

# Saikat Dutta | Curriculum Vitae

Urbana, IL, USA

+1 217-721-7567 • saikatd2@illinois.edu  
saikatdutta.web.illinois.edu • Saikat Dutta

## Research Interests

---

My research interests lie broadly at the intersection of **software engineering**, **program analysis**, and **machine learning**. The goal of my research is to develop intelligent and cost-effective techniques to improve reliability of **Machine Learning**-based systems.

## Education

---

- **University of Illinois at Urbana-Champaign (UIUC)** **Illinois, USA**  
*PhD, Computer Science, Programming Languages and Software Engineering Group* *2017–present*  
**Advisor:** Prof. Sasa Misailovic
- **Jadavpur University** **Kolkata, India**  
*Bachelor of Computer Science and Engineering* *2011–2015*

## Honors

---

- Awarded **Mavis Future Faculty Fellowship**, College of Engineering, UIUC, 2022-23.
- Selected to represent UIUC for **Google PhD Fellowship 2022**.
- Awarded **Facebook PhD Fellowship 2020-22**: \$42,000 annual stipend for 2 years.
- Selected to represent UIUC for **Microsoft PhD Fellowship 2020**.
- Awarded **3M Foundation Fellowship 2018-19**: \$10,000 fellowship award for 1 year.
- **Facebook PhD Fellowship 2019** Finalist.

## Publications

---

*13 full conference papers (12 technical track, 1 industry track), 2 journal papers. 10 first-author papers.*  
*Full papers available on my website: saikatdutta.web.illinois.edu*

15. **S. Dutta**, D. Garbervetsky, S. Lahiri, and M. Shaefer. InspectJS: Leveraging Code Similarity and User-Feedback for Effective Taint Specification Inference for JavaScript. *44th International Conference on Software Engineering - Software Engineering in Practice (ICSE-SEIP) 2022*.
14. **S. Dutta**, A. Arunachalam, and S. Misailovic. To Seed or Not to Seed? An Empirical Analysis of Usage of Seeds for Testing in Machine Learning Projects. *15th IEEE International Conference on Software Testing, Verification and Validation (ICST) 2022*.
13. **S. Dutta**, Z. Huang, and S. Misailovic. SixthSense: Learning to Debug Convergence Problems in Probabilistic Programs. *25th International Conference on Fundamental Approaches to Software Engineering (FASE) 2022*.
12. Z. Huang, **S. Dutta**, and S. Misailovic. Automated Quantized Inference for Probabilistic Programs with AQUA. *Innovations in Systems and Software Engineering: A NASA Journal (ISSE NASA) 2022*.
11. Z. Huang, **S. Dutta**, and S. Misailovic. AQUA: Automated Quantized Inference for Probabilistic Programs. *19th International Symposium on Automated Technology for Verification and Analysis (ATVA) 2021*.
10. **S. Dutta**, J. Selvam, A. Jain, and S. Misailovic. TERA: Optimizing stochastic tests in Machine Learning Projects. *30th ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA) 2021*.

9. **S. Dutta**, A. Shi, and S. Misailovic. FLEX: Fixing Flaky Tests in Machine Learning Projects by Updating Assertion Bounds. *29th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE) 2021*.
8. **S. Dutta**, A. Shi, R. Choudhary, Z. Zhang, A. Jain, and S. Misailovic. Detecting Flaky Tests in Probabilistic and Machine Learning Applications. *29th ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA) 2020*.
7. **S. Dutta**, W. Zhang, Z. Huang, and S. Misailovic. Storm: Program Reduction for Testing and Debugging Probabilistic Programming Systems. *27th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE) 2019*.
6. **S. Dutta**, O. Legunsen, Z. Huang, and S. Misailovic. Testing Probabilistic Programming Systems. *26th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE) 2018*.
5. B. Nongpoh, R. Ray, **S. Dutta**, and A. Banerjee. Autosense: A Framework for Automated Sensitivity Analysis of Program Data. *IEEE Transactions on Software Engineering (TSE) 2017*. Invited for presentation at *ESEC/FSE 2017*.
4. S. Chattopadhyay, **S. Dutta**, and A. Banerjee. A Framework For Fast Service Verification and Query Execution for Boolean Service Rules. *In 9th Asia-Pacific Services Computing Conference (APSCC) 2015*.
3. **S. Dutta**, S. Chattopadhyay, A. Banerjee, and P. Dasgupta. A New Approach For Minimal Environment Construction for Modular Property Verification. *In 24th IEEE Asian Test Symposium, (ATS) 2015*.
2. **S. Dutta**, M. Das, and A. Banerjee. Enhancing Branch Prediction Using Software Evolution. *In 10th IEEE International Conference on Networking, Architecture and Storage (NAS) 2015*.
1. N. Jain, **S. Dutta**, A. Banerjee, A. K. Ghosh, L. Xu, and H. Zhu. Using Daikon to Prioritize and Group Unit Bugs. *In Formal Aspects of Component Software - 10th International Symposium, (FACS) 2013*.

## Experience

---

- Amazon Web Services** **Seattle, USA**  
*Applied Research Intern, Automated Reasoning Group* *May 2021–August 2021*  
 During my internship, I worked on developing automated techniques for testing Deep Learning Compilers.  
**Manager:** Daniel Kroening, **Mentor:** Willem Visser
- Microsoft Research** **Redmond, USA**  
*Research Intern, RISE* *May 2020–August 2020*  
 During my internship with the RISE team, I worked on harnessing program analysis, big code, and machine learning techniques to automate environment modeling to significantly boost static analyzers for security and reliability.  
**Manager:** Shuvendu Lahiri, **Mentor:** Madan Musuvathi
- Microsoft India Development Centre** **Hyderabad, India**  
*Software Engineer, Bing Division* *July 2015–July 2017*  
 I worked with the Bing and Cortana team in Hyderabad. I developed key features for Bing and Cortana and contributed towards maintenance of health and availability of its services.
- Microsoft India Development Centre** **Hyderabad, India**  
*Software Development Engineer - Intern, ICE Division* *May 2014–July 2014*
- IIT Kharagapur** **West Bengal, India**  
*Research Intern, Software Engineering* *December 2014 – January 2015*
- Indian Statistical Institute** **West Bengal, India**  
*Research Intern, Software Engineering* *December 2013 – January 2014*
- Indian Statistical Institute** **West Bengal, India**  
*Research Intern, Software Engineering* *May 2013–July 2013*

## Research Mentoring

---

I have mentored the research and collaborated with two PhD, one masters, and 15 undergraduate students (including four women undergraduates):

- Steven Xia (PhD, UIUC; Co-authored [paper under submission])
- Rutvik Choudhary (PhD, UIUC; Co-authored [8])
- Peilun Zhang (MS, UIUC)
- Rem Yang (BS, UIUC)
- Süleyman Ateş (BS, Middle East Technical University, Turkey)
- Selim Kuzuku (BS, Middle East Technical University)
- Muhammet Emin Cihangeri (BS, Middle East Technical University)
- Furkan Genç (BS, Middle East Technical University)
- Steven Pan (BS, UIUC)
- Ankitha Damisetty (BS, UIUC)
- Sanjana Sarkar (BS, UIUC)
- Anshul Arunachalam (BS, UIUC; Co-authored [14])
- Jeeva Selvam (BS, UIUC; Co-authored [10])
- Enguang Fan (BS, UIUC)
- Aryaman Jain (BS, UIUC; Co-authored [10,8])
- Zhekun Zhang (BS, UIUC; Co-authored [8])
- Wenxian Zhang (BS, UIUC; Co-authored [7])
- Zixin Huang (BS, UIUC; Co-authored [6])

## Teaching Experience

---

- Teaching Assistant for CS 526 (Advanced Compiler Construction), UIUC, Spring 2020

## Open-Source Contributions

---

- *ProbFuzz*: ProbFuzz is a fuzzing-based tool for detecting bugs in Probabilistic Programming Systems. Till date, ProbFuzz has revealed more than 50 bugs in three popular probabilistic programming systems and their underlying frameworks. ProbFuzz is available at <https://www.probfuzz.com>. We also curated a dataset of bugs in probabilistic programming systems. It is available at <https://github.com/uiuc-arc/probfuzz-db>.
- *Storm*: Storm is a tool for automatically reducing fault-exposing programs and data for probabilistic programming systems. For Stan, Storm's reduced programs had 49% less code, 67% less data, and 96% fewer inference iterations. For Pyro, Storm's reduced programs had 58% less code, 96% less data, and 99% fewer inference iterations. Overall, the reduced programs run up to 126x faster. Storm is available at <https://github.com/uiuc-arc/Storm>.
- *Storm-Framework*: Storm-Framework is a unified framework for testing, analysis, and transformation of probabilistic programs. At its core, Storm consists of four main components: the Storm-IR language, a Translator, a Transformer, and a Static Analysis engine. The key advantage of Storm-Framework is the common intermediate language, which allows the programmer to write any kind of analysis/transformation once, and then translate it into any PPL of their choice. This rids the programmer from having to deal with the intricacies of individual PPSs. Storm-Framework is available at <https://github.com/uiuc-arc/storm-framework>.
- *FLASH*: FLASH is a tool for automatically detecting flaky tests caused due to different sequences of random numbers produced in each execution, which is common in Machine Learning libraries that implement stochastic algorithms. FLASH helped discover 11 new flaky tests in Machine Learning libraries. FLASH is available at <https://github.com/uiuc-arc/flash>.
- *FLEX*: FLEX is a tool for automatically fixing flaky tests caused due to randomness of stochastic algorithms in Machine Learning libraries. FLEX helped fix 28 flaky tests in Machine Learning libraries. FLEX is available at <https://github.com/uiuc-arc/flex>.
- *TERA*: TERA is a tool for reducing the execution time of stochastic regression tests in Machine Learning libraries. TERA obtains a speed-up of 2.23x over 160 tests collected from 15 Machine Learning libraries. TERA is available at <https://github.com/uiuc-arc/tera>.

- **XSEED**: XSEED is a tool for automatically running and comparing tests with and without seeds. XSEED allows users to study the impact of setting seeds on their test execution. XSEED is available at <https://github.com/uiuc-arc/xseed>.
- **AQUA**: AQUA is a tool for performing Bayesian inference for probabilistic programs. AQUA uses an efficient quantization technique that scales better than purely symbolic techniques and is more accurate than approximate techniques. AQUA is available at <https://github.com/uiuc-arc/aqua>.
- **SixthSense**: SixthSense is a learning-based approach for predicting convergence of probabilistic programs. Given a probabilistic program, SixthSense can statically predict whether it will converge for a given inference algorithm and indicate likely program features that contribute to non-convergence. SixthSense is available at <https://github.com/uiuc-arc/sixthsense>.

## Talks

---

### Conference Talks and Posters

- **Conference Talk**: Presented my paper: *InspectJS: Leveraging Code Similarity and User- Feedback for Effective Taint Specification Inference for JavaScript* at **ICSE 2022** (Virtual)
- **Conference Talk**: Presented my paper: *To Seed or Not to Seed? An Empirical Analysis of Usage of Seeds for Testing in Machine Learning Projects* at **ICST 2022** (Virtual)
- **Conference Talk**: Presented my paper on *Flex: Fixing Flaky Tests in Machine-Learning Projects by Updating Assertion Bounds* at **FSE, 2021** (Virtual)
- **Conference Talk**: Presented my paper on *TERA: Optimizing Stochastic Regression Tests in Machine Learning Projects* at **ISSTA, 2021** (Virtual)
- **Conference Talk**: Presented my paper on *Detecting Flaky Tests in Probabilistic and Machine Learning Applications* at **ISSTA, 2020** (Virtual)
- **Invited Talk**: Presented my work at the Midwest Programming Languages Summit, 2019 in Purdue University, West Lafayette
- **Conference Talk**: Presented my paper on *Storm: Program Reduction for Testing and Debugging Probabilistic Programming Systems* at **FSE, 2019** (Tallinn, Estonia)
- **Conference Talk**: Presented my paper on *Testing Probabilistic Programming Systems* at **FSE, 2018** (Lake Buena Vista, Orlando)
- **Invited Talk**: Presented my work in Midwest Programming Languages Summit, 2018 at University of Wisconsin-Madison
- **Poster Presentation** at 1st Conference on Probabilistic Programming: **ProbProg, 2018** (Boston)
- **Conference Talk**: Presented my paper on *Minimal environment construction for modular property verification* at **ATS, 2015** (Mumbai, India)

### Guest Lectures

- **Guest Lecture**: TERA: Optimizing Stochastic Regression Tests in Machine Learning Projects, Flex: Fixing Flaky Tests in Machine-Learning Projects by Updating Assertion Bounds, CS 521 (Topics in Programming Languages: Approximate And Probabilistic Programming Systems), UIUC, Spring 2022
- **Guest Lecture**: *Detecting and Fixing Flaky Tests in Machine Learning Projects*, CS 527 (Topics in Software Engineering), UIUC, Fall 2021
- **Guest Lecture**: *Detecting Flaky Tests in Probabilistic and Machine Learning Applications*, CS 598sm (Approximate and probabilistic computing across the system stack), UIUC, Fall 2020
- **Guest Lecture**: *Gen: a general-purpose probabilistic programming system with programmable inference*, CS 598sm (Approximate and probabilistic computing across the system stack), UIUC, Fall 2020
- **Guest Lecture**: *Control Flow Analysis*, CS 526 (Advanced Compiler Construction), Spring 2020
- **Guest Lecture**: *Dependence Analysis*, CS 526 (Advanced Compiler Construction), Spring 2020

## Awards

---

- SIGSOFT CAPS Travel Grant for ICSE 2022.
- SIGSOFT CAPS Travel Grant for ESEC/FSE 2019.
- SIGSOFT CAPS Travel Grant for ESEC/FSE 2018.
- Travel Grant for the Midwest Programming Languages Summit, 2018.

- PLMW scholarship for POPL, 2015.

## Service

---

- Reviewer, TSE 2022
- Reviewer, MSR 2022, Shadow PC Track
- Reviewer, PLDI 2021 Artifact Evaluation Track
- Reviewer, OOPSLA 2020 Artifact Evaluation Track
- Mentor, Undergraduate Research Apprenticeship Program 2020