

REFERENCES

- [1] C. Jones, Software assessments, benchmarks, and best practices. Boston, Mass.: Addison-Wesley, 2000.
- [2] "Gartner Magic Quadrant for Software Test Automation 2019", Tricentis, 2020. [Online]. Available:<https://www.tricentis.com/resources/gartner-magic-quadrant-software-test-automation/>. [Accessed: 29- Mar- 2020].
- [3] P. Runeson, "A survey of unit testing practices", *IEEE Software*, vol. 23, no. 4, pp. 22-29, 2006. Available: 10.1109/ms.2006.91.
- [4] P. Louridas, "JUnit: Unit Testing and Coding in Tandem", *IEEE Software*, vol. 22, no. 4, pp. 12-15, 2005. Available: 10.1109/ms.2005.100.
- [5] R. Osherove, The art of unit testing, 2nd ed. Greenwich, CT.: Manning, 2014.
- [6] Rafael Lotufo, Leonardo Passos, Krzysztof Czarnecki, "Towards improving bug tracking systems with game mechanisms", *Mining Software Repositories (MSR) 2012 9th IEEE Working Conference on*, pp. 2-11, 2012.
- [7] Edward Aftandilian, Raluca Sauciuc, Siddharth Priya, and Sundaresan Krishnan. 2012. Building useful program analysis tools using an extensible java compiler. In Proceedings of the International Working Conference on Source Code Analysis and Manipulation (SCAM'12). 14--23.
- [8] K. Sen, D. Marinov and G. Agha, "CUTE", ACM SIGSOFT Software Engineering Notes, vol. 30, no. 5, p. 263, 2005. Available: 10.1145/1095430.1081750.
- [9] N. Rutar, C. B. Almazan, and J. S. Foster. A comparison of bug-finding tools for Java. In Proceedings of the 15th International Symposium on Software Reliability Engineering (ISSRE'04), pages 245–256. IEEE Computer Society Press, Nov. 2004
- [10] J. Langr, A. Hunt, D. Thomas and S. Davidson Pfalzer, Pragmatic unit testing in Java 8 with JUnit, 1st ed. The Pragmatic Programmers, LLC., 2015.
- [11] W. WONG and Y. QI, "BP NEURAL NETWORK-BASED EFFECTIVE FAULT LOCALIZATION", *International Journal of Software Engineering and Knowledge Engineering*, vol. 19, no. 04, pp. 573-597, 2009. Available: 10.1142/s021819400900426x.
- [12] "Models and layers | TensorFlow.js", TensorFlow, 2020. [Online]. Available: https://www.tensorflow.org/js/guide/models_and_layers. [Accessed: 02- Apr- 2020].

- [13] "Introduction to TensorFlow", TensorFlow, 2020. [Online]. Available: <https://www.tensorflow.org/learn>. [Accessed: 19- Jul- 2020].
- [14] K. Team, "Keras: the Python deep learning API", Keras.io, 2020. [Online]. Available: <https://keras.io/>. [Accessed: 19- Jul- 2020].
- [15] "What are microservices?", microservices.io, 2020. [Online]. Available: <https://microservices.io/>. [Accessed: 19- Jul- 2020].
- [16] "Foundation Project", Apache.org, 2021. [Online]. Available: <https://www.apache.org/foundation/>. [Accessed: 26- Mar- 2021].
- [17] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>.
- [18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS'12: Proc. of the Advances in Neural Information Processing Systems*, 2012
- [19] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition," in *ICASSP'12: Proc. of the International Conference on Acoustics, Speech and Signal Processing*, 2012.
- [20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, 1998
- [21] X. Huo, M. Li, and Z.H. Zhou, "Learning unified features from natural and programming languages for locating buggy source code," in *Proceedings of IJCAI'2016*.
- [22] V. Raychev, M. Vechev, and E. Yahav, "Code completion with statistical language models," in *ACM SIGPLAN Notices*, vol. 49, no. 6. ACM, 2014, pp. 419–428.
- [23] V. Raychev, P. Bielik, and M. Vechev, "Probabilistic model for code with decision trees," in *OOPSLA'16: Proc. of the International Conference on Object-Oriented Programming, Systems, Languages, and Applications*, 2016.
- [24] M. White, M. Tufano, C. Vendome, and D. Poshyanyk, "Deep learning code fragments for code clone detection," in *ASE'16: Proc. of the International Conference on Automated Software Engineering*, 2016.
- [25] L. Mou, G. Li, L. Zhang, T. Wang, and Z. Jin, "Convolutional neural networks over tree structures for programming language processing," *arXiv preprint arXiv:1409.5718*, 2014.

- [26] X. Gu, H. Zhang, D. Zhang, and S. Kim, “Deep api learning,” in *FSE’16: Proc. of the ACM SIGSOFT International Symposium on Foundations of Software Engineering*, 2016.
- [27] C. Liu, X. Chen, E. C. Shin, M. Chen, and D. Song, “Latent attention for if-then program synthesis,” in *NIPS’16: Proc. of the Advances in Neural Information Processing Systems*, 2016.
- [28] M. Balog, A. L. Gaunt, M. Brockschmidt, S. Nowozin, and D. Tarlow, “Deepcoder: Learning to write programs,” *arXiv preprint arX-iv:1611.01989*, 2016.
- [29] C. Shu and H. Zhang, “Neural programming by example,” in *AAAI’17: Proc. of the AAAI Conference on Artificial Intelligence*, 2017.
- [30] X. Yang, D. Lo, X. Xia, Y. Zhang, and J. Sun, “Deep learning for just-in-time defect prediction,” in *QRS’15: Proc. of the International Conference on Software Quality, Reliability and Security*, 2015.
- [31] D. Hovemeyer and W. Pugh, “Finding bugs is easy,” in *Companion to the Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA)*. ACM, 2004, pp. 132–136.
- [32] J.A. Zachman, “A Framework for Information Systems Architecture,” *IBM Systems J.*, vol.26, no. 3, 1987, pp. 276–292.
- [33] C. Calcagno, D. Distefano, J. Dubreil, D. Gabi, P. Hooimeijer, M. Luca, P. O’Hearn, I. Papakonstantinou, J. Purbrick, and D. Rodriguez, “Moving fast with software verification,” in *NASA Formal Methods Symposium*. Springer, 2015, pp. 3–11.
- [34] "What is reputation? How do I earn (and lose) it? - Help Center", *Stack Overflow*, 2021. [Online]. Available: <https://stackoverflow.com/help/whats-reputation>. [Accessed: 07- Apr- 2021].
- [35] M. Allamanis, E. T. Barr, P. Devanbu, and C. Sutton, “A survey of machine learning for big code and naturalness,” *ACM Computing Surveys (CSUR)*, vol. 51, no. 4, p. 81, 2018.
- [36] M. Weiser, “Programmers use slices when debugging,” *Communications of the ACM*, vol. 25, no. 7, pp. 446–452, Jul. 1982.
- [37] I. Vessey, “Expertise in debugging computer programs,” *International Journal of Man-Machine Studies: A Process Analysis*, vol. 23, no. 5, pp. 459–494, 1985.

- [38] W. E. Wong, V. Debroy, and B. Choi, "A family of code coverage-based heuristics for effective fault localization," *Journal of Systems and Software*, vol. 83, no. 2, pp. 188–208, February 2010.
- [39] W. E. Wong and V. Debroy, A Survey on Software Fault Localization Department of Computer Science, University of Texas, Dallas, Technical Report UTDCS-45-09, November 2009.
- [40] "Apache HBase – Apache HBase™ Home", *Hbase.apache.org*, 2021. [Online]. Available: <https://hbase.apache.org/>. [Accessed: 08- Apr- 2021].
- [41] "Apache Pinot™ (Incubating): Realtime distributed OLAP datastore | Apache Pinot™ (Incubating)", *Pinot.incubator.apache.org*, 2021. [Online]. Available: <https://pinot.incubator.apache.org/>. [Accessed: 07- Apr- 2021].
- [42] "Welcome to Apache Hudi !", 2021. [Online]. Available: <https://hudi.apache.org/>. [Accessed: 07- Apr- 2021].
- [43] "Distributed Database - Apache Ignite®", *Ignite.apache.org*, 2021. [Online]. Available: <https://ignite.apache.org/>. [Accessed: 09- Apr- 2021].
- [44] "Apache ZooKeeper", *Zookeeper.apache.org*, 2021. [Online]. Available: <https://zookeeper.apache.org/>. [Accessed: 07- Apr- 2021].
- [45] "spring-projects/spring-data-rest", *GitHub*, 2021. [Online]. Available: <https://github.com/spring-projects/spring-data-rest>. [Accessed: 09- Apr- 2021].
- [46] "spring-projects/spring-ws", *GitHub*, 2021. [Online]. Available: <https://github.com/spring-projects/spring-ws>. [Accessed: 07- Apr- 2021].
- [47] "Helidon Project", *Helidon.io*, 2021. [Online]. Available: <https://helidon.io/>. [Accessed: 07- Apr- 2021].
- [48] "Switching from Oracle OpenGrok to Sourcegraph - Sourcegraph docs", *Docs.sourcegraph.com*, 2021. [Online]. Available: https://docs.sourcegraph.com/code_search/how-to/opengrok. [Accessed: 09- Apr- 2021].
- [49] "Developer Guide - AWS SDK for Java 1.x - AWS SDK for Java", *Docs.aws.amazon.com*, 2021. [Online]. Available: <https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/welcome.html>. [Accessed: 07- Apr- 2021].
- [50] Y. S. Su and C. Y. Huang, "Neural-network-based approaches for software reliability estimation using dynamic weighted combinational models," *Journal of Systems and Software*, vol. 80, no. 4, pp. 606–615, April 2007.