MAX PLANCK INSTITUTE
FOR SECURITY AND PRIVACY

December 2, 2024

Seongmin Lee, Ph.D.
Research Fellow
Max Planck Institute for Security and Privacy
Universitätsstraße 140
Bochum, Nordrhein-Westfalen 44799
Germany
Phone: (+49) 1771835480
Email: seongmin.lee@mpi-sp.org

Dear Faculty Hiring Committee,

I am writing to apply for a Tenure-track Assistant Professor position in the Siebel School of Computing and Data Science at the University of Illinois at Urbana-Champaign (UIUC).

I am a postdoctoral researcher at the Max Planck Institute for Security and Privacy, where I conduct research on **software engineering**, focusing on **program analysis** and **software testing**. My goal is to develop scalable and reliable methodologies to ensure software correctness in complex and critical modern software systems. To achieve this, I employ interdisciplinary statistical techniques to analyze program behavior, enhancing the precision and robustness of analysis and testing. I am particularly inspired by the Grainger College of Engineering's vision to lead transformative innovation and tackle global challenges through interdisciplinary collaboration and impactful research. UIUC's commitment to excellence, inclusivity, and fostering bold ideas aligns closely with my goal of developing transformative methodologies to ensure software correctness in modern software systems, while mentoring the next generation of engineering leaders in a dynamic research environment.

Research: My research applies statistical methods, such as causal inference and biostatistics, to analyze software behavior, addressing the fundamental limitations of conventional software engineering techniques in practice. This work has led to two key contributions: 1) Addressing the unreliability of software testing caused by insufficient test coverage, I estimated the quantitative likelihood of residual risks in software systems using discovery probability estimation from biostatistics. 2) Overcoming the limited scalability of conventional static analysis, I uncovered hidden dependencies between software components by establishing a novel dependency analysis framework based on counterfactual causal reasoning. My work has been published in top-tier venues, including ICSE, ESEC/FSE, and JSS, and is supported by a research grant from the German Research Foundation (DFG) under Germany's Excellence Strategy, amounting to approximately €136,000.

If I were to join the University of Illinois at Urbana-Champaign, I would continue advancing software engineering by integrating advanced statistical techniques into the field. My vision is to drive a paradigm shift by establishing a statistical foundation for software analysis, leveraging interdisciplinary methods to address challenges where conventional approaches fall short. I am confident in achieving this goal, particularly in an era of big data in software engineering, where overlooked statistical methods offer unique contributions, such as explainability and addressing data scarcity—needs that have become increasingly critical in areas less served by extensively studied ML-like techniques.

Teaching and Research Mentoring: I have served as a teaching assistant six times for four courses at the Korea Advanced Institute of Science and Technology (KAIST), spanning introductory computer science to advanced software engineering. In this role, I co-designed course materials, led weekly discussions, and graded assignments. As a research mentor, I have guided three undergraduate students and one Ph.D. student through projects that resulted in three peer-reviewed publications, including two first-authored by the students, at top-tier software engineering venues.

Service: I am committed to contributing to both academia and institutional service. I have served as a program committee member for top-tier conferences, including FSE, ASE, and ISSTA, as well as a reviewer for leading journals such as TSE and TOSEM. Additionally, I serve as an Open Science Ambassador at the Max Planck Society, where I promote Open Science practices and provide valuable resources to colleagues. I am also a member of the Early Career Researcher Board at the Cyber Security in the Age of Large-Scale Adversaries (CASA) Graduate School, where I help organize activities and address priorities for early career researchers.

Enclosed for your review are my (1) a complete curriculum vitae, (2) a research statement, (3) a teaching statement, and (4) a statement on commitment to diversity, along with names and contact information of three references who will be contacted to provide letters. My achievements in research, mentorship, and collaboration demonstrate my readiness for a tenure-track faculty position. I am excited about the opportunity to continue my research and contribute to the academic community at the University of Illinois at Urbana-Champaign. Thank you for considering my application.

Sincerely yours,

[illegible]

Seongmin Lee

Seongmin Lee

Max Planck Institute for Security and Privacy (MPI-SP)
Universitätsstraße 140
44799 Bochum
Germany

+49 177 783 5480 seongmin.lee@mpi-sp.org Google Scholar

Research Summary

The overarching objective of my research is to *ensure software correctness in complex and critical modern software systems* through **scalable program analysis** and **reliable software testing**. To achieve this, I **leverage interdisciplinary statistical methods**, such as *causal inference, biostatistics, and machine learning*, to analyze the dynamic behavior of software in operational environments. My work has been published in top-tier software engineering venues, including ICSE, FSE, and JSS, and I have served as a program committee member for leading conferences, such as FSE, ASE, and ISSTA.

Education and Employment

Max Planck Institute for Security and Privacy — Germany
Postdoctoral Researcher, Software Security Research group — Sep. 2022 – Present
Group head: Dr. Marcel Böhme

Korea Advanced Institute of Science and Technology — Republic of Korea
Doctor of Philosophy, School of Computing — Sep. 2016 – Aug. 2022
Advisor: Dr. Shin Yoo

Bachelor of Science, School of Computing — Feb. 2012 – Aug. 2016
Bachelor of Science, Department of Mathematical Sciences

Publications

Refereed Journal Articles

SCP'25 **Seongmin Lee**, Dave Binkley, Robert Feldt, Nicolas Gold, and Shin Yoo. Causal program dependence analysis. *Science of Computer Programming*, 240:103208, February 2025

JSS'21 **Seongmin Lee**, David Binkley, Robert Feldt, Nicolas Gold, and Shin Yoo. Observation-based approximate dependency modeling and its use for program slicing. *Journal of Systems and Software*, 179:110988, September 2021

JSS'20 **Seongmin Lee**, David Binkley, Nicolas Gold, Syed Islam, Jens Krinke, and Shin Yoo. Evaluating lexical approximation of program dependence. *Journal of Systems and Software*, 160:110459, February 2020

Refereed Conference Publications

ICSE'25 **Seongmin Lee**, Shreyas Minocha, and Marcel Böhme. Accounting for Missing Events in Statistical Information Leakage Analysis. In *Proceedings of the IEEE/ACM 47th International Conference on Software Engineering*, ICSE '25, pages 1–12, New York, NY, USA, 2025. Association for Computing Machinery

ICSE'24 Danushka Liyanage*, **Seongmin Lee***, Chakkrit Tantithamthavorn, and Marcel Böhme. Extrapolating Coverage Rate in Greybox Fuzzing. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, ICSE '24, pages 1–12, New York, NY, USA, April 2024. Association for Computing Machinery (**Co-first authors with equal contribution*)

FSE'23 **Seongmin Lee** and Marcel Böhme. Statistical Reachability Analysis. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2023, pages 326–337, New York, NY, USA, November 2023. Association for Computing Machinery

SCAM'19 **Seongmin Lee**, David Binkley, Robert Feldt, Nicolas Gold, and Shin Yoo. MOAD: Modeling Observation-Based Approximate Dependency. In *2019 19th International Working Conference on Source Code Analysis and Manipulation (SCAM)*, pages 12–22, September 2019

ICST'19 **Seongmin Lee**, Shin Hong, Jungbae Yi, Taeksu Kim, Chul-Joo Kim, and Shin Yoo. Classifying False Positive Static Checker Alarms in Continuous Integration Using Convolutional Neural Networks. In *2019 12th IEEE Conference on Software Testing, Validation and Verification (ICST)*, pages 391–401, April 2019

Preprints

Seongmin Lee and Marcel Böhme. Structure-aware Residual Risk Analysis, 2025 – *Under review with ISSTA'25*

Jing Liu*, **Seongmin Lee***, Eleonora Losiouk, and Marcel Böhme. Can LLM Generate Regression Tests for Software Commits?, 2025 – *Under review with FSE'25*
(**Co-first authors with equal contribution*)

Seongmin Lee and Marcel Böhme. How Much is Unseen Depends Chiefly on Information About the Seen, February 2024 – *Under review with ICLR'25, Initial review score: [8, 8, 6] (8: Accept, 6: Weak Accept)*

Invited Articles

GI'20 William B. Langdon, Westley Weimer, Justyna Petke, Erik Fredericks, **Seongmin Lee**, Emily Winter, Michail Basios, Myra B. Cohen, Aymeric Blot, Markus Wagner, Bobby R. Bruce, Shin Yoo, Simos Gerasimou, Oliver Krauss, Yu Huang, and Michael Gerten. Genetic Improvement @ ICSE 2020. *SIGSOFT Softw. Eng. Notes*, 45(4):24–30, October 2020

Refereed Workshop Publications

ICST'21 Saeyoon Oh, **Seongmin Lee**, and Shin Yoo. Effectively Sampling Higher Order Mutants Using Causal Effect. In *2021 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pages 19–24, April 2021

ICSE'20 **Seongmin Lee**. Scalable and approximate program dependence analysis. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Companion Proceedings*, ICSE '20, pages 162–165, New York, NY, USA, October 2020. Association for Computing Machinery

KCC'19 Gabin An, Jinhan Kim, **Seongmin Lee**, and Shin Yoo. PYGGI: Python General Framework for Genetic Improvement. *Journal of Korean Institute of Information Scientists and Engineers*, pages 536–538, December 2017

ICSE'18 **Seongmin Lee**, David Binkley, Nicolas Gold, Syed Islam, Jens Krinke, and Shin Yoo. MOBS: Multi-operator observation-based slicing using lexical approximation of program dependence. In *Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings*, ICSE '18, pages 302–303, New York, NY, USA, May 2018. Association for Computing Machinery

SBSE'17 **Seongmin Lee** and Shin Yoo. Hyperheuristic Observation Based Slicing of Guava. In Tim Menzies and Justyna Petke, editors, *Search Based Software Engineering*, pages 175–180, Cham, 2017. Springer International Publishing

SBSE'16 Jeongju Sohn, **Seongmin Lee**, and Shin Yoo. Amortised Deep Parameter Optimisation of GPGPU Work Group Size for OpenCV. In Federica Sarro and Kalyanmoy Deb, editors, *Search Based Software Engineering*, pages 211–217, Cham, 2016. Springer International Publishing

Grants and Fellowships

- Title: *Statistical Security Analysis for Large, Evolving Software*
  Funding Agency: CASA - Cyber Security in the Age of Large-Scale Adversaries
  Grant ID: DFG under Germany's Excellence Strategy - **EXC 2092 CASA - 390781972**
  Amount: Salary according to the remuneration group E 14 TV-L (full time, ∼ €136,000)
  Duration: 2024.01.01 – 2025.12.31
  Role: Sole Principal Investigator (PI)

Awards and Honors

- **Distinguished Artifact Reviewer Award**, 33rd USENIX Security Symposium, 2024
- **PhD Dissertation Award**, School of Computing, KAIST, 2022
  - *Title of Dissertation: Statistical Program Dependence Approximation*
- **2021 Naver Ph.D. Fellowship Award**: Awarded by NAVER Corp. to Ph.D. candidates who have published an outstanding research paper or have excellent publication performance, 2021
- Government-sponsored Scholarship, Ministry of Science and ICT of Korea, 2016 - 2022
- Government-sponsored Scholarship, Ministry of Science and ICT of Korea, 2012 - 2016

Service

Academic Service

- Program committee: (Main Track) CauSE'25, ASE'24, ISSTA'24, FUZZING'24, SCAM'24, ASE'23 / (Artifact Evaluation Track) ISSTA'24, ECOOP'24, USENIX Security'24, ICSE'24, ISSTA'23, ICSME'22, ICSME'21 / (Student Research Competition Track) FSE'24 / (Tool Demonstration Track) ASE'24
- Reviewer: TOSEM'24, TSE'24, IST'24, ASE'24, TOSEM'22, JSS'21, JSS'20 / (External) ICSE'24, FSE'24, ECOOP'24, ICSE'23, ISSTA'23

Institutional Service

- Early Career Researcher (ECR) Board, CASA - Cyber Security in the Age of Large-Scale Adversaries 2024 – Present

  The ECR Board decides all funding allocations for the CASA Graduate School (CGS). The board has its own budget and the autonomy to dynamically adapt the programs to the changing needs of ECR members. Representing ECRs in CASA governance, the board organizes activities and meets quarterly to address priorities.
- Open Science Ambassador, Max Planck Institute for Security and Privacy 2023 – Present

  As representatives of the institute, Open Science Ambassadors raise awareness and provide valuable information to their colleagues about Open Science practices. They meet annually, in person or online, to discuss strategies, participate in workshops, and evaluate the status of Open Science within and beyond the Society.

Advising Experience

Ph.D. Students

- **Danushka Liyanage** (Monash University, moved on to University of Sydney Postdoc, 2024) Nov. 2022 – Dec. 2023

  Co-advised with Dr. Marcel Böhme on extrapolating the coverage rate of the Greybox Fuzzing using the statistical model. A **full conference paper** where Danushka is the first author has been accepted to ICSE'24.

Undergraduate Students

- **Jing Liu** (Shanghai University, incoming Ph.D. student in UC Irvine) Apr. 2024 – Present

  Co-advised with Dr. Marcel Böhme on the LLM-based regression test generation for the software testing. A **full conference paper** where Jing is the first author has been submitted to FSE'25.
- **Shreyas Minocha** (Rice University, moved on to Georgia Tech Ph.D. student, 2024) Feb. 2023 – Jul. 2024

  Primary advisor on the statistical information leakage analysis to suggest the accurate and safe information leakage estimator. A **full conference paper** has been accepted to ICSE'25.
- **Saeyoon Oh** (KAIST, moved on to FuriosaAI) Jul. 2020 – Apr. 2021

  Co-advised with Dr. Shin Yoo on the effective sampling of higher-order mutants for mutation testing. A **workshop paper** where Saeyoon is the first author has been accepted to ICST'21.

Teaching Experience

Guest Lecturer

- *Guarantees in Software Security – 2nd part: Extrapolating Software Testing*,
  Fuzzing and Software Security Summer School @ National University of Singapore (NUS) May 2024

Invited Talks

- *Statistical Program Analysis*, Korea Advanced Institute of Science and Technology (KAIST) Jan. 2024
- *Statistical Program Analysis*, Ulsan National Institute of Science and Technology (UNIST) Jan. 2024
- *Causal Program Dependence Analysis*, Sheffield Causality and Testing Workshop Sep. 2023
- *Statistical program dependence analysis*, Handong Global University Aug. 2022
- *Observation-based approximate dependency modeling and its use for program slicing*,
  Korea Conference on Software Engineering Jan. 2022
- *MOBS: Multi-Operator Observation-Based Slicing using Lexical Approximation of Program Dependence*,
  59th CREST Open Workshop – Multi-language Software Analysis Mar. 2018

Teaching Assistant

- Automated Software Testing (CS453), School of Computing (SoC), KAIST Spring 2019

- Artificial Intelligence Based Software Engineering (CS454), SoC, KAIST — Fall 2018
- Introduction to Logic for Computer Science (CS402), SoC, KAIST — Spring 2018
- Artificial Intelligence Based Software Engineering (CS454), SoC, KAIST — Fall 2017
- Introduction to Logic for Computer Science (CS402), SoC, KAIST — Spring 2017
- Special Topics in Computer Science ⟨Search Based Software Engineering⟩ (CS492), SoC, KAIST — Fall 2016

Research Experience

Software Security Group, MPI-SP — **Sep. 2022 – Present**
Postdoc — *Bochum, Germany*

- Working on unbiased estimation of the missing mass, probability, or expected number of newly discovered classes in an unknown multinomial distribution
- Working on structure-aware residual risk estimation to achieve more accurate and efficient software security testing
- Worked on LLM-based regression test generation for software security testing – Cleverest
- Worked on applying biostatistics to information leakage analysis to suggest an accurate and safe information leakage estimator – Statistical Information Leakage Analysis
- Worked on extrapolating the coverage rate of Greybox Fuzzing using a statistical model – Greybox Fuzzing Extrapolation
- Worked on applying statistical methods to program analysis to overcome the scalability issues of static analysis – Statistical Reachability Analysis

Computational Intelligence for Software Engineering Laboratory (COINSE), KAIST — **Sep. 2016 – Aug. 2022**
Ph.D. Student — *Daejeon, Republic of Korea*

- Worked on approximating the degree of dependence between program elements using causal inference – CPDA
- Worked on applying statistical models to observational data to approximate program dependence – MOAD
- Worked on inferring type information in binary executables using RNNs in collaboration with the National Security Research Institute
- Worked on classifying false-positive alarms from a static checker in a continuous integration pipeline using CNNs in collaboration with Samsung Research
- Worked on program dependence approximation using a lexical model on source code – MOBS

Computational Intelligence for Software Engineering Laboratory (COINSE), KAIST — **Mar. 2016 – Aug. 2016**
Undergraduate Research Intern — *Daejeon, Republic of Korea*

- Worked on the amortized deep parameter optimization of GPGPU workgroup sizes for OpenCV.
- Accelerated the scalablility of Observation based slicing (ORBS) by applying a code distance metric during the slicing.

Programming Language Research Group (PLRG) Lab, KAIST — **Jul. 2015 – Feb. 2016**
Undergraduate Research Intern — *Daejeon, Republic of Korea*

- Developed a source code translator from C# to C++ with F#.
- Developed a frontend of Scalable Analysis Framework for ECMAScript (SAFE), a Javascript static analysis tool.

I specialize in **software engineering**, with a particular focus on **program analysis** and **software testing**. Ensuring software correctness is essential, as software increasingly governs critical aspects of modern life. To achieve this, we must analyze whether a program's behavior aligns with its intended purpose and rigorously test its functionality. However, traditional formal-semantics-based program analysis often struggles with scalability when addressing the complexity of modern software systems. At the same time, empirical methods like software testing, while practical, inevitably miss certain behaviors, leaving critical gaps in verification. The overarching objective of my research is to develop **scalable and reliable methodologies** that bridge these gaps. To achieve this, I employ **interdisciplinary statistical techniques to analyze dynamic information from program execution**, advancing the precision and robustness of software systems.

Over the past few decades, the widespread adoption of *program analysis* and software testing has become integral to ensuring the reliability and security of software applications. Conventional program analysis relies on formal semantics, which assigns rigorous mathematical meaning to the syntax of a programming language, to deduce a program's semantic features. However, formal semantics are limited in their ability to handle the heterogeneous features prevalent in modern software, such as network communication, system-level behavior, and third-party libraries, which are often beyond what formal semantics cover.

Software testing, which finds defects by actively executing the software, has gained notable attention since formally proving correctness is often unfeasible due to the vast program state. However, its reliability is challenged by its inherent incompleteness: there is always an unseen behavior in the software, and whether there is an undetected defect and if the testing process will find it, given the limited number of test cases, is unknown.

My research addresses the inherent limitations of program analysis and software testing fundamentally by reframing these tasks as *statistical problems* and solve them by employing the statistical methods to the dynamic information from the program execution. The primary advantage of *inferring* the program behavior statistically for program analysis is that they *operate irrespective of the system's complexity*, even in cases where the entire system is unknown, inaccessible, and/or undecidable. Statistical inference for software testing, focusing on the distribution of program executions in operational environments, provides *predictions* and *guarantees* for the software testing process in practice. In addressing the aforementioned limitations, I leverage a diverse range of statistical methods, including *causal inference*, *biostatistics*, and *machine learning*, drawn from fields such as *ecology*, *linguistics*, and *social sciences*. These methods are not only adapted but also customized to address the intricacies of software engineering.

I have pursued two problem-driven research directions addressing challenges in program analysis and software testing, along with a theoretical direction integrating interdisciplinary statistical methods into software engineering:

- **Counterfactual program analysis [1, 2, 3, 9, Section 1]**: I have developed statistical methodologies to infer program dependencies, determining which program elements affect others, by considering counterfactual events in sample program executions. Our work demonstrates that these statistical methods can identify dependencies not recognized by conventional program analysis.
- **Reasoning the unseen in software testing [4, 6, 7, 8, Section 2]**: I have developed statistical methodologies to estimate the likelihood of unseen events in software testing, providing reliable interpretations of testing results and predicting the future performance of the testing process. These methodologies outperform existing models based on formal semantics and statistical approaches that mishandle the unseen in practical scenarios.
- **Refining interdisciplinary statistical methods for SE [2, 4, 5, 6, Section 3]**: Statistical models from fields like social science and ecology offer valuable insights, but software's distinct traits—discreteness, determinism, and structural dependencies—require tailored approaches. To address these challenges, we develop specialized methodologies to enhance the applicability of statistical methods in software engineering.

Long-term Vision. Building on a foundation of scalable and reliable program analysis and software testing, my research integrates statistical methods with dynamic program information to address the challenges of modern software systems. **This is just the beginning: advancing statistical methodologies in software engineering promises to tackle enduring challenges and open transformative research avenues**. Modern software development generates vast amounts of data, and leveraging this data—particularly through machine learning—has become standard practice. Yet, other statistical methods, which offer complementary benefits, remain underexplored.

For example, my work on *unseen event estimation* [4, 5, 6, 7, 8] addresses data scarcity by estimating missing behaviors and improving test coverage. Similarly, *counterfactual causal analysis*[1, 2, 3, 9] provides explanation-based insights, bridging gaps where traditional machine learning methods fall short. Looking ahead, I aim to integrate advanced statistical approaches—such as *Bayesian modeling*, *extreme value theory*, and *robust inference*—into software engineering. These methods offer untapped potential for addressing scalability, uncertainty, and data sparsity, paving the way for a paradigm shift in how software is analyzed and tested to ensure reliability and efficiency.

The rest of the research statement will elaborate on the **research topics** I have been working on and the **short-term research plans for the next five years**.

1 Counterfactual Program Analysis

When undesired behaviors, such as bugs, arise in software, it is crucial to understand *how* and *why* these behaviors occur. Software operation is a sequence of interdependent instructions, and the software itself is a complex system composed of diverse elements for instructions, such as functions, statements, and variables. Identifying how these elements interact—i.e., *the dependency relations between program elements*—is fundamental to program analysis.

In my research, I have introduced a novel paradigm for dependency analysis, *observation-based dependency analysis* [3, 1, 2], which leverages statistical methods to infer dependencies between program elements. This approach employs *counterfactual reasoning*: if altering the value of program element B causes a change in the value of program element A, it is inferred that program element A depends on program element B.

Observation-based dependency analysis is purely data-driven and, thus, overcomes key limitations of traditional formal semantics-based methods. It avoids reliance on formal semantics, making it effective for features they cannot cover, reducing false positives from over-approximation, and addressing scalability challenges posed by undecidability. By leveraging statistical methods, this approach works even in heterogeneous systems or when parts of the system are unknown or inaccessible. These methods are *data-driven*, *empirically validated*, and provide a *quantifiable* approximation of dependencies, offering practical solutions to an inherently undecidable problem.

Following are the research works I have conducted in this direction.

Identifying the dependency. In our prior work, MOAD [1] successfully approximated program element dependencies using statistical methods by reframing dependency as the likelihood that one program element affects another. We developed an efficient sampling strategy to capture changes in program elements during intervened executions and applied statistical techniques to estimate these probabilities. Our evaluation showed that, for programs analyzable by conventional methods, MOAD achieved high accuracy in identifying elements influencing a target, outperforming traditional approaches. For programs beyond conventional analysis, MOAD uncovered hidden dependencies, such as those mediated through file I/O or network communication, which traditional methods fail to detect.

Quantifying the strength of the dependency. Not all dependencies are equal; A variable's value may exhibit sensitivity to changes in some variables (strong dependency), be rarely affected by others (weak dependency), or remain unresponsive to changes in the rest (no dependency). Reframing the program dependency as a sensitivity to changes can overcome the limitations of the undecidable nature of the program semantics and provide a nuanced understanding of the program operation. In our work, CPDA [2], we utilized causal inference to quantify the dependency strength solely from dynamic information. Our empirical findings revealed that fine-grained dependency information can group program elements into functional clusters, enhancing debugging productivity. Our Subsequent research [9] demonstrated the effectiveness of this information in software testing. Mutating pairs of program elements with strong dependencies are more likely to generate higher-order mutants that are challenging to detect with existing test cases.

2 Reasoning the Unseen in Software Testing

Software testing is fundamentally a *sampling process*, where test cases from the operational distribution—i.e., the distribution of program executions in practice—are executed to uncover defects. Given the vast space of possible test cases, exhaustively covering all program behaviors is infeasible. Consequently, software testing is inherently vulnerable to unseen behaviors, leading to two concrete challenges: 1) the interpretation of test results becomes unreliable, and 2) there is a never-ending concern about the sufficiency of the testing process.

Another research direction I have pursued focuses on developing a statistical inference model to address unseen behaviors in software testing. By quantifying the likelihood of unseen behaviors and incorporating it into test results, the model provides a *reliable interpretation of testing outcomes* [4, 7]. Additionally, it predicts the future performance of the testing process with statistical guarantees, enabling a rational assessment of testing effectiveness and serving as a decision-maker for resource allocation [8, 6]. These advancements enhance the *practicality of software testing.*

The following are the research works I have conducted in this direction.

Extrapolating the greybox fuzzing. Fuzzing, an automated software testing technique generating numerous test cases, is one of the industry's most widely adopted methods. Yet, little is known about how to determine the effectiveness of fuzzing, whether it will uncover new defects, or what its future performance will be. While blackbox fuzzing, with its consistent sampling distribution, is relatively predictable, greybox fuzzing introduces complexities due to adaptive bias, where the sampling distribution evolves based on test input execution coverage. In our work [8], we introduced a novel statistical model for predicting greybox fuzzing performance. Leveraging ecological statistics, our model forecasts future coverage increases in the stochastic process. To address adaptive bias, we partition the coverage record into sub-records, apply ecological statistics to each, and regress predicted coverage increases for extrapolating future performance. Our evaluation shows that this adaptive bias-aware model outperforms existing approaches, offering improved predictions for greybox fuzzing performance.

Reliable information leakage analysis. Information leakage analysis quantifies the information leaked from a secret source to a public sink during program execution. Existing statistical methods rely on mutual information estimation, which is sensitive to missing observations, often leading to either a significant bias or a false sense of security. In our work [7], we developed a novel mutual information estimator that addresses missing observations, enabling accurate and secure leakage estimation even with limited data. Our evaluation shows that our estimator outperforms existing methods in both accuracy and security, enhancing the reliability of information leakage analysis.

Estimating the reaching probability. Quantitative reachability analysis measures the probability of reaching a specific program state, such as an errornous state or a defect-inducing state, during execution. While conventional methods compute this probability through symbolic execution and model counting, we developed the statistical reachability analysis [4] that estimates the reaching probability from the sample program executions. In small-scale programs with known operational distribution, our approach outperformed conventional methods in both accuracy, due to the limited coverage of the formal semantics, and time cost. Our method also demonstrated its effectiveness in large-scale real-world software with an unknown operational distribution, as encountered in software fuzzing.

3 Refining Interdisciplinary Statistical Methods for SE

The software domain has unique characteristics that set it apart from fields where statistical methods are commonly applied, such as ecology, linguistics, and social sciences. Unlike these nature-based environments, the software domain features a discrete and deterministic nature, structural dependencies, and unnatural distributions. Recognizing these distinct traits, we refine statistical methods to improve their applicability in software engineering.

Program-specific characteristics. Programs exhibit a highly structured nature, with dependencies between program elements at the core of their operation. When applying statistical methods to software engineering tasks, we account for this structural aspect, refining the methods for more accurate estimations. For instance, in our reachability analysis [4], we proposed a structure-aware reaching probability estimator. Unlike existing statistical estimators (e.g., Laplace smoothing), which treat all unreached program elements equally, our estimator distinguishes them based on structural dependencies, assigning probabilities accordingly. Similarly, in residual risk analysis [6], we recognized the importance of structural dependency. Residual risk—the risk of a defect remaining after testing—is typically upper-bounded by the discovery probability of uncovered program elements. While existing methods assume independent coverage events, our structure-aware analysis incorporates structural dependencies, i.e., control-flow, providing a significantly tighter upper bound on residual risk.

Recognizing the deterministic nature of software enables more accurate identification of dependencies between program elements. In causal analysis, structure discovery identifies causal relationships from observational data. While existing methods are designed for probabilistic systems, we introduced a novel structure discovery method [2] tailored to deterministic dependencies in software. Specifically, if the value of program element A changes when B is manipulated, A is unequivocally dependent on B. Our method significantly improves causal structure accuracy for program dependency analysis compared to existing approaches.

Distribution-specific estimation. Many statistical methods propose a single general estimator for the unknown quantity of the underlying distribution, irrespective of its shape. While some well-known distributions, like Zipf's law, are common in natural environments, the software domain often exhibits heterogeneous and unnatural distributions. In such cases, a distribution-specific estimator may outperform the general estimator. Our recent work [5] contributes to a purely statistical domain, focusing on estimating the missing probability mass of a distribution. Combining statistical theory with optimization methods, specifically genetic algorithms, we introduce a 'distribution-free methodology' to discover a 'distribution-specific estimator' that surpasses the general estimator in performance.

4 Future Research Directions – Next Five Years Plan

Despite its success, statistical program analysis faces distinct challenges that are fundamentally different from the conventional program analysis. Those challenges includes, but not limited to,

C1. *Making sufficiently precise statements about properties of rarely executed components*: Most of the hard-to-find bugs lie in the rarely executed program components, and missing the behavior of those components can lead to a misprediction of the analysis result, which can be critical in various safety-critical scenarios.

C2. *Efficiently adjusting statistical program analysis in the presence of program evolution*: Re-executing the updated program is required in order to generate the new analysis result, which is time-consuming.

C3. *Adapting the statistical reasoning to the different domain/distribution*: For every empirical research, the domain shift is a critical issue. It occurs when the distribution of the observational data is different from the distribution of the data that the model is trained on, which can lead to a significant bias in the analysis result.

In the short term, I aim to address these challenges through specific research plans, some already in progress.

Modularized statistical program analysis (C1,2,3). I propose a new statistical program analysis that is *modularized* and *compositional* by design. The methodology involves generating observational data for each software component, inferring its statistical reachability model, and composing these models to analyze the entire system. By focusing on individual components, ample data is collected even for rarely observed ones, contributing to the final analysis result. The modularized approach simplifies adaptation to software changes; only updated components require re-analysis, enabling reuse of prior analysis. Additionally, observation and inference for each component can be parallelized, significantly reducing analysis time.

The main challenge is composing statistical models for each component to derive the final analysis for the entire software system. This involves adapting analysis from the independent domain of each component to the context-aware domain of whole program execution—a topic explored in the next research direction. This ongoing research recently secured funding from CASA–Cyber Security in the Age of Large-Scale Adversaries, with me as the sole PI.

Adapting to the different domain (C3). The domain shift is a prevalent issue in the software testing due to the varied execution environments and differences between in-house and production settings. Previously the machine-learning community has developed the domain adaptation method to address the domain shift, which I aim to adapt to the software domain. Two potential approaches are to be explored: 1) composing non-parametric models from each component using a covariate shifting method (e.g., importance sampling) and 2) designing parametric models for each component for the Markov chain model of the reachability.

I propose collaboration with industry partners, highlighting the paramount importance of this issue in practical applications. Many bugs occur in the software, even though it is tested during the development, due to the discrepancy between the development and production environment, and it could cause a significant loss in the industry. The industry's vested interest in overcoming domain shift challenges makes this research particularly appealing, as it aligns with their pressing concerns. Through this collaboration, we aim to validate the effectiveness of proposed solutions in real-world settings, offering practical insights to mitigate domain shift challenges within the industry.

Integration of static and statistical program analysis (C1). Static and dynamic analyses complement each other in program analysis. Static analysis scales effectively in systems with full semantic coverage and provides formal guarantees, while dynamic analysis addresses unknown or inaccessible systems through execution observation with statistical guarantees. Static analysis excels at capturing rare behaviors triggered by specific conditions (e.g., `if (val == 42)`), which are difficult to observe in arbitrary executions. I will explore integrating static and statistical analyses to combine their strengths: using statistical methods to infer the behavior of components not analyzable by static methods, expanding the analysis domain while maintaining both formal and statistical guarantees. The main challenge is identifying suitable components for each method and integrating their guarantees.

References

[1] Lee, S., Binkley, D., Feldt, R., Gold, N., and Yoo, S. Observation-based approximate dependency modeling and its use for program slicing. *Journal of Systems and Software 179* (Sept. 2021), 110988.

[2] Lee, S., Binkley, D., Feldt, R., Gold, N., and Yoo, S. Causal program dependence analysis. *Science of Computer Programming 240* (Feb. 2025), 103208.

[3] Lee, S., Binkley, D., Gold, N., Islam, S., Krinke, J., and Yoo, S. Evaluating lexical approximation of program dependence. *Journal of Systems and Software 160* (Feb. 2020), 110459.

[4] Lee, S., and Böhme, M. Statistical Reachability Analysis. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (New York, NY, USA, Nov. 2023), ESEC/FSE 2023, Association for Computing Machinery, pp. 326–337.

[5] Lee, S., and Böhme, M. How Much is Unseen Depends Chiefly on Information About the Seen, Feb. 2024.

[6] Lee, S., and Böhme, M. Structure-aware Residual Risk Analysis, 2025.

[7] Lee, S., Minocha, S., and Böhme, M. Accounting for Missing Events in Statistical Information Leakage Analysis. In *Proceedings of the IEEE/ACM 47th International Conference on Software Engineering* (New York, NY, USA, 2025), ICSE '25, Association for Computing Machinery, pp. 1–12.

[8] Liyanage, D., Lee, S., Tantithamthavorn, C., and Böhme, M. Extrapolating Coverage Rate in Greybox Fuzzing. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering* (New York, NY, USA, Apr. 2024), ICSE '24, Association for Computing Machinery, pp. 1–12.

[9] Oh, S., Lee, S., and Yoo, S. Effectively Sampling Higher Order Mutants Using Causal Effect. In *2021 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)* (Apr. 2021), pp. 19–24.

"Tell me and I forget. Teach me and I remember. Involve me and I learn." — *Benjamin Franklin*

Over time, I have realized that I am most *motivated to learn*, *retain knowledge*, and *apply it in my life* when I ***understand why it is meaningful***. This insight shapes the central theme of my teaching philosophy: helping students grasp the underlying context of **why the subject matters**, fostering both **self-motivation** and **self-discovery**—guiding them to explore what they truly enjoy and value. I see my role as a mentor on the meta-level of education: *'I am not merely an instructor but a teacher, not a supervisor but an advisor.'* My ultimate goal is to equip students with the skills to learn independently, develop self-motivation, and uncover their passions, empowering them to adapt, grow, and thrive beyond the classroom.

Experience Overview

Classroom Teaching: I have served as a teaching assistant six times for four different courses, including theoretical courses (Logic in CS) and practical courses (Automated or AI-based SE) at the Korea Advanced Institute of Science and Technology (KAIST). These courses ranged from introductory computer science to advanced software engineering. As a teaching assistant, I co-designed course materials, led weekly discussion sections, and graded assignments. In addition, I have delivered guest lectures at the Fuzzing and Software Security Summer School hosted by the National University of Singapore (NUS) and presented several invited talks at various institutions and conferences, both nationally and internationally.

Academic Mentoring: I have mentored three undergraduate students and one Ph.D. student through their research projects, guiding them from defining research questions to writing and submitting research papers, with continuous feedback. All of my mentees have successfully published or submitted their research papers—most as first authors—to prestigious conferences, and many have continued their research careers in academia. As an Open Science Ambassador at the Max Planck Institute for Security and Privacy (MPI-SP), I promote Open Science practices, share resources with colleagues, and participate in annual strategy meetings. As a member of the Early Career Researcher Board at the CASA Graduate School, I help organize activities and address priorities for early career researchers.

Classroom Teaching

I believe what is important for students is not just earning a good grade, but understanding why the subject is meaningful; more importantly, they should genuinely share this belief and feel they have truly achieved it.

This belief stems from two key experiences during my undergraduate years. In a programming language course, I vividly recall the moment before the final exam when I understood not just the features of the minimal programming language we designed, but also why each was necessary, how they connected, and how they were implemented. Building a language from scratch gave me a clear mental map of the process—a turning point that inspired me to approach other subjects similarly and pursue a research career in computer science. In contrast, an engineering course[1] introduced numerous methodologies without context or connections, making them difficult to grasp and less engaging. These experiences shaped my conviction that understanding why concepts are meaningful is crucial for effective learning.

As a faculty member, I aim to help students achieve this understanding by emphasizing connections and context in my teaching. I plan to incorporate **scaffolded learning**, breaking complex topics into manageable steps and gradually reducing support as students gain independence. To help students develop a clear mental framework, I will use **concept mapping**, enabling them to visualize relationships between ideas and understand how individual concepts fit into a broader structure. These approaches encourage students to internalize knowledge, making it easier for them to adapt and apply it in diverse situations throughout their lives. To support this process, I will design my teaching materials to include a broad overview of the subject early on, consistently connect each concept to its surrounding context, and link all ideas back to the overarching framework.

Beyond mastering course content, I want to prepare students for lifelong learning by cultivating critical thinking and adaptability. Through **Socratic seminars**, I aim to create an interactive environment where students engage with open-ended questions, articulate their reasoning, and reflect on their understanding. *University education is just the starting point*—life continually presents challenges that require us to learn anew. By helping students cultivate this mindset, I hope to prepare them not just for exams, but for a

[1]Course name omitted for privacy.

lifetime of learning and growth. My ultimate goal is to empower students to approach new challenges with confidence, apply their knowledge creatively, and thrive in diverse academic and professional contexts.

Academic Mentoring

"... the publishing of research papers does not need to be the goal of a PhD student. Instead, I believe that the most important goal of a PhD student should be to learn how to do research. [1]" — Prof. Yoshi Kohno

Pursuing a Ph.D. is a long and challenging journey, filled with obstacles such as rejections, burnout, self-doubt, and personal crises. I believe the key to overcoming these challenges lies in cultivating the right mindset, understanding the meaning of research, and fostering self-motivation—principles aligned with my teaching philosophy. The ultimate goal of a Ph.D. should not be merely publishing in top-tier conferences, as such outcomes are often influenced by factors beyond the student's control. Instead, **the focus should be on learning to conduct meaningful research and contributing to the field and society**.

This perspective comes from my own experience as a Ph.D. student. One of the final pieces of my dissertation faced multiple rejections during the review process. However, I persevered by adopting a mindset that emphasized the intrinsic value of my research rather than its immediate acceptance. I reminded myself that my work offered unique contributions and that my growth as a researcher was not solely defined by publication success. This mindset helped me stay motivated and resilient, allowing me to continue refining the work until, after years of effort, it was finally published. This experience taught me the value of persistence and the importance of staying committed to meaningful work, even in the face of setbacks.

To support this vision, I employ three core strategies in mentoring:

- **Early Exposure to the Research Cycle**: I encourage students to experience the full research cycle as early as possible, from defining research questions and designing experiments to analyzing results, writing papers, and presenting their work. By engaging with the submission process—including short papers, posters, and doctoral symposiums—and attending conferences, students can gain early insights into what it means to be a researcher. This hands-on experience is invaluable for building motivation and confidence.

- **Frequent Feedback and Communication**: Long gaps between feedback can frustrate students. I prioritize timely, constructive feedback and foster a collaborative environment where students feel supported and empowered to take ownership of their work, emphasizing my role as an advisor rather than a supervisor.

- **Exposure to Diverse Topics**: Exposing students to diverse research areas helps them develop independence and discover their passions. For example, I organized reading groups at the Max Planck Institute, enabling students to explore various topics, engage in discussions, and grow as independent thinkers.

As A. Nico Habermann aptly said: *"Focus on the students, since graduating great students means you'll produce great research, while focusing on the research may or may not produce great students."* This philosophy resonates deeply with me. By prioritizing the growth and well-being of my students, I strive to guide them not only toward academic success but also toward becoming resilient, curious, and capable researchers prepared to tackle the challenges of the future.

References

[1] Kohno, Yoshi. "The Goal of a PhD Program Is Not to Write and Publish Research Papers." Navigating Academia (blog), September 10, 2024. https://medium.com/navigating-academia/the-goal-of-a-phd-program-is-not-to-write-and-publish-research-papers-01e6ace7727d.

Foundations of My Commitment to Diversity, Equity, and Inclusion

Growing up, I often found myself in environments that highlighted the importance of diversity, equity, and inclusion (DEI). Much of my early education took place in settings with significant gender imbalances. I attended a single-sex middle school in South Korea and later enrolled in Ulsan Science High School and the Korea Advanced Institute of Science and Technology (KAIST), both specializing in science and engineering. These environments revealed the systemic challenges faced by women, underscoring the need for deliberate efforts to foster equity. At the same time, I was fortunate to work in research groups that actively prioritized DEI. During my internships and graduate studies, I had advisors who created inclusive spaces—one of whom was a woman, and another who organized annual Ada Lovelace Day events celebrating women in STEM. Participating in these initiatives showed me how fostering inclusivity supports individual well-being and enhances team collaboration and innovation. These experiences have shaped my commitment to DEI in both academic and professional contexts.

Building on these foundations, my understanding of DEI evolved further during my time in Europe. As an Asian male navigating new cultural and academic systems, I experienced firsthand the challenges of being part of a minority group. The Max Planck Institute's international and welcoming community made this transition not only manageable but deeply enriching. Collaborating with colleagues from diverse backgrounds broadened my understanding of DEI, reaffirming its role not only as a moral responsibility but also as a cornerstone of innovation and excellence. However, I recognize that achieving the full benefits of diversity requires deliberate action—such as **promoting diversity through recruitment** and **fostering inclusive environments**.

Promoting Diversity through Recruitment

My understanding of DEI deepened through leadership roles in extracurricular activities. As the club master of the undergraduate Classical Guitar Club and a managing board member of the graduate Badminton Club—both of which were among the larger clubs at my institutes, with 30 to 50 active members—I prioritized fostering diversity during recruitment. While inclusiveness and a welcoming attitude were central values, I recognized that achieving diversity within a team is often the first step toward building an inclusive community. For individuals from minority groups, having peers who genuinely understand and share their experiences is especially important.

At KAIST, I worked to encourage a balanced and diverse membership in both clubs, achieving a roughly 50% gender balance. This approach not only promoted diversity but also enabled us to create a more supportive environment. For example, we organized small peer support groups for female members, providing spaces for mutual understanding, support, and joy in both club activities and broader academic life. These initiatives fostered active participation from all members and ensured continuity of these values—successive boards included women who carried forward the club's commitment to diversity, equity, and inclusion.

These experiences taught me the importance of thoughtful recruitment as a means to promote diversity, as well as the sustained support needed to build inclusive communities. As a faculty member, I will apply these lessons in recruiting diverse teams, mentoring students, and fostering a culture where everyone feels valued and empowered. By aligning these efforts with the institution's broader commitment to excellence and inclusion, I aim to contribute to an environment where DEI principles are not just ideals but lived experiences.

Fostering Inclusive Environments

Creating welcoming environments where individuals from all backgrounds feel supported has been a key focus of my efforts. Currently, I serve on the Early Career Researchers (ECR) Board in the Cluster of Excellence CASA, where we oversee funding allocations and organize events such as summer schools, symposiums, and regular meetings. Addressing inclusiveness is central to our work, and we regularly incorporate feedback to ensure our programs meet the needs of a diverse community. For instance, I proposed "speed-dating" activities to connect German-speaking and non-German-speaking members, fostering engagement and helping international researchers feel more integrated into the community.

I also take personal responsibility for creating a welcoming atmosphere. As an international researcher, I understand the importance of having someone reach out during transitions into a new community. I make a point to welcome newcomers—whether in my research group or elsewhere—by initiating conversations, encouraging participation in activities, and fostering a friendly environment. One intern remarked, “Seongmin’s welcoming attitude made a big difference. He was the first to reach out, and he created an atmosphere where everyone felt comfortable and included.”

My earlier leadership experiences in the Classical Guitar and Badminton Clubs also shaped my approach to inclusiveness. I organized small, diverse support groups pairing seniors with juniors to foster mentorship and camaraderie, helping members navigate both club activities and broader academic life. Drawing from my experiences of being both in the majority and minority, I understand how inclusiveness enhances individual well-being and strengthens community cohesion.

I will promote diversity through recruitment within my research group and foster welcoming environments across my research group, department, and institute. By thoughtfully recruiting diverse teams and establishing support systems that ensure individuals feel valued and empowered, I will cultivate a culture where collaboration and innovation thrive. These approaches will enhance individual well-being while contributing to stronger, more inclusive academic communities that reflect the values of diversity, equity, and inclusion.

List of References

[1] **Dr. Shin Yoo**
Tenured Associate Professor
School of Computing
Korea Advanced Institute of Science and Technology
Address: 291 Daehak Ro, Yuseong Gu, Daejeon 34141, Republic of Korea
Email: shin.yoo@kaist.ac.kr

[2] **Dr. Marcel Böhme**
Faculty Member
Max Planck Institute for Security and Privacy (MPI-SP)
Address: 140 Universitätsstraße, Bochum, Nordrhein-Westfalen 44799, Germany
Email: marcel.boehme@acm.org

[3] **Dr. David W. Binkley**
Professor of Computer Science
Computer Science Department
Loyola University Maryland
Address: 4501 North Charles Street, Baltimore, MD 21210-2699, USA
Email: binkley@cs.loyola.edu