Research Statement

David Lo
School of Computing and Information Systems
Singapore Management University
Tel: (65) 6828-0599; Email: davidlo@smu.edu.sg
Updated on December 20, 2024

Background

Software and software development activities produce a huge amount of data daily. The amount of new software code written by software companies and open source projects easily goes to millions of lines of code daily. Modern software development practices often include deployment of repositories, e.g., Git, etc, which contains other forms of information aside from the code. These include information on when a piece of code is written, who is writing into what file, etc. Bug reports and bug tracking information stored in systems like Bugzilla and Jira are also widely available. These data sources covering people, processes, products, provide a rich source of information to be analyzed.

Software development itself faces many challenges. Difficulties in managing legacy systems and presence of bugs have cost billions of dollars annually. It is estimated that a substantial proportion of software cost is due to the difficulties in understanding existing/legacy systems especially during maintenance tasks, i.e. when new feature updates, bug fix, etc. are performed. US National Institute of Standards and Technology (NIST) estimated that software bugs have caused US economy to lose 59.5 billion dollars annually.

As a step forward to reduce software maintenance cost and detect bugs, machine learning and data mining techniques have been employed to mine knowledge from existing program artifacts (either from source code, execution traces, bug reports, comments, developer socio-technical network, etc). This is termed as software analytics and has been one of the new, hot topics in software engineering. The mined knowledge can be used for understanding legacy systems, reducing software maintenance cost, re-engineering legacy system, improving regression tests, aiding verification of programs, detecting bugs, etc.

Motivated by the above mentioned challenges and opportunities, application-wise, my research goal focuses on this area of software analytics; in particular, I'm interested in extending data analytics solution to transform the wealth of data available and could be collected from software and its development activities into actionable knowledge useful for software developers and other stakeholders in the software development process. Algorithm-wise, I work on improving frequent pattern mining, extending it to mine for more expressive patterns more efficiently from various data sources related to primarily, but not limited to, software engineering, and also: social network, spatio-temporal information, text data, etc.

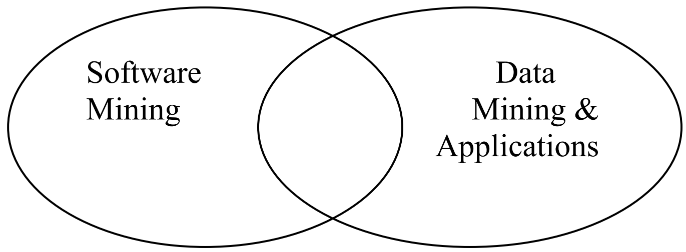

Figure 1. My Research Goals

Research Areas

Most of my work could be grouped into 5 topics: mining software specifications, bug management, code search, frequent pattern mining algorithms, and social network mining. I describe these five topics in more detail in the following paragraphs. These studies were performed together with various collaborators around the globe.

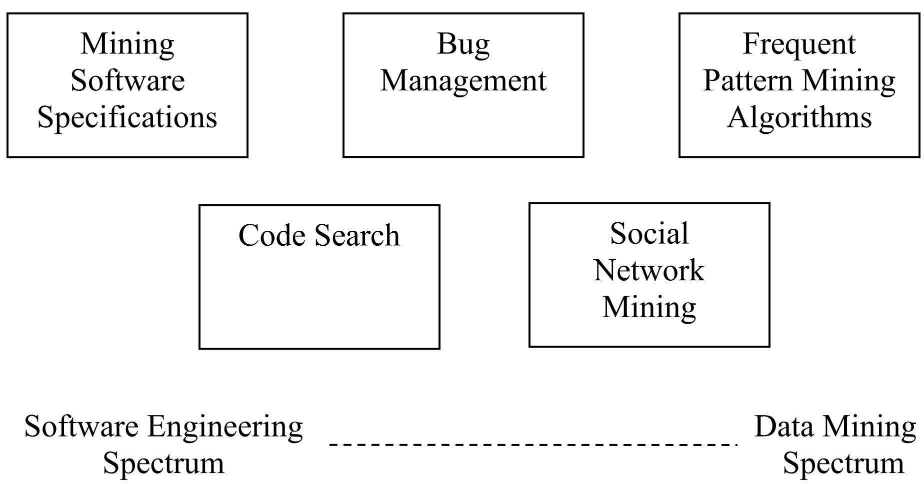

Figure 2. My Current Research Topics of Interest

Mining Software Specifications. Software specifications are often not available, incomplete, or outdated in the industry. I'm interested in reverse engineering or mining specifications from programs. I especially focus on the mining of specifications from program behaviors exhibited in systems' execution traces. In the past, we have mined specifications in various formats ranging from: finite state machines, temporal rules, frequent usage patterns, and sequence diagrams [1-7,40-41,52,65].

Bug Management. Bugs are prevalent. We are interested in managing bugs in the various phases of its lifecycle: identification/detection, reporting, localization, and fixing. I have been working on the four phases. For bug identification, we have proposed various approaches that automatically find likely bugs from programs [8,9,42,56,57,68]. For bug reporting, we have investigated the problem of duplicate bug reports and propose approaches to detect those duplicates using a combination of information retrieval and data mining approaches [10,25,26,29,71]. More recently, we have also developed a novel approach to identify invalid bug reports [61]. We have also proposed approaches to recommend the best developers to work on a bug report [46,47]. For bug localization, we have investigated various approaches that localize bugs from failure reports [11,12,30,37,38,39,43,48,58,66,69]. In addition to the above, we have also performed an empirical study on types of bugs that appear in real systems [31] and proposed an approach that can categorize bugs into types [32]. For bug fixing, we have proposed various solutions that leverage historical bug fixing data and utilize program synthesis engines and deductive verification [49,50,51]. More recently, we have also looked into fixing bugs in specialized software, e.g., smart contracts [62]. We have also investigated practitioners' perception on bug report management techniques highlighting numerous opportunities for future work in this area [63]. We have also developed techniques to manage vulnerabilities, especially considering the software supply chain [72,73]. Furthermore, we developed large language models (LLM) powered methods to find and repair vulnerabilities [74,75].

Code Search. Just like a regular search engine helps users in finding information that they want, a code search engine helps developers locate desired pieces of code in a code base. This would greatly help in performing maintenance tasks, e.g., finding a piece of code to be changed. We have proposed approaches that allow for dependency and basic textual search on a code base [13,27,44]. We are planning to extend this approach further to support more advanced queries. We have also proposed an approach that can recover similar software applications leveraging collaborative tagging [33]. Our recent work introduces advanced code search solutions that leverage the power of crowd-generated contents in StackOverflow [53-54,70] and YouTube [60], and a search-and-replace solution to perform many similar transformations across a large code base [59,67]. We have also designed an approach that can convert a piece of code to its embedding (distributed representation) that can improve several downstream code search tasks [64].

Frequent Pattern Mining. I also work on novel pattern mining algorithms, especially sequential pattern mining. Along with co-authors, I have worked on mining sequence generators [14] and repetitive sequential patterns (closed patterns [15] and generators [16]). We also work on mining rules; different from patterns, a significant rule must have sufficient confidence. We've investigated non-redundant sequential rules [17] and temporal rule mining [18,19]. We are also interested in mining discriminative patterns; we have worked on mining discriminative sequential patterns [20], and dyadic sequential patterns [21]. We have applied discriminative graph mining to the problem of bug localization [11].

Social Network Mining. Recently, I'm also interested to mine patterns from social networks. We mine for patterns from software developer networks [22]. We also mine friendship propagation rules in social networks [23]. Furthermore, we also extract antagonistic communities from social networks [24,28,34,45]. Our recent work proposes an advanced method to recommend who-to-follow in the software engineering Twitter space [55].

For the above studies, I benefited from collaborations with co-authors from Zhejiang University, National University of Singapore, Inria, University of Illinois Urbana-Champaign, University of California-Berkeley, NASA, Tel Aviv University, Chinese University of Hong Kong, University of Milano-Biccoca, Peking University, University of Copenhagen, etc.

In addition to the above, I'm interested with the following research directions:

- Large language model for software engineering
- Software engineering methodologies to develop, test, and deploy machine learning and AI solutions (aka. SE4AI and MLOps), including large language models
- Application of existing mining techniques to interesting research problems in:
    - Security and intrusion detection
    - Program comprehension
    - Verification
    - Debugging
    - Testing
    - Re-engineering
- Further improvement to the efficiency and accuracy of existing mining techniques and expressiveness of mined specifications and patterns.
- Utilization of the synergy of static and dynamic analysis in specification mining
- Investigation of new context-based automated debugging approaches
- Merging social network mining and analysis to software engineering
- Analyzing textual software engineering data
- Empirical studies in software engineering
- Big software data analytics
- Construction of more research "bridges" joining the areas of data mining, information retrieval, programming languages, and software engineering

Selected Publications and Research Outputs

[1] David Lo and Siau-Cheng Khoo. QUARK: Empirical Assessment of Automaton-based Specification Miners. In proceedings of the 13th Working Conference on Reverse Engineering (WCRE'06) . Benevento, Italy. Oct 23-27, 2006.

[2] David Lo and Siau-Cheng Khoo. SMArTIC: Towards Building an Accurate, Robust and Scalable Specification Miner. In proceedings of the 14th SIGSOFT Symposium on Foundation of Software Engineering (FSE'06). Portland, Oregon. Nov 5-11, 2006.

[3] David Lo, Siau-Cheng Khoo and Chao Liu. Efficient Mining of Iterative Patterns for Software Specification Discovery. In proceedings of the 13th SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'07). San Jose, California. Aug 12-15, 2007.

[4] David Lo, Siau-Cheng Khoo, Chao Liu. Mining temporal rules for software maintenance, Journal of Software Maintenance and Evolution: Research and Practice, vol. 20, no. 4, pp. 227–247, John Wiley & Sons, Inc., New York, NY, USA, 2008

[5] David Lo, Shahar Maoz and Siau-Cheng Khoo. Mining Modal Scenario-based Specifications from Execution Traces of Reactive Systems. In proceedings of the 22nd IEEE/SIGSOFT International Conference on Automated Software Engineering (ASE'07). Atlanta, Georgia. Nov 5-9, 2007.

[6] David Lo and Shahar Maoz. Mining Scenario-Based Triggers and Effects. In proceedings of the 23rd IEEE/SIGSOFT International Conference on Automated Software Engineering (ASE'08). L'Aquila, Italy. September 15-19, 2008.

[7] David Lo and Shahar Maoz. Scenario-based and value-based specification mining: better together, in proceedings of the 25th IEEE/ACM International Conference on Automated Software Engineering (ASE’10). Antwerp, Belgium. September 20-24, 2010.

[8] Julia L. Lawall and David Lo. An automated approach for finding variable-constant pairing bugs, in proceedings of the 25th IEEE/ACM International Conference on Automated Software Engineering (ASE’10). Antwerp, Belgium. September 20-24, 2010.

[9] David Lo, Ganesan Ramalingam, Venkatesh-Prasad Ranganath, and Kapil Vaswani. Mining Quantified Temporal Rules: Formalism, Algorithms, and Evaluation, in Science of Computer Programming (SCP), 2011.

[10] Chengnian Sun, David Lo, Xiaoyin Wang, Jing Jiang, and Siau-Cheng Khoo. A Discriminative Model Approach for Accurate Duplicate Bug Report Retrieval, in proceedings of the ACM/IEEE International Conference on Software Engineering (ICSE’10), Cape Town, South Africa

[11] Hong Cheng, David Lo, Yang Zhou, Xiaoyin Wang, and Xifeng Yan. Identifying Bug Signatures using Discriminative Graph Mining. In proceedings of the ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA’09), Chicago, IL

[12] Lucia, David Lo, Lingxiao Jiang, and Aditya Budi. Comprehensive Evaluation of Association Measures for Fault-Localization, in proceedings of the 26th IEEE International Conference on Software Maintenance (ICSM’10). Timisoara, Romania. September 12-18, 2010.

[13] Xiaoyin Wang, David Lo, Jiefeng Cheng, Lu Zhang, Hong Mei, and Jeffrey Xu Yu. Matching dependence-related queries in the system dependence graph, in

proceedings of the 25th IEEE/ACM International Conference on Automated Software Engineering (ASE'10). Antwerp, Belgium. September 20-24, 2010.

[14] David Lo, Siau-Cheng Khoo and Jinyan Li. Mining and Ranking Generators of Sequential Patterns. In proceedings of the 8th SIAM International Conference on Data Mining (SDM'08). Atlanta, USA. April 24-26, 2008.

[15] Bolin Ding, David Lo, Jiawei Han and Siau-Cheng Khoo. Efficient Mining of Closed Repetitive Gapped Subsequences from a Sequence Database. In proceedings of the 25th International Conference on Data Engineering (ICDE'09), Shanghai, China. March 29-April 4, 2009

[16] David Lo, Jinyan Li, Limsoon Wong, and Siau-Cheng Khoo. Mining Iterative Generators and Representative Rules for Software Specification Discovery. IEEE Transaction on Knowledge and Data Engineering, Feb 2011.

[17] David Lo, Siau-Cheng Khoo, and Limsoon Wong. Non-Redundant Sequential Rules - Theory and Algorithms, Information Systems, vol. 34, no. 4-5, pp. 438–453, Elsevier, 2009

[18] David Lo, Siau-Cheng Khoo and Chao Liu. Efficient Mining of Recurrent Rules from a Sequence Database. In proceedings of the 13rd International Conference on Database Systems for Advance Applications (DASFAA'08). New Delhi, India. March 19-21, 2008.

[19] David Lo, Bolin Ding, Lucia, and Jiawei Han. Bidirectional Mining of Non-Redundant Recurrent Rules from a Sequence Database. In proceedings of the 27th International Conference on Data Engineering (ICDE'11). Hannover, Germany. April 11-16, 2011.

[20] David Lo, Hong Cheng, Jiawei Han, Siau-Cheng Khoo, and Chengnian Sun. Classification of Software Behaviors for Failure Detection: A Discriminative Pattern Mining Approach. In Proceedings of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'09), Paris, France. June 28-July 1, 2009

[21] David Lo, Hong Cheng and Lucia, "Mining Closed Discriminative Dyadic Sequential Patterns", Proceedings of the 2011 International Conference on Extending Data Base Technology (EDBT 11). Uppsala, Sweden, Mar. 2011.

[22] Didi Surian, David Lo, and Ee-Peng Lim. Mining Collaboration Patterns from a Large Developer Network, in proceedings of the 17th IEEE Working Conference on Reverse Engineering (WCRE'10) (Short Paper). Boston, USA. October 13-16, 2010.

[23] Cane Wing-ki Leung, Ee-Peng Lim, David Lo and Jianshu Weng. Mining Interesting Link Formation Rules in Social Networks, in Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM 2010). Toronto, Canada. October 26-30, 2010.

[24] Kuan Zhang, David Lo, and Ee-Peng Lim. Mining Antagonistic Communities from Social Networks, in proceedings of the Pacific Asia Conference on Knowledge Discovery and Data Mining (PAKDD'10), Hyderabad

[25] Chengnian Sun, David Lo, Siau-Cheng Khoo, and Jing Jiang. Towards More Accurate Retrieval of Duplicate Bug Reports, in proceedings of the 26th IEEE/ACM International Conference on Automated Software Engineering (ASE 2011), Lawrence, USA.

[26] Tian Yuan, Chengnian Sun, and David Lo. Improved Duplicate Bug Report Identification, in Proceedings of 15th European Conference on Software Maintenance and Reengineering (CSMR 2012), ERA Track, Szeged, Hungary

[27] Shaowei Wang, David Lo, and Lingxiao Jiang. Code Search via Topic-Enriched Dependency Graph Matching, in Proceedings of the 18th IEEE Working Conference on Reverse Engineering (WCRE 2011), Limerick, Ireland

[28] David Lo, Didi Surian, Zhang Kuan, and Ee Peng Lim. Mining Direct Antagonistic Communities in Explicit Trust Network, in Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM 2011), Glasgow, United Kingdom

[29] Anh Tuan Nguyen, Tung Nguyen, Tien Nguyen, David Lo and Chengnian Sun. Duplicate Bug Report Detection with a Combination of Information Retrieval and Topic Modeling, in Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering (ASE 2012), Essen, Germany

[30] Liang Gong, David Lo, Lingxiao Jiang and Hongyu Zhang. Diversity Maximization Speedup for Fault Localization, in Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering (ASE 2012), Essen, Germany

[31] Ferdian Thung, Shaowei Wang, David Lo, and Lingxiao Jiang. An Empirical Study of Bugs in Machine Learning Systems, in Proceedings of the 23rd IEEE International Symposium on Software Reliability Engineering (ISSRE 2012), Dallas, Texas, USA

[32] Ferdian Thung, David Lo, and Lingxiao Jiang. Automatic Defect Categorization, in Proceedings of the 19th Working Conference on Reverse Engineering (WCRE 2012), Kingston, Ontario, Canada

[33] Ferdian Thung, David Lo, and Lingxiao Jiang. Detecting Similar Applications with Collaborative Tagging, in Proceedings of the 28th IEEE International Conference on Software Maintenance (ICSM 2013), Riva del Garda, Trento, Italy

[34] Zhang Kuan, David Lo, Ee-Peng Lim, and Philips K. Prasetyo. Mining Indirect Antagonistic Communities from Social Interactions, in Knowledge and Information Systems (KAIS), 2013.

[35] Ferdian Thung, David Lo, and Julia Lawall. Automatic Recommendation of API Methods from Feature Requests, in Proceedings of the 28th IEEE/ACM International Conference on Automated Software Engineering (ASE 2013), Palo Alto, California, USA

[36] Yuan Tian, David Lo, and Julia Lawall. DRONE: Predicting Priority of Reported Bugs by Multi-factor Analysis, in Proceedings of the 29th IEEE International Conference on Software Maintenance (ICSM 2013), Eindhoven, Netherland

[37] Shaowei Wang and David Lo. Version history, similar report, and structure: putting them together for improved bug localization, in Proceedings of the 22nd International Conference on Program Comprehension (ICPC 2014), Hyderabad, India

[38] Shaowei Wang, David Lo, Julia Lawall. Compositional Vector Space Models for Improved Bug Localization, in Proceedings of the 30th International Conference on Software Maintenance and Evolution (ICSME 2014), Victoria, Canada

[39] Xin Xia, David Lo, Xingen Wang, Chenyi Zhang, Xinyu Wang. Cross-language bug localization, in Proceedings of the 22nd International Conference on Program Comprehension (ICPC 2014), Hyderabad, India

[40] Tien-Duy B. Le, Xuan-Bach D. Le, David Lo, and Ivan Beschastnikh. Synergizing Specification Miners through Model Fissions and Fusions, in Proceedings of the 30th IEEE/ACM International Conference on Automated Software Engineering (ASE 2015), Lincoln, USA.

[41] Tien-Duy B. Le and David Lo. Beyond Support and Confidence: Exploring Interestingness Measures for Specification Mining, in Proceedings of the 22nd IEEE International Conference on Software Analysis, Evolution, and Reengineering (SANER 2015), Montreal, Canada.

[42] Xinli Yang, David Lo, Xin Xia, Yun Zhang and Jianling Sun. Deep Learning for Just-In-Time Defect Prediction, in Proceedings of IEEE International Conference on Software Quality, Reliability, and Security (QRS 2015), Vancouver, Canada.

[43] Tien-Duy B. Le, Richard J. Oentaryo, and David Lo. Information Retrieval and Spectrum Based Bug Localization: Better Together, in Proceedings of the 10th Joint Meeting of the European Software Engineering Conference and ACM SIGSOFT Symposium on Foundations of Software Engineering (ESEC-FSE 2015), Bergamo, Italy.

[44] Wing-Kwan Chan, Hong Cheng, and David Lo. Searching Connected API Subgraph via Text Phrases, in Proceedings of the 20th ACM International Symposium on the Foundations of Software Engineering (FSE 2012), Cary, USA.

[45] Ming Gao, Ee-Peng Lim, David Lo, Philips K. Prasetyo. On Detecting Maximal Quasi Antagonistic Communities in Signed Graphs. Data Mining and Knowledge Discovery. (DMKD), 2016.

[46] Xin Xia, David Lo, Xinyu Wang, Bo Zhou. Dual Analysis for Recommending Developers to Resolve Bugs. Journal of Software: Evolution and Process (JSEP), 2015

[47] Yuan Tian, Dinusha Wijedasa, David Lo, Claire Le Goues. Learning to Rank for Bug Report Assignee Recommendation, in Proceedings of 24th IEEE International Conference on Program Comprehension (ICPC 2016), Austin, USA.

[48] Tien-Duy B. Le, David Lo, Claire Le Goues, Lars Grunske. A Learning-to-Rank Based Fault Localization Approach using Likely Invariants, in Proceedings of the 25th International Symposium on Software Testing and Analysis (ISSTA 2016), Saarbrucken, Germany.

[49] Xuan-Bach D. Le, David Lo, Claire Le Goues. History Driven Program Repair, in Proceedings of the 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER 2016), Osaka, Japan.

[50] Xuan-Bach D. Le, David Lo and Claire Le Goues. Empirical Study on Synthesis Engines for Semantics-based Program Repair, in Proceedings of the 32nd IEEE International Conference on Software Maintenance and Evolution (ICSME 2016), Raleigh, USA.

[51] Xuan-Bach D. Le, Quang Loc Le, David Lo and Claire Le Goues. Enhancing Automated Program Repair with Deductive Verification, in Proceedings of the 32nd IEEE International Conference on Software Maintenance and Evolution (ICSME 2016), Raleigh, USA.

[52] Tien-Duy B. Le, and David Lo. Deep Specification Mining, in Proceedings of the 27th ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2018), Amsterdam, Netherlands.

[53] Qiao Huang, Xin Xia, Zhenchang Xing, David Lo, and Xinyu Wang. API Method Recommendation without Worrying about the Task-API Knowledge Gap, in Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering (ASE 2018), Montpellier, France.

[54] Raphael Sirres, Tegawendé F. Bissyandé, Dongsun Kim, David Lo, Jacques Klein, Kisub Kim, and Yves Le Traon. Augmenting and Structuring User Queries to Support Efficient Free-Form Code Search. Empirical Software Engineering 23(5): 2622-2654 (2018)

[55] Abhishek Sharma, Yuan Tian, Agus Sulistya, Dinusha Wijedasa, and David Lo. Recommending Who to Follow in the Software Engineering Twitter Space. ACM Transactions of Software Engineering Methodology 27(4): 16:1-16:33 (2018).
[56] Shu-Ting Shi, Ming Li, David Lo, Ferdian Thung, Xuan Huo. Automatic Code Review by Learning the Revision of Source Code, in Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI 2019), Hawaii, USA.
[57] Thong Hoang, Julia Lawall, Richard Jayadi Oentaryo, Yuan Tian, David Lo. PatchNet: A Tool for Deep Patch Classification, in Proceedings of the 41st International Conference on Software Engineering (ICSE 2019), Montreal, Canada.
[58] Thong Hoang, Richard Jayadi Oentaryo, Tien-Duy B. Le, David Lo. Network-Clustered Multi-Modal Bug Localization. IEEE Transactions on Software Engineering 45(10): 1002-1023 (2019).
[59] Hong Jin Kang, Ferdian Thung, Julia Lawall, Gilles Muller, Lingxiao Jiang, David Lo: Semantic Patches for Java Program Transformation. 33rd European Conference on Object-Oriented Programming (ECOOP 2019), London, UK.
[60] Lingfeng Bao, Zhenchang Xing, Xin Xia, David Lo, Minghui Wu, Xiaohu Yang: psc2code: Denoising Code Extraction from Programming Screencasts. ACM Transactions Software Engineering Methodology 29(3): 21:1-21:38 (2020)
[61] Yuanrui Fan, Xin Xia, David Lo, Ahmed E. Hassan: Chaff from the Wheat: Characterizing and Determining Valid Bug Reports. IEEE Transactions on Software Engineering 46(5): 495-525 (2020)
[62] Xiao Liang Yu, Omar I. Al-Bataineh, David Lo, Abhik Roychoudhury: Smart Contract Repair. ACM Transactions Software Engineering Methodology 29(4): 27:1-27:32 (2020)
[63] Weiqin Zou, David Lo, Zhenyu Chen, Xin Xia, Yang Feng, Baowen Xu: How Practitioners Perceive Automated Bug Report Management Techniques. IEEE Transactions on Software Engineering 46(8): 836-862 (2020)
[64] Thong Hoang, Hong Jin Kang, David Lo, Julia Lawall: CC2Vec: Distributed Representations of Code Changes. Proceedings of the 42nd International Conference on Software Engineering (ICSE 2020), Seoul, South Korea.
[65] Hong Jin Kang, David Lo: Adversarial Specification Mining. ACM Transactions on Software Engineering Methodology 30(2): 16:1-16:40 (2021)
[66] Xuan Huo, Ferdian Thung, Ming Li, David Lo, Shu-Ting Shi: Deep Transfer Bug Localization. IEEE Transactions on Software Engineering 47(7): 1368-1380 (2021)
[67] Stefanus A. Haryono, Ferdian Thung, David Lo, Julia Lawall, Lingxiao Jiang: Characterization and Automatic Updates of Deprecated Machine-Learning API Usages, in Proceedings of the 37nd IEEE International Conference on Software Maintenance and Evolution (ICSME 2021), Luxembourg.
[68] Hong Jin Kang, David Lo: Active Learning of Discriminative Subgraph Patterns for API Misuse Detection. IEEE Transactions on Software Engineering 48(8): 2761-2783 (2022)
[69] Darryl Jarman, Jeffrey Berry, Riley Smith, Ferdian Thung, David Lo: Legion: Massively Composing Rankers for Improved Bug Localization at Adobe. IEEE Transactions on Software Engineering 48(8): 3010-3024 (2022)
[70] Bowen Xu, Thong Hoang, Abhishek Sharma, Chengran Yang, Xin Xia, David Lo: Post2Vec: Learning Distributed Representations of Stack Overflow Posts. IEEE Transactions on Software Engineering 48(9): 3423-3441 (2022)
[71] Ting Zhang, DongGyun Han, Venkatesh Vinayakarao, Ivana Clairine Irsan, Bowen Xu, Ferdian Thung, David Lo, Lingxiao Jiang: Duplicate Bug Report

Detection: How Far Are We? ACM Transactions on Software Engineering Methodologies 32(4): 97:1-97:32 (2023)

[72] Truong Giang Nguyen, Thanh Le-Cong, Hong Jin Kang, Ratnadira Widyasari, Chengran Yang, Zhipeng Zhao, Bowen Xu, Jiayuan Zhou, Xin Xia, Ahmed E. Hassan, Xuan-Bach Dinh Le, David Lo: Multi-Granularity Detector for Vulnerability Fixes. IEEE Transactions on Software Engineering 49(8): 4035-4057 (2023)

[73] Yunbo Lyu, Thanh Le-Cong, Hong Jin Kang, Ratnadira Widyasari, Zhipeng Zhao, Xuan-Bach Dinh Le, Ming Li, David Lo: CHRONOS: Time-Aware Zero-Shot Identification of Libraries from Vulnerability Reports. Proceedings of the 45th International Conference on Software Engineering (ICSE 2023), Melbourne, Australia.

[74] Xin Zhou, Kisub Kim, Bowen Xu, DongGyun Han, David Lo: Out of Sight, Out of Mind: Better Automatic Vulnerability Repair by Broadening Input Ranges and Sources. Proceedings of the 46th International Conference on Software Engineering (ICSE 2024), Lisbon, Portugal.

[75] Xin Zhou, Ting Zhang, David Lo: Large Language Model for Vulnerability Detection: Emerging Results and Future Directions. Proceedings of the 46th International Conference on Software Engineering (ICSE 2024), Lisbon, Portugal.