

Jeongju Sohn

Assistant Professor

Research Interests

Software Engineering, Search Based Software Engineering, Fault Localisation, Defect Prediction, Software Evolution, Code Change Analysis

Education

- 2017.03 – **PhD. School of Computing**, Korea Advanced Institute of Science and Technology
2021.08 (KAIST), Republic of Korea
- 2015.03 – **MSc. School of Computing**, Korea Advanced Institute of Science and Technology
2017.02 (KAIST), Republic of Korea
- 2011.03 – **BSc. Computer Science and Engineering**, Ewha Womans University, Republic
2015.02 of Korea

Experience

- 2015.03 – **Database and Multimedia Lab, KAIST**, Research on object-relational database
2016.01 management system specialised in big data analysis: application and evaluation of
the ORDBMS in processing artificial web application queries
- 2016.01 – **Computational Intelligence for Software Engineering Lab, KAIST**, Completed
2021.10 Master and Ph.D. Mainly worked on various topics of software debugging, especially
on fault localization and defect prediction
- 2021.10 – **SerVal, SnT, University of Luxembourg**, Joined Serval Group as a research
2024.02 associate since October, 2021. Mainly working on software evolution.
- 2024.03 – **School of Computer Science and Engineering at Kyungpook National Uni-
versity**, Appointed as an Assistant Professor in School of Computer Science and
Engineering at Kyungpook National University

Academic Service

Reviewer for international journals Journal of Systems and Software (2), Information and Software Technology (4), Transaction on Software Engineering and Methodology (9), Transactions on Software Engineering (4), Software Testing, Verification and Reliability (1), Empirical Software Engineering (10)

PC Member ESEC/FSE 2021 (Demonstrations track)
RAISE 2021
ICSME 2021 (Artefact Evaluation)
GECCO 2022, (GI Workshop)
SCAM 2022, (RENE)
ISSTA 2022, (Tool Demonstration)
FSE 2022 (Student Research Competition)
ISSTA 2023 (Tool Demonstration)
ICDM-AIOPS workshop 2023
SSBSE 2023
GI@ICSE workshop 2023
ICSE 2024
GI@ICSE workshop 2024
ISSTA 2024

Program SSBSE 2022 (RENE/NIER)
Chair ISSTA 2025 (Tool Demonstration)

Publication

Journals

- Sohn, J., Kang, S., Yoo, S. (2023) *Arachne: Search-Based Repair of Deep Neural Networks*. *ACM Trans. Softw. Eng. Methodol.* 32, 4, Article 85, 26 pages
- Sohn, J. and Yoo, S. (2019). *Empirical Evaluation of Fault Localisation Using Code and Change Metrics*. *IEEE Transactions on Software Engineering.* 47, 8, 1605–1625.

Conferences

- Morita, I., Kashiwa, Y., Kondo, M., Sohn, J., McIntosh, S., Kamei, Y., Ubayashi, N., *TraceJIT: Evaluating the Impact of Behavioral Code Change on JIT Defect Prediction*, *Proceedings of the IEEE 28th International Conference on Software Analysis, Evolution, and Reengineering*. Rovaniemi, Finland
- Pei, Y., Sohn, J., Habchi, S., Papadakis, M. (2023) *Time-based Repair for Asynchronous Wait Flaky Tests in Web Testing*, arXiv:2305.08592
- Sohn, J., Papadakis, M. (2022). *CEMENT: On the use of Evolutionary Coupling between tests and code units. A case study on fault localization*, *Proceedings of the 33rd IEEE International Symposium on Software Reliability Engineering*. Charlotte, North Carolina, USA
- Habchi, S., Haben, G., Sohn, J., Franci, A., Papadakis, M., Cordy, M., Traon, Y. (2022). *What Made This Test Flake? Pinpointing Classes Responsible for Test Flakiness*, *Proceedings of the 38th IEEE International Conference on Software Maintenance and Evolution*. Limassol, Cyprus
- An, G., Yoon, J., Sohn, J., Hong, J., Hwang, D., Yoo, S., (2022). *Automatically Identifying Shared Root Causes of Test Breakages in SAP HANA*, *Proceedings of the IEEE/ACM 44th International Conference on Software Engineering: Software Engineering in Practice*. (pp. 65-74). Pittsburgh, USA

- Sohn, J., An, G., Hong, J., Hwang., D, Yoo, S. (2021). *Assisting Bug Report Assignment Using Automated Fault Localisation: An Industrial Case Study*, *Proceedings of the International Conference on Software Testing* (pp. 284-294). Virtual
- Sohn, J., Kamei, Y., McIntosh, S., Yoo, S. (2021). *Leveraging Fault Localisation to Enhance Defect Prediction*, *Proceedings of the IEEE 28th International Conference on Software Analysis, Evolution, and Reengineering* (pp. 284-294). Virtual
- Sohn, J. (2020). *Bridging Fault Localisation and Defect Prediction*, *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Companion Proceedings* (pp. 214-217). Virtual
- Sohn, J. and Yoo, S. (2019). *Why Train-and-Select When You Can Use Them All: Ensemble Model for Fault Localisation*. *Proceedings of the Genetic and Evolutionary Computation Conference* (pp. 1408-1416). Prague, Czech
- Choi, K., Sohn, J., Yoo, S. (2018). *Learning Fault Localisation for Both Humans and Machines using Multi-Objective GP*. *Proceedings of the 10th International Symposium on Search Based Software Engineering* (pp. 349-355). Montpellier, France
- Kang, D., Sohn, J., Yoo, S. (2017). *Empirical evaluation of conditional operators in GP based fault localization*. *Proceedings of the Genetic and Evolutionary Computation Conference* (pp.1295-1302). Berlin, Germany
- Sohn, J., Yoo, S. (2017). *FLUCCS: Using Code and Change Metrics to Improve Fault Localization*. *Proceeding of the 2017 International Symposium on Software Testing and Analysis* (pp. 273-283). Santa Barbara, CA, USA
- Sohn, J., Lee, S., Yoo, S. (2016). *Amortised Deep Parameter Optimisation of GPGPU Work Group Size for OpenCV*. *Search-Based Software Engineering*. Springer International Publishing (pp. 211-217). Raleigh, NC, USA

Workshops

- Pei, Y., Habchi. S., Rwemalika, R., Sohn, J., Papadakis, M. (2022). *An empirical study of async wait flakiness in front-end testing*, *21st Belgium-Netherlands Software Evolution Workshop*

Ongoing and Future (*) Projects

UNITE: Considering fault localisation and fault prediction as pre- and post-identification of Unified Fault Identification through Test and Code Co-evolution framework for evolving software systems. The project will establish and leverage the co-evolution of test and code elements to connect fault-identification activities in the past and present, aiming to improve overall software quality through simultaneous interactions between these identification activities. This project was submitted and recently accepted at the CORE JUNIOR track.

Explainable Debugging Using Large Language Models to automatically generate an explanation of why a given failure occurs in natural language by exploiting past relevant changes and issues in addition to the current failure information.

Latent Mutant	<i>Latent mutants</i> are the mutants dormant at the time of introduction but later revealed in the project timeline by new failing tests. We investigate the frequency and characteristics of latent mutants, building a simple prediction model. We further explore the potential benefit of predicting these mutants, e.g., technical debt reduction.
Robust Locators of Web Test Cases	Identifying robust locators in web test cases using the change rate of locators based on their past evolution history. The project further aims to automatically synthesise robust locators by analysing the patterns of the identified locators.
Relevant Mutant Identification	Identifying a set of mutants that are likely to be relevant (i.e., valuable) for a given code snippet, for instance, those that can reveal hidden bugs in the code, without any execution. The objective is to reduce the cost of mutation testing.

Past Projects

FLUCCS	Using code and change metrics that have been widely studied in the context of defect prediction to improve the accuracy of fault localisation. The main goal of FLUCCS is finding the connections between defect prediction and fault localisation, which could benefit both tasks.
FL2DP	Leveraging fault localisation to improve defect prediction. FL2DP, which stands for Fault Localisation to Defect Prediction, aims to enhance Just-In-Time defect prediction by exploiting the insights about defective code from past fault localisation.
with SAP Labs Korea	Industrial case study of applying existing fault localisation techniques to aid the bug report assignment phase in the Continuous Integration (CI) pipe-line of SAP HANA2 Database. Collaboration with SAP Labs Korea.
EMF	Using ensemble learning to improve the performance of learn-to-rank algorithm-based fault localisation against a wider range of faults. The main goal of EMF is to overcome the limitation of a single ranking model that can be overfitted to a certain set of faults by aggregating the results of multiple ranking models.
with NSRI	Classifying abnormal flows in network by exploiting contextual information in the flow using Long Short-term Memory. Collaboration with National Security Research Institute (NSRI), Korea.
Sherlock	Identifying the root cause of flaky tests using Spectrum-Based Fault Localisation
Arachne	Search-based repair of deep neural network. Compared to previous automatic DNN repair techniques, Arachne targets to repair a specific type of faults of a DNN model (e.g., classify input A to B) at the minimised expense of initially correct behaviour without requiring additional data. Arachne can be effective in situations where some misbehaviour of a DNN model must be fixed even at the cost of breaking other less critical behaviour.
CEMENT	Mining evolutionary couplings between test and code association by analysing how the software has evolved. The main objective of this project is to mine the <i>general</i> association between test and code, which describes their evolutionary coupling, and apply this association to various software debugging tasks.

Automated Repair of Async Wait flakiness Asynchronous wait is one of the most frequent causes of flaky test failures. In this project, we aim to automatically identify and repair the root cause of async wait flaky test failures, going beyond simply generating hot-fixes for them.