

3.7 Study-5: Comparison of Characteristics (Proposed)

Motivation: In the previous subsections we have discussed which group of code, process, and structural characteristics metrics can be related to defects, and used for predicting defective IaC scripts. However, we have not discussed prediction performance-wise, which group of characteristics are better. Through systematic investigation we can identify which group of characteristics correlate more with defective IaC scripts, and can yield better defect prediction performance.

Methodology: We will perform our comparison using four performance measures: precision, recall, F-measure, and AUC. We will apply the Scott Knott Test to compare if the three types of characteristics namely, code, process, and structural characteristics perform significantly better than each other for the four performance measures. We will perform comparison using two evaluation schemes: (i) 10×10-fold cross validation, and (ii) cross dataset evaluation.

3.8 Study-6: Characteristics that Violate Security and Privacy Objectives (Proposed)

Motivation: As IaC scripts hold crucial information about the deployment environment, violation of security and privacy (S&P) objectives can be disastrous. We refer to characteristics of IaC scripts that violate S&P objectives as S&P-related anti-patterns. As an example anti-pattern, if administrator credentials are hard-coded in IaC scripts, attackers can use those credentials and hack into the deployment infrastructure. Systematic investigation can help in identifying which S&P-related anti-patterns occur in IaC scripts and at which frequency.

Methodology: As the first step to extract the S&P-related anti-patterns, we will apply grounded theory analysis [2]. Next, we will create an automated tool that will identify the S&P-related anti-patterns in IaC scripts using syntax-driven techniques. We will add custom heuristics derived from our analysis and extend existing commercial tools such as puppet-lint⁵, and ansible-lint⁶.

4 TIMELINE

The author of the publication is a fourth-year PhD student, who passed his PhD proposal in January, 2018. He has completed Study-1, and 2, and they are currently under review respectively, at the IEEE Transactions on Software Engineering journal, and the Empirical Software Engineering journal. He aims to submit Study-3, 4, 5, and 6, in prestigious conferences and journals related to software engineering.

REFERENCES

- [1] Christian Bird, Nachiappan Nagappan, Brendan Murphy, Harald Gall, and Premkumar Devanbu. 2011. Don'T Touch My Code!: Examining the Effects of Ownership on Software Quality. In *Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering (ESEC/FSE '11)*. ACM, New York, NY, USA, 4–14.
- [2] Kathy Charmaz. 2014. *Constructing grounded theory*. Sage Publishing, London, UK.
- [3] R. Chillarege, I. S. Bhandari, J. K. Chaar, M. J. Halliday, D. S. Moebus, B. K. Ray, and M. Y. Wong. 1992. Orthogonal defect classification—a concept for in-process measurements. *IEEE Transactions on Software Engineering* 18, 11 (Nov 1992), 943–956.
- [4] MITRE Corporation. 2017. Common Weakness Enumeration. <https://cwe.mitre.org/>. (2017). [Online; accessed 10-November-2017].
- [5] Wei Fu, Tim Menzies, and Xipeng Shen. 2016. Tuning for software analytics: Is it really necessary? *Information and Software Technology* 76 (2016), 135 – 146.
- [6] Oliver Hanappi, Waldemar Hummer, and Schahram Dustdar. 2016. Asserting Reliable Convergence for Configuration Management Scripts. *SIGPLAN Not.* 51, 10 (Oct. 2016), 328–343.
- [7] P. Hudak. 1998. Modular domain specific languages and tools. In *Proceedings, Fifth International Conference on Software Reuse (Cat. No.98TB100203)*. 134–142.
- [8] Jez Humble and David Farley. 2010. *Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation* (1st ed.). Addison-Wesley Professional.
- [9] Yujuan Jiang and Bram Adams. 2015. Co-evolution of Infrastructure and Source Code: An Empirical Study. In *Proceedings of the 12th Working Conference on Mining Software Repositories (MSR '15)*. IEEE Press, Piscataway, NJ, USA, 45–55.
- [10] Nachiappan Nagappan and Thomas Ball. 2005. Use of Relative Code Churn Measures to Predict System Defect Density. In *Proceedings of the 27th International Conference on Software Engineering (ICSE '05)*. ACM, New York, NY, USA, 284–292.
- [11] Nachiappan Nagappan, Brendan Murphy, and Victor Basili. 2008. The Influence of Organizational Structure on Software Quality: An Empirical Case Study. In *Proceedings of the 30th International Conference on Software Engineering (ICSE '08)*. ACM, New York, NY, USA, 521–530.
- [12] Alberto S. Nunez-Varela, Hector G. Perez-Gonzalez, Francisco E. Martinez-Perez, and Carlos Soule-Mulle-Morales. 2017. Source code metrics: A systematic mapping study. *Journal of Systems and Software* 128 (2017), 164 – 197.
- [13] C. Parnin, R. Helms, C. Lange, H. Boughton, M. Ghattas, A. Glover, J. Holman, M. Micco, B. Murphy, T. Scalet, M. Stumm, S. Whitaker, and L. Williams. 2017. The Top 10 Antipatterns in Continuous Deployment. *IEEE Software* 34, 3 (May 2017), 86–95.
- [14] M. Pelech and S. Rana. 2012. Detection of Software Failures through Event Logs: An Experimental Study. In *2012 IEEE 23rd International Symposium on Software Reliability Engineering*. 31–40.
- [15] Akond Rahman, Asif Partho, David Meder, and Laurie Williams. 2017. Which Factors Influence Practitioners' Usage of Build Automation Tools?. In *Proceedings of the 3rd International Workshop on Rapid Continuous Software Engineering (CoSE '17)*. IEEE Press, Piscataway, NJ, USA, 20–26.
- [16] Akond Rahman and Laurie Williams. 2018. Characterizing Defective Configuration Scripts Used for Continuous Deployment. In *2018 IEEE International Conference on Software Testing, Verification and Validation (ICST)*. To appear. Pre-print: http://akondrahman.github.io/papers/icst2018_tm.pdf.
- [17] Brian Shambaugh, Aaron Weiss, and Arjun Guha. 2016. Rehearsal: A Configuration Verification Tool for Puppet. *SIGPLAN Not.* 51, 6 (June 2016), 416–430.
- [18] Tushar Sharma, Marios Fragkoulis, and Diomidis Spinellis. 2016. Does Your Configuration Code Smell?. In *Proceedings of the 13th International Conference on Mining Software Repositories (MSR '16)*. ACM, New York, NY, USA, 189–200.
- [19] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. 2005. *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- [20] Chakkrit Tantithamthavorn, Shane McIntosh, Ahmed E. Hassan, and Kenichi Matsumoto. 2016. Automated Parameter Optimization of Classification Techniques for Defect Prediction Models. In *Proceedings of the 38th International Conference on Software Engineering (ICSE '16)*. ACM, New York, NY, USA, 321–332.
- [21] John Viega and Gary McGraw. 2011. *Building Secure Software: How to Avoid Security Problems the Right Way (Paperback)* (Addison-Wesley Professional Computing Series) (1st ed.). Addison-Wesley Professional.
- [22] Feng Zhang, Audris Mockus, Iman Keivanloo, and Ying Zou. 2016. Towards Building a Universal Defect Prediction Model with Rank Transformed Predictors. *Empirical Softw. Engg.* 21, 5 (Oct. 2016), 2107–2145.
- [23] Thomas Zimmermann, Rahul Premraj, and Andreas Zeller. 2007. Predicting Defects for Eclipse. In *Proceedings of the Third International Workshop on Predictor Models in Software Engineering (PROMISE '07)*. IEEE Computer Society, Washington, DC, USA, 9–15.

⁵<http://puppet-lint.com/>

⁶<https://github.com/willthames/ansible-lint>