Guest Editorial: A Retrospective of Special Sections on Software Testing and Program Analysis

I. Introduction

SOFTWARE plays an integral part in our lives today because of its near-ubiquitous influence on our increasingly technological society. Taking appropriate steps to improve software quality is, therefore, of paramount importance. This has fueled the demand for techniques in software testing and program analysis.

To meet the demand, there has been a surge in the study and development of techniques such as test case generation, static and dynamic program analysis, and test case prioritization. However, these techniques are still suffering from immaturity and impracticality. Test case generation can hardly be automated; efficiency and state-explosion are still the biggest concerns for most program analysis; and the effectiveness of test case prioritization is still based on small empirical comparisons. Furthermore, the vast majority of empirical studies available on the subject are lab based. Researchers typically evaluate their techniques based on simplified assumptions and selected subject programs that do not necessarily reflect the complexity of large-scale, industrial software. These shortcomings have prevented contemporary software testing and program analysis techniques from yielding truthful benefits to the software in the real world.

In this context, the Special Sections on Software Testing and Program Analysis solicited original work that would provide novel techniques and/or comprehensive empirical validation to show the applicability to the industry. Articles were reviewed and selected based on their innovation, technical correctness, presentation, and practical relevance.

II. Great Success and Small Challenge

We attracted 73 submissions covering diverse hot topics in the area. Hence, it was a great success among similar special sections in the IEEE Transactions on Reliability. Each manuscript was carefully reviewed by three experts in the field. After the rigorous evaluation process by 219 hard-working reviewers, possibly with recommendations for major and minor revisions, 22 articles were accepted. In some cases, the articles were authored by internationally renowned researchers, while in some other cases, they were written by young contributors for the first time in their career. Five of the authors had two submissions accepted for the Special Sections.

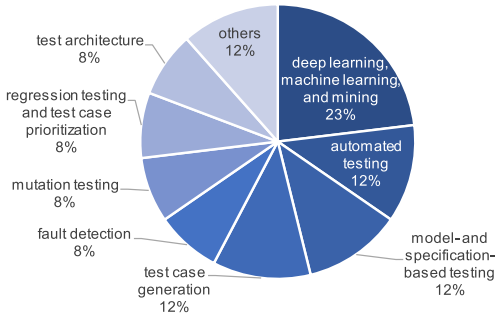

Fig. 1. Distribution of articles by testing and analysis techniques.

Because of the bulk volume, the accepted articles were published in four separate issues of the IEEE Transactions on Reliability in June 2018 [items 1)–4) in the Appendix], September 2018 [items 5)–14) in the Appendix], December 2018 [items 15)–17) in the Appendix], and January 2019 [items 18)–22) in the Appendix].

It was unfortunate that the Guest Editors could not have the foresight to determine the exact number of articles to be accepted. Some of the pending articles were rejected or withdrawn after January 2019. Thus, we missed our chance to publish a concluding editorial in the last of the four issues.

We considered that it was unfair to the diligent authors and reviewers that the Special Sections would end without any publicity and, more importantly, without any word of thanks to the awesome contributors and referees. After several rounds of discussions with the Editor-in-Chief, we decided to find an issue with a reasonable number of regular articles on software testing and program analysis, group these articles in an extended Special Section, and publish a retrospective Guest Editorial. This June 2021 issue indeed gives us the outstanding opportunity with four new articles [items 23)–26) in the Appendix] on the topic.

We declare that, during our attempt to find the present issue, we have not jeopardized the publication schedule of these four articles or any other article in the pipeline.

III. Analysis of Contributions

The 26 accepted articles represent a comprehensive cross section of testing and analysis techniques, as summarized in Fig. 1. The techniques include the following:

1) deep learning, machine learning, and mining [item 4) in the Appendix], [item 6) in the Appendix], [item 7) in the

Digital Object Identifier 10.1109/TR.2021.3083375

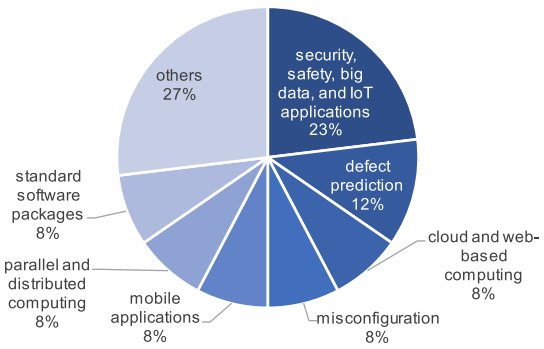

Fig. 2. Distribution of articles by applications.

Appendix], [item 11) in the Appendix], [item 18) in the Appendix], [item 26) in the Appendix];
2) automated testing [item 5) in the Appendix], [item 20) in the Appendix], [item 21) in the Appendix];
3) model- and specification-based testing and analysis [item 2) in the Appendix], [item 3) in the Appendix], [item 17) in the Appendix];
4) test case generation [item 8) in the Appendix], [item 13) in the Appendix], [item 22) in the Appendix];
5) fault detection [item 9) in the Appendix], [item 12) in the Appendix];
6) mutation testing and analysis [item 16) in the Appendix], [item 24) in the Appendix];
7) regression testing and test case prioritization [item 14) in the Appendix], [item 23) in the Appendix];
8) test architecture and design [item 1) in the Appendix], [item 19) in the Appendix];
9) fault injection [item 15) in the Appendix];
10) faulty location analysis [item 10) in the Appendix];
11) workflow analysis [item 25) in the Appendix].

The 26 accepted articles also represent a wide variety of the applications of the techniques to real-world projects, as summarized in Fig. 2. The application areas include the following:

1) security, safety, big data, and IoT applications [item 2) in the Appendix], [item 3) in the Appendix], [item 5) in the Appendix], [item 7) in the Appendix], [item 18) in the Appendix], [item 19) in the Appendix];
2) defect prediction [item 4) in the Appendix], [item 14) in the Appendix], [item 26) in the Appendix];
3) cloud and web-based computing [item 6) in the Appendix], [item 10) in the Appendix];
4) misconfiguration [item 11) in the Appendix], [item 15) in the Appendix];
5) mobile applications [item 20) in the Appendix], [item 21) in the Appendix];
6) parallel and distributed computing [item 1) in the Appendix], [item 9) in the Appendix];
7) standard software packages [item 8) in the Appendix], [item 23) in the Appendix];
8) cryptographic hash functions [item 13) in the Appendix];
9) nuclear software systems [item 16) in the Appendix];
10) protocol bugs [item 12) in the Appendix];
11) railway signaling [item 24) in the Appendix];
12) simulation [item 22) in the Appendix];
13) unmanned aerial vehicles [item 17) in the Appendix];
14) workflow management systems [item 25) in the Appendix].

IV. Concluding Remarks

We are pleased to note the immense interest of potential authors of the IEEE TRANSACTIONS ON RELIABILITY on software testing and program analysis. In particular, we are grateful to the 109 authors and the 219 reviewers. The overwhelming success of the Special Sections would not be possible without their excellent contributions and meticulous evaluations. We look forward to another special section on a related topic.

T. H. Tse, *Guest Editor*
Department of Computer Science
The University of Hong Kong
Pokfulam, Hong Kong
thtse@cs.hku.hk

Yves Le Traon, *Guest Editor*
Interdisciplinary Centre for Security, Reliability and Trust
University of Luxembourg
4365 Esch-sur-Alzette, Luxembourg
yves.letraon@uni.lu

Zhenyu Chen, *Guest Editor*
State Key Laboratory for Novel Software Technology
Nanjing University
Nanjing 210023, China
zychen@nju.edu.cn

Appendix

List of Articles in Special Sections on Software Testing and Program Analysis

1. R. M. Hierons, M. G. Merayo, and M. Núñez, "Bounded reordering in the distributed test architecture," *IEEE Trans. Rel.*, vol. 67, no. 2, pp. 522–537, Jun. 2018.
2. R. Blanco, J. G. Enríquez, F. J. Domínguez-Mayo, M. J. Escalona, and J. Tuya, "Early integration testing for entity reconciliation in the context of heterogeneous data sources," *IEEE Trans. Rel.*, vol. 67, no. 2, pp. 538–556, Jun. 2018.
3. M. Bures, K. Frajtak, and B. S. Ahmed, "Tapir: Automation support of exploratory testing using model reconstruction of the system under test," *IEEE Trans. Rel.*, vol. 67, no. 2, pp. 557–580, Jun. 2018.
4. F. Wu *et al.*, "Cross-project and within-project semisupervised software defect prediction: A unified approach," *IEEE Trans. Rel.*, vol. 67, no. 2, pp. 581–597, Jun. 2018.

5. J. Morán, A. Bertolino, C. Riva, and J. Tuya, "Automatic testing of design faults in MapReduce applications," *IEEE Trans. Rel.*, vol. 67, no. 3, pp. 717–732, Sep. 2018.
6. D. Appelt, C. D. Nguyen, A. Panichella, and L. C. Briand, "A machine-learning-driven evolutionary approach for testing web application firewalls," *IEEE Trans. Rel.*, vol. 67, no. 3, pp. 733–757, Sep. 2018.
7. C. Ieva, A. Gotlieb, S. Kaci, and N. Lazaar, "Discovering program topoi via hierarchical agglomerative clustering," *IEEE Trans. Rel.*, vol. 67, no. 3, pp. 758–770, Sep. 2018.
8. C. Oliveira, A. Aleti, L. Grunske, and K. Smith-Miles, "Mapping the effectiveness of automated test suite generation techniques," *IEEE Trans. Rel.*, vol. 67, no. 3, pp. 771–785, Sep. 2018.
9. J. Yang, B. Jiang, and W. K. Chan, "HistLock+: Precise memory access maintenance without lockset comparison for complete hybrid data race detection," *IEEE Trans. Rel.*, vol. 67, no. 3, pp. 786–801, Sep. 2018.
10. G. Qi, W.-T. Tsai, C. J. Colbourn, J. Luo, and Z. Zhu, "Test-algebra-based fault location analysis for the concurrent combinatorial testing," *IEEE Trans. Rel.*, vol. 67, no. 3, pp. 802–831, Sep. 2018.
11. X. Liao, S. Zhou, S. Li, Z. Jia, X. Liu, and H. He, "Do you really know how to configure your software? Configuration constraints in source code may help," *IEEE Trans. Rel.*, vol. 67, no. 3, pp. 832–846, Sep. 2018.
12. D. Cotroneo, L. De Simone, and R. Natella, "Run-time detection of protocol bugs in storage I/O device drivers," *IEEE Trans. Rel.*, vol. 67, no. 3, pp. 847–869, Sep. 2018.
13. N. Mouha, M. S. Raunak, D. R. Kuhn, and R. Kacker, "Finding bugs in cryptographic hash function implementations," *IEEE Trans. Rel.*, vol. 67, no. 3, pp. 870–884, 2018.
14. X. Yang and W. Wen, "Ridge and lasso regression models for cross-version defect prediction," *IEEE Trans. Rel.*, vol. 67, no. 3, pp. 885–896, Sep. 2018.
15. S. Li *et al.*, "ConfVD: System reactions analysis and evaluation through misconfiguration injection," *IEEE Trans. Rel.*, vol. 67, no. 4, pp. 1393–1405, Dec. 2018.
16. P. Delgado-Pérez, I. Habli, S. Gregory, R. Alexander, J. Clark, and I. Medina-Bulo, "Evaluation of mutation testing in a nuclear industry case study," *IEEE Trans. Rel.*, vol. 67, no. 4, pp. 1406–1419, Dec. 2018.
17. L. Chaves, I. V. Bessa, H. Ismail, A. B. S. Frutuoso, L. Cordeiro, and E. B. L. Filho, "DSVerifier-aided verification applied to attitude control software in unmanned aerial vehicles," *IEEE Trans. Rel.*, vol. 67, no. 4, pp. 1420–1441, Dec. 2018.
18. H. Jiang, X. Li, Z. Ren, J. Xuan, and Z. Jin, "Toward better summarizing bug reports with crowdsourcing elicited attributes," *IEEE Trans. Rel.*, vol. 68, no. 1, pp. 2–22, Mar. 2019.
19. S. Siboni *et al.*, "Security testbed for internet-of-things devices," *IEEE Trans. Rel.*, vol. 68, no. 1, pp. 23–44, Mar. 2019.
20. P. Kong, L. Li, J. Gao, K. Liu, T. F. Bissyandé, and J. Klein, "Automated testing of Android apps: A systematic literature review," *IEEE Trans. Rel.*, vol. 68, no. 1, pp. 45–66, Mar. 2019.
21. R. Coppola, M. Morisio, and M. Torchiano, "Mobile GUI testing fragility: A study on open-source Android applications," *IEEE Trans. Rel.*, vol. 68, no. 1, pp. 67–90, Mar. 2019.
22. M. Olsen and M. Raunak, "Increasing validity of simulation models through metamorphic testing," *IEEE Trans. Rel.*, vol. 68, no. 1, pp. 91–108, Mar. 2019.
23. Z. Q. Zhou, C. Liu, T. Y. Chen, T. H. Tse, and W. Susilo, "Beating random test case prioritization," *IEEE Trans. Rel.*, vol. 70, no. 2, 2021, doi: 10.1109/TR.2020.2979815.
24. L. Laibinis, A. Iliasov, and A. Romanovsky, "Mutation testing for rule-based verification of railway signaling data," *IEEE Trans. Rel.*, vol. 70, no. 2, 2021, doi: 10.1109/TR.2020.3047462.
25. P.-S. Huang, F. Fahmi, and F.-J. Wang, "Improving the detection of artifact anomalies in a workflow analysis," *IEEE Trans. Rel.*, vol. 70, no. 2, 2021, doi: 10.1109/TR.2020.3048612.
26. H. Wang, W. Zhuang, and X. Zhang, "Software defect prediction based on gated hierarchical LSTMs," *IEEE Trans. Rel.*, vol. 70, no. 2, 2021, doi: 10.1109/TR.2020.3047396.