

## APPENDIX A

## SELECTED PRIMARY STUDIES

- [Anu 19] H. Anu, J. Chen, W. Shi, J. Hou, B. Liang, and B. Qin. "An Approach to Recommendation of Verbosity Log Levels Based on Logging Intention". In: *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME '19)*, pp. 125–134, IEEE, Sep. 2019.
- [Chen 17a] B. Chen and Z. M. Jiang. "Characterizing and Detecting Anti-Patterns in the Logging Code". In: *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE '17)*, pp. 71–81, IEEE, May 2017.
- [Chen 17b] B. Chen and Z. M. J. Jiang. "Characterizing Logging Practices in Java-Based Open Source Software Projects – A Replication Study in Apache Software Foundation". *Empirical Software Engineering*, Vol. 22, No. 1, pp. 330–374, Feb. 2017.
- [Chen 19] B. Chen and Z. M. Jiang. "Extracting and Studying the Logging-Code-Issue-Introducing Changes in Java-Based Large-Scale Open Source Software Systems". *Empirical Software Engineering*, Vol. 24, No. 4, pp. 2285–2322, Aug. 2019.
- [Chen 20] B. Chen and Z. M. J. Jiang. "Studying the Use of Java Logging Utilities in the Wild". In: *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering (ICSE '20)*, pp. 397–408, Oct. 2020.
- [Chow 18] S. Chowdhury, S. D. Nardo, A. Hindle, and Z. M. Jiang. "An Exploratory Study on Assessing the Energy Impact of Logging on Android Applications". *Empirical Software Engineering*, Vol. 23, No. 3, pp. 1422–1456, June 2018.
- [Cinq 09] M. Cinque, D. Cotroneo, and A. Pecchia. "A Logging Approach for Effective Dependability Evaluation of Complex Systems". In: *2009 Second International Conference on Dependability (DEPEND '09)*, pp. 105–110, IEEE, June 2009.
- [Cinq 10] M. Cinque, D. Cotroneo, R. Natella, and A. Pecchia. "Assessing and Improving the Effectiveness of Logs for the Analysis of Software Faults". In: *2010 IEEE/IFIP International Conference on Dependable Systems & Networks (DSN '10)*, pp. 457–466, IEEE, June 2010.
- [Cinq 12] M. Cinque, D. Cotroneo, and A. Pecchia. "Event Logs for the Analysis of Software Failures: A Rule-Based Approach". *IEEE Transactions on Software Engineering*, Vol. 39, No. 6, pp. 806–821, Oct. 2012.
- [Cinq 20] M. Cinque, R. D. Corte, and A. Pecchia. "An empirical analysis of error propagation in critical software systems". *Empirical Software Engineering*, Vol. 25, No. 4, pp. 2450–2484, March 2020.
- [Ding 15] R. Ding, H. Zhou, J.-G. Lou, H. Zhang, Q. Lin, Q. Fu, D. Zhang, and T. Xie. "Log2: A Cost-Aware Logging Mechanism for Performance Diagnosis". In: *2015 USENIX Annual Technical Conference (USENIX ATC '15)*, pp. 139–150, USENIX Association, Santa Clara, CA, July 2015.
- [Fu 14] Q. Fu, J. Zhu, W. Hu, J.-G. Lou, R. Ding, Q. Lin, D. Zhang, and T. Xie. "Where Do Developers Log? An Empirical Study on Logging Practices in Industry". In: *Companion Proceedings of the 36th International Conference on Software Engineering (ICSE-Companion '14)*, pp. 24–33, ACM, New York, NY, USA, May 2014.
- [Ghol 20] S. Gholamian and P. A. S. Ward. "Logging Statements' Prediction Based on Source Code Clones". In: *Proceedings of the 35th Annual ACM Symposium on Applied Computing (SAC '20)*, pp. 82–91, ACM, March 2020.
- [Hass 18] M. Hassani, W. Shang, E. Shihab, and N. Tsantalis. "Studying and Detecting Log-Related Issues". *Empirical Software Engineering*, Vol. 23, No. 6, pp. 3248–3280, March 2018.
- [He 18] P. He, Z. Chen, S. He, and M. R. Lyu. "Characterizing the Natural Language Descriptions in Software Logging Statements". In: *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering (ASE '18)*, pp. 178–189, ACM, Sep. 2018.
- [Jia 18] Z. Jia, S. Li, X. Liu, X. Liao, and Y. Liu. "SMARTLOG: Place Error Log Statement by Deep Understanding of Log Intention". In: *2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER '18)*, pp. 61–71, IEEE, March 2018.
- [Kabi 16a] S. Kabinna, C.-P. Bezemer, W. Shang, and A. E. Hassan. "Logging Library Migrations: A Case Study for the Apache Software Foundation Projects". In: *Proceedings of the 13th International Workshop on Mining Software Repositories (MSR '16)*, pp. 154–164, ACM, May 2016.
- [Kabi 16b] S. Kabinna, W. Shang, C.-P. Bezemer, and A. E. Hassan. "Examining the Stability of Logging Statements". In: *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER '16)*, pp. 326–337, IEEE, March 2016.
- [Kim 19] T. Kim, S. Kim, S. Park, and Y. Park. "Automatic Recommendation to Appropriate Log Levels". *Software: Practice and Experience*, Vol. 50, No. 3, pp. 189–209, Nov. 2019.
- [Kubo 20] T. Kubota, N. Aota, and K. Kono. "Logging Inter-Thread Data Dependencies in Linux Kernel". *IEEE Transactions on Information and Systems*, Vol. E103.D, No. 7, pp. 1633–1646, July 2020.
- [Lal 15] S. Lal, N. Sardana, and A. Sureka. "Two Level Empirical Study of Logging Statements in Open Source Java Projects". *International Journal of Open Source Software and Processes (IJOSSP '15)*, Vol. 6, No. 1, pp. 49–73, Jan. 2015.
- [Lal 16a] S. Lal, N. Sardana, and A. Sureka. "Improving Logging Prediction on Imbalanced Datasets: A Case Study on Open Source Java Projects". *International Journal of Open Source Software and Processes (IJOSSP '16)*, Vol. 7, No. 2, pp. 43–71, Apr. 2016.
- [Lal 16b] S. Lal, N. Sardana, and A. Sureka. "LogOptPlus: Learning to Optimize Logging in Catch and If Programming Constructs". In: *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC '16)*, pp. 215–220, IEEE, June 2016.
- [Lal 16c] S. Lal and A. Sureka. "LogOpt: Static Feature Extraction from Source Code for Automated Catch Block Logging Prediction". In: *Proceedings of the 9th India Software Engineering Conference (ISEC '16)*, pp. 151–155, ACM, Feb. 2016.
- [Lal 17] S. Lal, N. Sardana, and A. Sureka. "ECLogger: Cross-Project Catch-Block Logging Prediction Using Ensemble of Classifiers". *e-Informatica Software Engineering Journal*, Vol. 11, No. 1, pp. 7–38, 2017.
- [Lal 19] S. Lal, N. Sardana, and A. Sureka. "Three-Level Learning for Improving Cross-Project Logging Prediction for If-Blocks". *Journal of King Saud University - Computer and Information Sciences*, Vol. 31, No. 4, pp. 481–496, Oct. 2019.
- [Li 17a] H. Li, W. Shang, and A. E. Hassan. "Which Log Level Should Developers Choose for a New Logging Statement?". *Empirical Software Engineering*, Vol. 22, No. 4, pp. 1684–1716, Aug. 2017.
- [Li 17b] H. Li, W. Shang, Y. Zou, and A. E. Hassan. "Towards Just-in-Time Suggestions for Log Changes". *Empirical Software Engineering*, Vol. 22, No. 4, pp. 1831–1865, Aug. 2017.
- [Li 18a] H. Li, T.-H. P. Chen, W. Shang, and A. E. Hassan. "Studying Software Logging Using Topic Models". *Empirical Software Engineering*, Vol. 23, No. 5, pp. 2655–2694, Oct. 2018.
- [Li 18b] S. Li, X. Niu, Z. Jia, J. Wang, H. He, and T. Wang. "Log-Tracker: Learning Log Revision Behaviors Proactively from Software Evolution History". In: *Proceedings of the 26th Conference on Program Comprehension (ICPC '18)*, pp. 178–188, ACM, May 2018.
- [Li 19] Z. Li, T.-H. Chen, J. Yang, and W. Shang. "DLFinder: Characterizing and Detecting Duplicate Logging Code Smells". In: *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE '19)*, pp. 152–163, IEEE, May 2019.
- [Li 20a] H. Li, W. Shang, B. Adams, M. Sayagh, and A. E. Hassan. "A Qualitative Study of the Benefits and Costs of Logging From Developers' Perspectives". *IEEE Transactions on Software Engineering*, pp. 1–1, Jan. 2020.
- [Li 20b] Z. Li, T.-H. Chen, and W. Shang. "Where Shall We Log? Studying and Suggesting Logging Locations in Code Blocks". In: *Proceedings of the 35th International Conference on Automated Software Engineering (ASE '20)*, pp. 361–372, IEEE, Sep. 2020.
- [Liu 19] Z. Liu, X. Xia, D. Lo, Z. Xing, A. E. Hassan, and S. Li. "Which Variables Should I Log?". *IEEE Transactions on Software Engineering*, Sep. 2019.
- [Liu 20] X. Liu, T. Jia, Y. Li, H. Yu, Y. Yue, and C. Hou. "Automatically Generating Descriptive Texts in Logging Statements:

- How Far Are We?”. In: *Asian Symposium on Programming Languages and Systems*, pp. 251–269, Springer, Nov. 2020.
- [Luo 18] L. Luo, S. Nath, L. R. Sivalingam, M. Musuvathi, and L. Ceze. “Troubleshooting Transiently-Recurring Errors in Production Systems with Blame-Proportional Logging”. In: *2018 USENIX Annual Technical Conference (USENIX ATC '18)*, pp. 321–334, USENIX Association, July 2018.
- [Marr 18] M. Marron. “Log++ Logging for a Cloud-Native World”. In: *Proceedings of the 14th ACM SIGPLAN International Symposium on Dynamic Languages (DLS '18)*, pp. 25–36, ACM, Oct. 2018.
- [Mizo 19] T. Mizouchi, K. Shimari, T. Ishio, and K. Inoue. “PADLA: A Dynamic Log Level Adapter Using Online Phase Detection”. In: *2019 IEEE/ACM 27th International Conference on Program Comprehension (ICPC '19)*, pp. 135–138, IEEE, May 2019.
- [Pecc 12] A. Pecchia and S. Russo. “Detection of Software Failures through Event Logs: An Experimental Study”. In: *2012 IEEE 23rd International Symposium on Software Reliability Engineering (ISSRE '12)*, pp. 31–40, IEEE, Nov. 2012.
- [Pecc 15] A. Pecchia, M. Cinque, G. Carrozza, and D. Cotroneo. “Industry Practices and Event Logging: Assessment of a Critical Software Development Process”. In: *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering (ICSE '15)*, pp. 169–178, IEEE, May 2015.
- [Rong 18] G. Rong, S. Gu, H. Zhang, D. Shao, and WanggenLiu. “How Is Logging Practice Implemented in Open Source Software Projects? A Preliminary Exploration”. In: *2018 25th Australasian Software Engineering Conference (ASWEC '18)*, pp. 171–180, IEEE, Nov. 2018.
- [Rong 20] G. Rong, Y. Xu, S. Gu, H. Zhang, and D. Shao. “Can You Capture Information As You Intend To? A Case Study on Logging Practice in Industry”. In: *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME '20)*, pp. 12–22, Sep. 2020.
- [Sain 16] S. Saini, N. Sardana, and S. Lal. “Logger4u: Predicting Debugging Statements in the Source Code”. In: *2016 Ninth International Conference on Contemporary Computing (IC3 '16)*, pp. 1–7, IEEE, Aug. 2016.
- [Salf 04] F. Salfner, S. Tschirpke, and M. Malek. “Comprehensive Logfiles for Autonomic Systems”. In: *Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS '04)*, pp. 211–219, IEEE, Apr. 2004.
- [Shan 14] W. Shang, Z. M. Jiang, B. Adams, A. E. Hassan, M. W. Godfrey, M. Nasser, and P. Flora. “An Exploratory Study of the Evolution of Communicated Information About the Execution of Large Software Systems”. *Journal of Software: Evolution and Process*, Vol. 26, No. 1, pp. 3–26, Feb. 2014.
- [Tova 13] D. Tovarňák, A. Vašeková, S. Novák, and T. Pitner. “Structured and Interoperable Logging for the Cloud Computing Era: The Pitfalls and Benefits”. In: *2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing (UCC '13)*, pp. 91–98, IEEE, Dec. 2013.
- [Yao 18] K. Yao, G. B. de Pádua, W. Shang, S. Sporea, A. Toma, and S. Sajedi. “Log4Perf: Suggesting Logging Locations for Web-Based Systems’ Performance Monitoring”. In: *Proceedings of the 2018 ACM/SPEC International Conference on Performance Engineering (ICPE '18)*, pp. 127–138, ACM, March 2018.
- [Yuan 12a] D. Yuan, S. Park, P. Huang, Y. Liu, M. M.-J. Lee, X. Tang, Y. Zhou, and S. Savage. “Be Conservative: Enhancing Failure Diagnosis with Proactive Logging”. In: *Presented as part of the 10th USENIX Symposium on Operating Systems Design and Implementation (OSDI '12)*, pp. 293–306, Oct. 2012.
- [Yuan 12b] D. Yuan, S. Park, and Y. Zhou. “Characterizing Logging Practices in Open-Source Software”. In: *2012 34th International Conference on Software Engineering (ICSE '12)*, pp. 102–112, IEEE, June 2012.
- [Yuan 12c] D. Yuan, J. Zheng, S. Park, Y. Zhou, and S. Savage. “Improving Software Diagnosability via Log Enhancement”. *ACM Transactions on Computer Systems*, Vol. 30, No. 1, p. 4, Feb. 2012.
- [Zeng 19] Y. Zeng, J. Chen, W. Shang, and T.-H. P. Chen. “Studying the Characteristics of Logging Practices in Mobile Apps: A Case Study on F-Droid”. *Empirical Software Engineering*, Vol. 24, No. 6, pp. 3394–3434, Feb. 2019.
- [Zhao 17] X. Zhao, K. Rodrigues, Y. Luo, M. Stumm, D. Yuan, and Y. Zhou. “Log20: Fully Automated Optimal Placement of Log Printing Statements Under Specified Overhead Threshold”. In: *Proceedings of the 26th Symposium on Operating Systems Principles (SOSP '17)*, pp. 565–581, ACM, Oct. 2017.
- [Zhi 19] C. Zhi, J. Yin, S. Deng, M. Ye, M. Fu, and T. Xie. “An Exploratory Study of Logging Configuration Practice in Java”. In: *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME '19)*, pp. 459–469, IEEE, Sep. 2019.
- [Zhi 20] C. Zhi, J. Yin, J. Han, and S. Deng. “A Preliminary Study on Sensitive Information Exposure Through Logging”. In: *2020 27th Asia-Pacific Software Engineering Conference (APSEC '20)*, pp. 470–474, IEEE, Dec. 2020.
- [Zhou 20] R. Zhou, M. Hamdaqa, H. Cai, and A. Hamou-Lhadj. “MobiLogLeak: A Preliminary Study on Data Leakage Caused by Poor Logging Practices”. In: *2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER '20)*, pp. 577–581, Feb. 2020.
- [Zhu 15] J. Zhu, P. He, Q. Fu, H. Zhang, M. R. Lyu, and D. Zhang. “Learning to Log: Helping Developers Make Informed Logging Decisions”. In: *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering (ICSE '15)*, pp. 415–425, IEEE, May 2015.

## APPENDIX B

### EXPLANATION OF THE TYPICALLY EXCLUDED PAPERS

First, we did not include studies that focus on log analysis or the usage of log messages, which have been excluded according to E1 listed in Table 2.

For example, Xu et al. [32] proposed a general methodology to mine and analyze console logs to automatically detect system runtime problems. Yuan et al. [33] designed and implemented an effective diagnosis technique, which can analyze logs from a failed production run and source code to automatically generate useful information that assist engineers in diagnosing errors. Similarly, other studies like [34], [35], [36], [37], [38], [39], [40] focusing on this topic have been excluded from our study.

Second, we did not include studies that focus on the technologies that log and analyze the behaviors of users rather than software systems, which meet the exclusion criterion E2 listed in Table 2. For example, many studies on audit logs are outside the scope of our research because audit logs typically record the behavior of users. The purpose of such research is usually to ensure the security of the system [41], [42], [43]. King et al. [44] evaluated the use of a heuristics-driven method for identifying mandatory log events to support security analysts in performing forensic analysis. Similarly, some other studies have focused on the recording of user behaviors [45], [46] have been removed from the final study list.

Third, we did not include studies that did not explicitly discuss logging practices, which have been excluded according to exclusion criterion E3 listed in Table 2. For example, Rabkin et al. [47] described an abstract graphical representation of console logs called the identifier graph and a visualization based on this representation. Shang et al. [48] studied the relationship between logs and code quality, and the main focus of this paper is on the code quality rather than on logging practice. Some other studies [49], [50], [51] are excluded for similar reasons.

Last but not least, we did not include studies that are workshop papers, position papers, or ongoing work, which

have been excluded according to exclusion criterion E3–5 in Table 2. The typical examples for workshop papers are [52], [53], [54]. The typical examples for position papers or ongoing work are [55], [56], [57].