Ali Ouni

List of Publications by Year in descending order

98 papers	3,001 citations	331538 21 h-index	214721 47 g-index
99 all docs	99 docs citations	99 times ranked	1592 citing authors

#	Article	IF	Citations
1	Optimal Deep Learning LSTM Model for Electric Load Forecasting using Feature Selection and Genetic Algorithm: Comparison with Machine Learning Approaches â€. Energies, 2018, 11, 1636.	1.6	529
2	Many-Objective Software Remodularization Using NSGA-III. ACM Transactions on Software Engineering and Methodology, 2015, 24, 1-45.	4.8	197
3	Do developers update their library dependencies?. Empirical Software Engineering, 2018, 23, 384-417.	3.0	189
4	Multi-Sequence LSTM-RNN Deep Learning and Metaheuristics for Electric Load Forecasting. Energies, 2020, 13, 391.	1.6	122
5	Maintainability defects detection and correction: a multi-objective approach. Automated Software Engineering, 2013, 20, 47-79.	2.2	120
6	Multi-Criteria Code Refactoring Using Search-Based Software Engineering. ACM Transactions on Software Engineering and Methodology, 2016, 25, 1-53.	4.8	106
7	A Cooperative Parallel Search-Based Software Engineering Approach for Code-Smells Detection. IEEE Transactions on Software Engineering, 2014, 40, 841-861.	4.3	92
8	Design Defects Detection and Correction by Example. , 2011, , .		70
9	Search-Based Web Service Antipatterns Detection. IEEE Transactions on Services Computing, 2017, 10, 603-617.	3.2	69
10	Search-based software library recommendation using multi-objective optimization. Information and Software Technology, 2017, 83, 55-75.	3.0	66
11	Improving multi-objective code-smells correction using development history. Journal of Systems and Software, 2015, 105, 18-39.	3.3	59
12	Single and Multi-Sequence Deep Learning Models for Short and Medium Term Electric Load Forecasting. Energies, 2019, 12, 149.	1.6	51
13	Prioritizing code-smells correction tasks using chemical reaction optimization. Software Quality Journal, 2015, 23, 323-361.	1.4	49
14	Search-Based Peer Reviewers Recommendation in Modern Code Review. , 2016, , .		49
15	Search-based refactoring: Towards semantics preservation. , 2012, , .		48
16	Web service API recommendation for automated mashup creation using multi-objective evolutionary search. Applied Soft Computing Journal, 2019, 85, 105830.	4.1	47
17	Improving reusability of software libraries through usage pattern mining. Journal of Systems and Software, 2018, 145, 164-179.	3.3	45
18	tsDetect: an open source test smells detection tool. , 2020, , .		45

#	Article	IF	Citations
19	Web Service Antipatterns Detection Using Genetic Programming. , 2015, , .		42
20	Search-Based Refactoring Using Recorded Code Changes. , 2013, , .		39
21	The use of development history in software refactoring using a multi-objective evolutionary algorithm. , 2013, , .		36
22	On the Impact of Refactoring on the Relationship between Quality Attributes and Design Metrics. , 2019, , .		34
23	Can Refactoring Be Self-Affirmed? An Exploratory Study on How Developers Document Their Refactoring Activities in Commit Messages. , 2019, , .		32
24	How we refactor and how we document it? On the use of supervised machine learning algorithms to classify refactoring documentation. Expert Systems With Applications, 2021, 167, 114176.	4.4	32
25	MORE: A multiâ€objective refactoring recommendation approach to introducing design patterns and fixing code smells. Journal of Software: Evolution and Process, 2017, 29, e1843.	1.2	29
26	Recommending relevant classes for bug reports using multi-objective search. , 2016, , .		28
27	WhoReview: A multi-objective search-based approach for code reviewers recommendation in modern code review. Applied Soft Computing Journal, 2021, 100, 106908.	4.1	28
28	An Interactive and Dynamic Search-Based Approach to Software Refactoring Recommendations. IEEE Transactions on Software Engineering, 2020, 46, 932-961.	4.3	27
29	Toward the automatic classification of Self-Affirmed Refactoring. Journal of Systems and Software, 2021, 171, 110821.	3.3	27
30	Test Smell Detection Tools: A Systematic Mapping Study. , 2021, , .		27
31	An empirical study on the impact of refactoring activities on evolving client-used APIs. Information and Software Technology, 2018, 93, 186-199.	3.0	26
32	Detecting Android Smells Using Multi-Objective Genetic Programming. , 2017, , .		25
33	Learning to detect community smells in open source software projects. Knowledge-Based Systems, 2020, 204, 106201.	4.0	24
34	Search-based metamodel matching with structural and syntactic measures. Journal of Systems and Software, 2014, 97, 1-14.	3.3	23
35	Learning to recommend third-party library migration opportunities at the API level. Applied Soft Computing Journal, 2020, 90, 106140.	4.1	21
36	How do i refactor this? An empirical study on refactoring trends and topics in Stack Overflow. Empirical Software Engineering, 2022, 27, 1.	3.0	21

#	Article	IF	Citations
37	Predicting continuous integration build failures using evolutionary search. Information and Software Technology, 2020, 128, 106392.	3.0	20
38	Refactoring Practices in the Context of Modern Code Review: An Industrial Case Study at Xerox. , 2021, , .		19
39	On preserving the behavior in software refactoring: A systematic mapping study. Information and Software Technology, 2021, 140, 106675.	3.0	19
40	On the detection of community smells using genetic programming-based ensemble classifier chain. , 2020, , .		19
41	On the Use of Information Retrieval to Automate the Detection of Third-Party Java Library Migration at the Method Level. , 2019, , .		17
42	MigrationMiner: An Automated Detection Tool of Third-Party Java Library Migration at the Method Level. , 2019, , .		17
43	Towards Automated Microservices Extraction Using Muti-objective Evolutionary Search. Lecture Notes in Computer Science, 2019, , 58-63.	1.0	16
44	Revisiting the relationship between code smells and refactoring. , 2016, , .		15
45	Search-based detection of model level changes. Empirical Software Engineering, 2017, 22, 670-715.	3.0	15
46	Bi-level Identification of Web Service Defects. Lecture Notes in Computer Science, 2016, , 352-368.	1.0	15
47	An Exploratory Study on the Refactoring of Unit Test Files in Android Applications. , 2020, , .		14
48	Multiobjective Optimization for Software Refactoring and Evolution. Advances in Computers, 2014, 94, 103-167.	1.2	13
49	SIM: An Automated Approach to Improve Web Service Interface Modularization. , 2016, , .		13
50	Improving web service interfaces modularity using multi-objective optimization. Automated Software Engineering, 2019, 26, 275-312.	2.2	13
51	Improving the prediction of continuous integration build failures using deep learning. Automated Software Engineering, 2022, 29, 1.	2.2	13
52	On the documentation of refactoring types. Automated Software Engineering, 2022, 29, 1.	2.2	13
53	A Machine Learning-Based Approach to Detect Web Service Design Defects. , 2017, , .		12
54	How Does Library Migration Impact Software Quality and Comprehension? An Empirical Study. Lecture Notes in Computer Science, 2020, , 245-260.	1.0	12

#	Article	IF	Citations
55	Identification of Web Service Refactoring Opportunities as a Multi-objective Problem. , 2016, , .		11
56	A context-based refactoring recommendation approach using simulated annealing. , 2017, , .		11
57	Comparing Commit Messages and Source Code Metrics for the Prediction Refactoring Activities. Algorithms, 2021, 14, 289.	1.2	11
58	On the Relationship Between Developer Experience and Refactoring. , 2020, , .		11
59	A Hybrid Approach for Improving the Design Quality of Web Service Interfaces. ACM Transactions on Internet Technology, 2019, 19, 1-24.	3.0	10
60	An Empirical Study on the Impact of Refactoring on Quality Metrics in Android Applications. , 2021, , .		10
61	On the impact of Continuous Integration on refactoring practice: An exploratory study on TravisTorrent. Information and Software Technology, 2021, 138, 106618.	3.0	10
62	An exploratory study on library aging by monitoring client usage in a software ecosystem. , 2017, , .		9
63	csDetector: an open source tool for community smells detection. , 2021, , .		9
64	SATDBailiff-mining and tracking self-admitted technical debt. Science of Computer Programming, 2022, 213, 102693.	1.5	9
65	How Do Developers Refactor Code to Improve Code Reusability?. Lecture Notes in Computer Science, 2020, , 261-276.	1.0	9
66	c-JRefRec: Change-based identification of Move Method refactoring opportunities. , 2017, , .		8
67	Search based software engineering. , 2020, , .		8
68	A Hierarchical DBSCAN Method for Extracting Microservices from Monolithic Applications. , 2022, , .		8
69	Prediction of Web Services Evolution. Lecture Notes in Computer Science, 2016, , 282-297.	1.0	7
70	An Ontology-Based Approach for User Interface Adaptation. Advances in Intelligent Systems and Computing, 2017, , 199-215.	0.5	7
71	Behind the scenes: On the relationship between developer experience and refactoring. Journal of Software: Evolution and Process, 2024, 36, e2395.	1.2	7
72	On the Diffusion and Impact of Code Smells in Web Applications. Lecture Notes in Computer Science, 2020, , 67-84.	1.0	6

#	Article	IF	Citations
73	Detecting Skipped Commits in Continuous Integration Using Multi-objective Evolutionary Search. IEEE Transactions on Software Engineering, 2021, , 1-1.	4.3	6
74	Bayesian Optimized XGBoost Model for Traffic Speed Prediction Incorporating Weather Effects. , 2020, , .		6
75	On the Value of Quality of Service Attributes for Detecting Bad Design Practices. , 2017, , .		5
76	Interactive Refactoring of Web Service Interfaces Using Computational Search. IEEE Transactions on Services Computing, 2017, , 1-1.	3.2	5
77	A longitudinal exploratory study on code smells in server side web applications. Software Quality Journal, 2021, 29, 901-941.	1.4	5
78	A longitudinal study of the impact of refactoring in android applications. Information and Software Technology, 2021, 140, 106699.	3.0	5
79	On the prediction of continuous integration build failures using search-based software engineering. , 2020, , .		5
80	Refactoring for reuse: an empirical study. Innovations in Systems and Software Engineering, 2022, 18, 105-135.	1.6	5
81	Towards a Rigorous Consideration of Occupant Behaviours of Residential Households for Effective Electrical Energy Savings: An Overview. Energies, 2022, 15, 1741.	1.6	5
82	BF-detector: an automated tool for CI build failure detection. , 2021, , .		4
83	Multi-criteria Web Services Selection: Balancing the Quality of Design and Quality of Service. ACM Transactions on Internet Technology, 2022, 22, 1-31.	3.0	4
84	AndroLib: Third-Party Software Library Recommendation for Android Applications. Lecture Notes in Computer Science, 2020, , 208-225.	1.0	4
85	Increasing the Trust In Refactoring Through Visualization. , 2020, , .		4
86	Web Service API Anti-patterns Detection as a Multi-label Learning Problem. Lecture Notes in Computer Science, 2020, , 114-132.	1.0	4
87	Improving microservices extraction using evolutionary search. Information and Software Technology, 2022, 151, 106996.	3.0	4
88	Improving Web Services Design Quality Using Heuristic Search and Machine Learning. , 2017, , .		3
89	On the use of textual feature extraction techniques to support the automated detection of refactoring documentation. Innovations in Systems and Software Engineering, 0, , 1.	1.6	3
90	Search-based detection of code changes introducing performance regression. Swarm and Evolutionary Computation, 2022, 73, 101101.	4.5	3

#	Article	IF	Citations
91	Recommending peer reviewers in modern code review. , 2020, , .		2
92	An Exploratory Study on How Software Reuse is Discussed in Stack Overflow. Lecture Notes in Computer Science, 2020, , 292-303.	1.0	2
93	Toward a Smell-aware Prediction Model for CI Build Failures. , 2021, , .		2
94	Search-Based Third-Party Library Migration at the Method-Level. Lecture Notes in Computer Science, 2022, , 173-190.	1.0	2
95	On the Use of Refactoring in Security Vulnerability Fixes: An Exploratory Study on Maven Libraries. , 2022, , .		2
96	Tracking bad updates in mobile apps: a search-based approach. Empirical Software Engineering, 2022, 27, 1.	3.0	1
97	An Empirical Study on Code Smells Co-occurrences in Android Applications. , 2021, , .		0
98	On the Identification of Third-Party Library Usage Patterns for Android Applications. , 2022, , .		0