networks. , 2021, , .		25
15	How Do Users Revise Answers on Technical Q&A Websites? A Case Study on Stack Overflow. IEEE Transactions on Software Engineering, 2020, 46, 1024-1038.	5.6	24
16	Where shall we log?. , 2020, , .		22
17	The secret life of test smells - an empirical study on test smell evolution and maintenance. Empirical Software Engineering, 2021, 26, 1.	3.9	21
18	Topic-based software defect explanation. Journal of Systems and Software, 2017, 129, 79-106.	4.5	18

#	Article	IF	Citations
19	Reading Answers on Stack Overflow: Not Enough!. IEEE Transactions on Software Engineering, 2021, 47, 2520-2533.	5.6	17
20	Detecting problems in the database access code of large scale systems. , 2016, , .		15
21	An empirical study on the practice of maintaining object-relational mapping code in Java systems. , 2016, , .		14
22	A3: Assisting Android API Migrations Using Code Examples. IEEE Transactions on Software Engineering, 2022, 48, 417-431.	5.6	13
23	Explaining software defects using topic models. , 2012, , .		12
24	Are Comments on Stack Overflow Well Organized for Easy Retrieval by Developers?. ACM Transactions on Software Engineering and Methodology, 2021, 30, 1-31.	6.0	12
25	An Empirical Study on the Effect of Testing on Code Quality Using Topic Models: A Case Study on Software Development Systems. IEEE Transactions on Reliability, 2017, 66, 806-824.	4.6	11
26	Adopting autonomic computing capabilities in existing large-scale systems. , 2018, , .		11
27	Pathidea: Improving Information Retrieval-Based Bug Localization by Re-Constructing Execution Paths Using Logs. IEEE Transactions on Software Engineering, 2022, 48, 2905-2919.	5.6	11
28	A Study of C/C++ Code Weaknesses on Stack Overflow. IEEE Transactions on Software Engineering, 2022, 48, 2359-2375.	5.6	11
29	Understanding the factors for fast answers in technical Q&A websites. , 2018, , .		10
30	iPerfDetector: Characterizing and detecting performance anti-patterns in iOS applications. Empirical Software Engineering, 2019, 24, 3484-3513.	3.9	9
31	Demystifying the challenges and benefits of analyzing user-reported logs in bug reports. Empirical Software Engineering, 2021, 26, 1.	3.9	9
32	LogAssist: Assisting Log Analysis Through Log Summarization. IEEE Transactions on Software Engineering, 2022, 48, 3227-3241.	5.6	7
33	Improving the quality of large-scale database-centric software systems by analyzing database access code. , 2015, , .		6
34	Studying Duplicate Logging Statements and Their Relationships With Code Clones. IEEE Transactions on Software Engineering, 2022, 48, 2476-2494.	5.6	6
35	Revisiting Test Impact Analysis in Continuous Testing From the Perspective of Code Dependencies. IEEE Transactions on Software Engineering, 2022, 48, 1979-1993.	5.6	6
36	Towards Learning Generalizable Code Embeddings Using Task-agnostic Graph Convolutional Networks. ACM Transactions on Software Engineering and Methodology, 2023, 32, 1-43.	6.0	4

#	Article	IF	Citations
37	MLASP: Machine learning assisted capacity planning. Empirical Software Engineering, 2021, 26, 1.	3.9	3
38	Studying backers and hunters in bounty issue addressing process of open source projects. Empirical Software Engineering, 2021, 26, 1.	3.9	3