Ting Su

Homepage: http://tingsu.github.io/ **Email**: tsuletgo@gmail.com
Office: CNB H101.1, Universittstrasse 6, 8092 Zrich, Switzerland **Tel**: (+86)13501965682

RESEARCH INTERESTS

My research interests mainly involve *software engineering*, *programming language*, *software security* and *artificial intelligence*. I am particularly interested in developing novel program analysis, testing and verification techniques to improve reliability, security, and development productivity for different software systems.

EDUCATION

East China Normal University, China **Sep 2011 - July 2016**
Ph.D. in Computer Science
Dissertation: *Automated Coverage Criteria-based Test Data Generation: Approaches and Implementations*
Advisor: Prof. Jifeng He and Prof. Geguang Pu

University of California, Davis, USA **Feb 2014 - Feb 2015**
Visisting Ph.D. student
Advisor: Prof. Zhendong Su

East China Normal University, China **Sep 2007 - July 2011**
B.S. in Software Engineering

EMPLOYMENT

ETH Zurich, Switzerland **June 2019 - Now**
Postdoctoral Scholar
Advisor: Prof. Zhendong Su

Nanyang Technological University, Singapore **Aug 2016 - May 2019**
Postdoctoral Research Fellow
Advisor: Prof. Yang Liu

Rolls-Royce Research Lab, Singapore **Dec 2016 - Oct 2017**
Research Scientist

Synopsys Shanghai Co. Ltd, China **July 2015 - Sep 2015**
Software Engineer

HONORS AND AWARDS

ACM SIGSOFT Distinguished Paper Award (ASE 2019, CCF-A), ACM, 2019
(*one of 6 selected papers from total 409 submissions*)
ACM SIGSOFT Distinguished Paper Award (ICSE 2018, CCF-A), ACM, 2018
(*one of 8 selected papers from total 502 submissions*)
ACM SIGSOFT Distinguished Paper Award (ASE 2018, CCF-A), ACM, 2018
(*one of 6 selected papers from total 346 submissions*)
Nomination for Distinguished PhD Thesis, China Computer Federation (CCF), 2017
(*only 20 candidates were selected as finalist nationwide*)
First Place of Research Prototype Tool Competition (NASAC 2017), CCF, 2017
(*NASAC is the flagship annual software engineering and system conference in China*)
Third Place of Research Prototype Tool Competition (NASAC 2018), CCF, 2018
Golden Medal of ACM Student Research Competition (ICSE 2016, CCF-A), ACM, 2016
ACM SIGSOFT CAPS Award (ICSE 2015), ACM, 2015
National Scholarship for Doctorate, China, 2014

RESEARCH GRANTS

- NTUitive, NTUitive GAP Fund, "Cloud-based Mobile App Testing Service", Jan 2018-June 2019, NGF-2017-030033, S$ 240,250. **Co-PI** (with PI: Prof. Yang Liu from NTU, Singapore). **Finished**.
- NTU SCSE, Research Demo Fund, "Distributed and Automated Mobile App Testing Platform", Jan 2018-June 2018, S$ 50,000. **Co-PI** (with PI: Prof. Yang Liu,

Singapore). **Finished**.

- Continental, Continental Innovation Project, "Investigation of AI Techniques for Software Engineering", Jan 2019-May 2019, S$ 75,000. **Co-PI** (with PI: Prof. Shang-Wei Lin and Co-PI: Yang Liu from NTU, Singapore). **Finished**.
- Singapore NSF, "Enhancing Cyber Resilience of Deep Learning Models Against Adversarial Cyber Attacks", 2019 - 2022, S$ 3,849,762. Participate. **Onging**.
- China NSF, "Requirement Analysis for Data-driven Machine Learning System", 2019 - 2022, RMB 740,000. Participate. **Onging**.
- Rolls-Royce Research Lab, "Process Simulation Description Language", 2014 - 2017, S$ 403,726. Participate. **Finished**.

PROJECTS

Mobile Software Analysis, Testing and Security **2015 - now**

I lead several research work on analyzing, testing, debugging, and securing mobile applications. The *key goal* is to sharpen the competitive edge of mobile apps, and ensure their reliability and security. In particular, I spent two years and developed a state-of-the-art GUI testing tool ***Stoat*** (`https://github.com/tingsu/stoat`) for Android apps. Stoat has been deployed to extensively test 9,000+ open-source and commercial apps and found 8,000+ fatal crashes in one year. Stoat has contributed to such popular apps as `WeChat`, `Google+`, `Gmail` that have billions of users. Stoat won the *Golden Medal of ACM Student Research Competition* in ICSE 2016, *First Place of Research Tool Competition* in NASAC 2017, and received *NTUitive Gap Fund Grant* S$240,250 in 2018 for commercialization potential. Stoat was recognized as "*a state-of-the-art tool, (which) is able to trigger the highest number of unique crashes (in Android apps)*" by Prof. Tao Xie (IEEE Fellow) in their ASE'18 paper; "*Stoat is effective for discovering unique crashes*" by Prof. Andreas Zeller (ACM Fellow) in their ISSTA'19 paper. Stoat has also been widely used by the researchers from many renowned institutes worldwide, e.g., UIUC, UC Irvine, NUS, ETH Zurich, Peking University, Nanjing University. See the demo of Stoat testing platform: `https://youtu.be/41jzFM7WhP4`

I also lead the first, largest and most comprehensive fault study for Android apps (`https://crashanalysis.github.io/`), based on the bugs detected by Stoat and the data from open-source hosting platforms (e.g., `Github`, `Google Code`). This study involves 2,486 open-source apps, 3,230 commercial apps, and 135 professional app developers worldwide. We investigate the key question "*why an Android app may crash?*", understand how app developers analyze, test, reproduce and fix app crashes, and motivate several follow-up research directions. This work won an *ACM SIGSOFT Distinguished Paper Award* in ICSE 2018 with **three strong acceptance**. Our dataset and benchmark have been requested by more than 13 institutes worldwide (e.g., CMU, UNSW, Monash, HKUST). Following this study, we designed more effective testing techniques (Stoat+ and APEChecker) for detecting intricate bug types (e.g., asynchronous programming errors).

We also conducted a large-scale automated security risk assessment for mobile banking apps. This study revealed 2,157 security weaknesses in 693 banking apps worldwide. To date, 21 banking entities have confirmed 126 weaknesses, and 52 weaknesses have been patched. This work has been well-acknowledged by some bank entities from HonKong, Singapore and India, e.g., `HSBC`, `OCBC`, `DBS` and `BHIM`. The tool **Ausera** won *Third Place of Research Tool Competition* in NASAC 2018, and now has been integrated into one security scanning product of `Scantist`.

Testing and Analyzing Embedded Software **2014 - now**

Since my Ph.D., I continuously lead the research work on leveraging *dynamic symbolic execution* techniques to automatically generate test data for industrial embedded software. The *key goal* is to reduce human testing efforts and detect functional bugs as early as possible (especially at the unit testing stage). I developed and maintained an academic research tool **CAUT** that automatically generates test data to satisfy different testing adequacy criteria. Later, as a co-founder, I significantly contributed to

SmartRocket Unit (`https://www.ticpsh.com/air.html`), a commercial automated unit testing tool inspired by CAUT. SmartUnit fully supports unit testing for statement, branch, boundary value and MC/DC coverage, and has been applied in several industry-scale embedded control systems, and accumulatively tested 3 millions lines of C code. It is now serving more than ten Chinese industrial companies, including `CASCO` (the best railway signal corporation in China), `GAC MOTOR` (one of biggest car manufacturers in China), `CAST` (the main spacecraft production agency in China). It recently received the German PUV Rheinland security certificate.

Testing Compilers and Program Analyzers **2015 - now**
I also worked on testing compilers (e.g., Java Virtual Machines) and program analyzers (e.g., software model checkers, SAT/SMT solvers). The *key goal* is to ensure the correctness and reliability of many fundamental development tools we very depend on. We proposed two stress-testing techniques ***classming*** and ***classfuzz*** for JVMs. We discovered 62+ JVM discrepancies in the start-up process of `Oracle`'s HotSpot and `IBM`'s J9, which lead to several clarifications and changes to the Java SE 8 edition of the JVM specification; We revealed 30+ fatal bugs in `Oracle`'s HotSpot and `IBM`'s J9. In particular, we found one highly-critical security vulnerability (CVE-2017-1376, CVSS base score 9.8), which directly affects 6 IBM's products and 99 derivative products.

To validate the correctness of software model checkers (the implementations of a well-known automatic software verification technique proposed by Turing Award Winner E. M. Clarke et al.), we proposed an effective testing technique **MCFuzz** (`https://github.com/MCFuzzer/MCFuzz`), which is now the only available, effective approach in the world to test software model checkers. It has extensively tested three state-of-the-art and state-of-the-practice C software model checkers, `CPAchecker`, `CBMC`, and `SeaHorn`. We have found bugs in all three model checkers. To date, we have reported a total of 62 unique bugs — 59 were previously unknown, 58 have been confirmed, and 20 have been fixed. We are well-appreciated by the model checker developers and our tests have been integrated as their standard benchmarks.

Software Engineering for AI and AI for Software Engineering **2018 - now**
I am also looking into how to effectively test, secure and understand deep learning systems, whose correctness is very important in safety-critical domains like self-driving and medical diagnosis. We have designed a set of testing criteria for gauging the test adequacy of deep neural network. On the other hand, we have used deep learning techniques to automate GUI design and test gaming apps.

Business Process Model Mining and Validation **2017 - 2018**
At the Rolls-Royce research lab, I worked on developing an automated business process model translation and mining tool chain to validate the correctness of in-house business process, especially for the airplane engine development process.

Setting Exploration Automation **2015**
At Synopsys, I worked on developing a distributed testing framework to efficiently generate optimal parameter configurations for Synopsys's in-house hardware verification tools. This project was recognized as *an innovation project* in Synopsys.

PUBLICATIONS

Summary

- Note that the tradition of publication in computer science, and my field in particular, is mainly to publish full papers in peer-reviewed conference proceedings. ICSE, FSE, ASE, PLDI, and OOPSLA are fiercely competitive and widely accepted as the top-tier conferences in my research area. Top-tier conferences have a typical acceptance rate around 20%. TSE, TOSEM, CSUR are top-tier journals in my research area.
- In each publication, the label "*****" denotes I am the corresponding and the main author, "#" denotes I am the co-first author and make equal contribution with the first author, "**S**" denotes the students I directly supervised when the work was conducted. ***Without any specific annotations, each paper is full research paper***.

- Basically, for all papers, I have made significant contributions to the idea, research, discussions and paper writing. In particular, I am the corresponding and the main author of the following **11** *top-tier* published papers: [**FSE-19**], [**ICSE-19a**], [**ICSE-18a**], [**ICSE-18b**], [**ICSE-18c**], [**ASE-18**], [**FSE-18**], [**FSE-17**], [**CSUR-17**], [**ICSE-16**], [**ICSE-15**].

Public Profiles

- Google Scholar: `https://scholar.google.com/citations?user=GHPWI7oAAAAJ`
- DBLP: `https://dblp.uni-trier.de/pers/hd/s/Su:Ting`
- Personal Page: `http://tingsu.github.io/`

Highlight: Total number of published papers (29), among them **16** are top-tier papers (CCF-A/SCI Rank-1): ICSE(8), FSE(3), ASE(3), CSUR(1), PLDI(1)

Peer-reviewed Journal Papers

1. [**TSE-20**] **Ting Su**, Lingling Fan, Sen Chen, Yang Liu, Lihua Xu, Geguang Pu, and Zhendong Su. Why My App Crashes? Understanding and Benchmarking Framework-specific Exceptions of Android apps. IEEE Transactions on Software Engineering. TSE 2020. (IF: 3.33, **CCF-A**), *Minor Revision*.
2. [**CSUR-17**] **Ting Su**, Ke Wu, Weikai Miao, Geguang Pu, Jifeng He, Yuting Chen, Zhendong Su. A Survey on Data-Flow Testing. ACM Computing Surveys, 2017 (IF: 6.748. **SCI Rank 1**)
3. [**SCIS-16**] **Ting Su**, Geguang Pu, Weikai Miao, Jifeng He, Zhendong Su. Automated Coverage-driven Testing: Combining Symbolic Execution and Model Checking. SCIENCE CHINA Information Sciences, 2016 (IF: 1.628, CCF-B) (*Invited Special Session Paper*)
4. [**IJCM-13**] Yongxin Zhao, Hao Xiao, Zheng Wang, Geguang Pu, **Ting Su**. The Semantics and Verification of Timed Service Choreography. International Journal of Computer Mathematics. IJCM 2013

Peer-reviewed Conference Papers

1. [**FSE-17**] **Ting Su**, Guozhu Meng, Yuting Chen, Ke Wu, Weiming Yang, Yao Yao, Geguang Pu, Yang Liu, Zhendong Su. Guided, Stochastic Model-Based GUI Testing of Android Apps. The 11th joint meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering. FSE 2017. (**CCF-A**)

   First Place of Research Prototype Tool Competition in NASAC 2017.
2. [**ICSE-16**] **Ting Su**. FSMdroid: Guided GUI Testing of Android Apps. The 38th International Conference on Software Engineering. ICSE 2016. (**CCF-A**, Student Research Competition Paper)

   Golden Medal of ACM Student Research Competition.
3. [**ICSE-15**] **Ting Su**, Zhoulai Fu, Geguang Pu, Jifeng He, Zhendong Su. Combining Symbolic Execution and Model Checking for Data Flow Testing. 37th IEEE/ACM International Conference on Software Engineering. ICSE 2015. (**CCF-A**)
4. [**QRS-14**] **Ting Su**, Siyuan Jiang, Geguang Pu, Bin Fang, Jifeng He, Jun Yan, Jianjun Zhao. Automated Coverage-Driven Test Data Generation Using Dynamic Symbolic Execution. Eighth International Conference on Software Security and Reliability. SERE 2014. (now renamed as QRS, CCF-C)
5. [**FSE-19**] Chengyu Zhang$^{s\#}$, **Ting Su**$^{\#*}$, Yichen Yan, Fuyuan Zhang, Geguang Pu and Zhendong Su. Finding and Understanding Bugs in Software Model Checkers. The 27th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering. FSE 2019. (**CCF-A**)
6. [**ICSE-18a**] Lingling Fan$^{s\#}$, **Ting Su**$^{\#*}$, Sen Chen, Guozhu Meng, Yang Liu, Lihua Xu, Geguang Pu and Zhendong Su. Large-Scale Analysis of Framework-Specific

Exceptions in Android Apps. The 40th International Conference on Software Engineering. ICSE 2018. (**CCF-A**)

ACM SIGSOFT Distinguished Paper Award

7. [**ICSE-19a**] Yuting Chen, **Ting Su***, Zhendong Su. Deep Differential Testing of JVM Implementations. The 41st ACM/IEEE International Conference on Software Engineering. ICSE 2019. (**CCF-A**)

8. [**ICSE-18b**] Chunyang Chen**S**, **Ting Su***, Guozhu Meng, Zhenchang Xing and Yang Liu. From UI Design Image to GUI Skeleton: A Neural Machine Translator to Bootstrap Mobile GUI Implementation. The 40th International Conference on Software Engineering. ICSE 2018. (**CCF-A**)

9. [**ASE-18a**] Lingling Fan**S**, **Ting Su***, Sen Chen, Guozhu Meng, Yang Liu, Lihua Xu, Geguang Pu. Efficiently Manifesting Asynchronous Programming Errors in Android Apps. The 33rd IEEE/ACM International Conference on Automated Software Engineering. ASE 2018. (**CCF-A**)

10. [**FSE-18**] Sen Chen**S**, **Ting Su***, Lingling Fan, Guozhu Meng, Minhui Xue, Yang Liu, Lihua Xu. Are Mobile Banking Apps Secure? What Can be Improved? The 26th joint meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering. FSE 2018. (**CCF-A**) (Full Industry Paper)

11. [**PLDI-16**] Yuting Chen, **Ting Su**, Chengnian Sun, Zhendong Su, Jianjun Zhao. Coverage-Directed Differential Testing of JVM Implementations. ACM SIGPLAN Conference on Programming Language Design and Implementation. PLDI 2016. (**CCF-A**)

12. [**ICSE-18c**] Chengyu Zhang**S**, Yichen Yan**S**, Hanru Zhou, Yinbo Yao, Ke Wu, **Ting Su***, Weikai Miao and Geguang Pu. SmartUnit: Empirical Evaluations for Automated Unit Testing of Embedded Software in Industry. The 40th International Conference on Software Engineering. ICSE 2018. (**CCF-A**) (Full Industry Paper)

13. [**ASE-19a**] Yan Zheng, Xiaofei Xie, **Ting Su**, Lei Ma, Jianye Hao, Zhaopeng Meng, Yang Liu, Ruimin Shen, Yinfeng Chen, and Changjie Fan. Wuji: Automatic Online Combat Game Testing Using Evolutionary Deep Reinforcement Learning. The 34th IEEE/ACM International Conference on Automated Software Engineering. ASE 2019. (**CCF-A**).

    ACM SIGSOFT Distinguished Paper Award

14. [**ICSE-19b**] Yu Zhao, Tingting Yu, **Ting Su**, Yang Liu, Wei Zheng, Jingzhi Zhang, William G.J. Halfond. ReCDroid: Automatically Reproducing Android Application Crashes from Bug Reports. The 41st ACM/IEEE International Conference on Software Engineering. ICSE 2019. (**CCF-A**)

15. [**ICSE-19c**] Sen Chen, Lingling Fan, Chunyang Chen, **Ting Su**, Wenhe Li, Yang Liu, Lihua Xu. StoryDroid: Automated Generation of Storyboard for Android Apps. The 41st ACM/IEEE International Conference on Software Engineering. ICSE 2019. (**CCF-A**)

16. [**TASE-19**] Li Hao, Jianqi Shi, **Ting Su** and Yanhong Huang. Automated Test Generation for IEC 61131-3 ST Programs via Dynamic Symbolic Execution. The 13th International Symposium on Theoretical Aspects of Software Engineering. TASE 2019. (CCF-C)

17. [**ASE-18b**] Lei Ma, Felix Juefei-Xu, Fuyuan Zhang, Jiyuan Sun, Minhui Xue, Bo Li, Chunyang Chen, **Ting Su**, Li Li, Yang Liu, Jianjun Zhao, and Yadong Wang. DeepGauge: Multi-Granularity Testing Criteria for Deep Learning Systems. The 33rd IEEE/ACM International Conference on Automated Software Engineering. ASE 2018. (**CCF-A**)

    ACM SIGSOFT Distinguished Paper Award

18. [**ICFEM-16**] Weikai Miao, Geguang Pu, Yinbo Yao, **Ting Su**, Danzhu Bao, Yang Liu, Shuohao Chen and Kunpeng Xiong. Automated Requirements Validation for ATP Software via Specification Review and Testing. The 18th International Conference on Formal Engineering Methods. ICFEM 2016. (CCF-C)
19. [**KSEM-15**] Ke Wu, Shiping Tang, Geguang Pu, Min Wu, **Ting Su**. Fm-QCA: A Novel Approach to Multi-value Qualitative Comparative Analysis. 8th International Conference on Knowledge Science, Engineering and Management. KSEM 2015. (CCF-C)
20. [**APSEC-14**] Yan Shen, Jianwen Li, Zheng Wang, **Ting Su**, Bin Fang, Geguang Pu and Wangwei Liu. Runtime Verification by Convergent Formula Progression. 21st Asia-Pacific Software Engineering Conference. APSEC 2014. (CCF-C)
21. [**TASE-13**] Yunhui Peng, Yanhong Huang, Ting Su, Jian Guo. Modeling and Verification of AUTOSAR OS and EMS Application. Seventh International Symposium on Theoretical Aspects of Software Engineering. TASE 2013. (CCF-C)

Papers Under Review and Submission

1. [**PLDI-20**] **Ting Su**, Yichen Yan, Junxin Li, Zhendong Su et al. Property-based Functional Fuzzing for Android Apps. 41st ACM SIGPLAN Conference on Programming Language Design and Implementation. 2020 (**CCF-A**)
2. [**TOSEM-19**] **Ting Su**, Chengyu Zhang, Yichen Yan, Lingling Fan, Geguang Pu, Yang Liu, Zhoulai Fu and Zhendong Su. Towards Efficient Data-flow Test Data Generation. ACM Transactions on Software Engineering and Methodology. TOSEM 2019. (IF: 2.516, **CCF-A**) *Major Revision.*
3. [**ICSE-19**] Anders Fischer-Nielsen, Zhoulai Fu, **Ting Su** and Andrzej Wsowski. The Forgotten Case of the Dependency Bugs: On the Example of the Robot Operating System. 42nd International Conference on Software Engineering. SEIP. 2020. **CCF-A**.
4. [**TDSC-19**] Sen Chen, Guozhu Meng, **Ting Su**, Lingling Fan, Minhui Xue, Yinxing Xue, Yang Liu, and Lihua Xu. AUSERA: Large-Scale Automated Security Risk Assessment of Global Mobile Banking Apps. IEEE Transactions on Dependable and Secure Computing. 2019. (IF: 4.41, **CCF-A**), *Major Revision.*

Posters and Workshop Papers

1. [**AI4Mobile-19**] Sen Chen, Lingling Fan, **Ting Su**, Lei Ma, Yang Liu, and Lihua Xu. Automated Cross-Platform GUI Code Generation for Mobile Apps, 1st International Workshop on Artificial Intelligence for Mobile. AI4Mobile 2019.
2. [**ICSTW-19**] Ruitao Feng, Guozhu Meng, Xiaofei Xie, **Ting Su**, Yang Liu, Shang-Wei Lin. Learning Performance Optimization from Code Changes for Android Apps. 2019 IEEE International Conference on Software Testing, Verification and Validation Workshops. ICSTW 2019.
3. [**KLEE-18**] Chengyu Zhang, **Ting Su**, Yichen Yan, Ke Wu, and Geguang Pu. Towards Efficient Data-flow Test Data Generation Using KLEE. The 1st International KLEE Workshop on Symbolic Execution. KLEE 2018.

STUDENT SUPERVISION

During my postdoc, I have co-supervised several Ph.D/master/undergraduate students at Nanyang Technological University (NTU), ETH Zurich (ETH), and East China Normal University (ECNU). For each of their work, I have made significant contributions to the idea, research, discussions and paper writing.

Ph.D students

1. **Lingling Fan**. Visiting Ph.D student at NTU from ECNU.
   Topic: Bug analysis of mobile apps.
   Published Papers: ICSE'18, ASE'18.
   Co-supervision, supervision percentage: 90%

2. **Chengyu Zhang**. Visiting Ph.D student at ETH from ECNU.
   Topic: Testing software model checkers.
   Published Papers: FSE'19.
   Co-supervision, supervision percentage: 90%
3. **Sen Chen**. Visiting Ph.D student at NTU from ECNU.
   Topic: Security analysis of mobile banking apps.
   Published Papers: FSE'18.
   Co-supervision, supervision percentage: 50%
4. **Jingling Sun**. Ph.D student at ECNU.
   Topic: Configuration analysis of mobile apps.
   Co-supervision, supervision percentage: 90%

Master and Undergraduate students

1. **YiChen Yan**. Visiting master student at ETH from ECNU.
   Topic: Testing C++ standard library.
   Co-supervision, supervision percentage: 90%
2. **Junxin Li**. master student at ECNU.
   Topic: Semantic testing of mobile apps
   Co-supervision, supervision percentage: 90%
3. **Wee Soon Lee Aaron**. Undergraduate student at NTU.
   Topic: Fuzzing of mobile apps.
   Co-supervision, supervision percentage: 90%

TEACHING EXPERIENCE

1. Teaching Assistant, Master Seminar, Research Topics in Software Engineering, Fall 2019, ETH Zurich (Prof. Zhendong Su)
2. Teaching Assistant, CZ2006 Software Engineering, 2017/2018 Semester 1, NTU (Prof. Yang Liu)
3. Teaching Assistant, Tools of Program Analysis, Testing, and Verification, 2015 Semester 1, ECNU (Prof. Geguang Pu)

PATENTS

Chinese Patents

All these patents are based on my research work.

1. Geguang Pu (Ph.D supervisor), **Ting Su**, Chengyu Zhang, Ke Wu, Yibo Yao, Hanru Zhou. An approach to cut point guided data-flow testing and its system. CN107656863A. Filed November 3, 2016
2. Geguang Pu (Ph.D supervisor), **Ting Su**, Ke Wu, Weiming Yang, Qiming Yao, Yao Yao. An stochastic model based testing approach for mobile applications. CN107656864A. Filed November 9, 2016

INVITED TALKS

- **Ting Su**. Large-Scale Analysis of Framework-Specific Exceptions in Android Apps. 12th Innovations in Software Engineering Conference (ISEC 2019, *the flagship software engineering conference in India*), February 2019, Pune, India. (Invited by Indian Chapter of SIGSOFT)
- **Ting Su**. Pushing the Limits of Automated Testing for Mobile Applications. The First International Workshop on Advances in Mobile App Analysis, A-MOBILE 2018 (co-located with ASE 2018), September 2018, Montpellier, France.
- **Ting Su**. Automated GUI Testing of Mobile Applications: Experience and Lessons Learned. ShanghaiTech University, May 2019, Shanghai, China.
- **Ting Su**. Guided, Stochastic Model-based GUI Testing of Android Apps. Cybertech Asia 2018. March 2018, Singapore.
- **Ting Su**. Stoat: A cloud-based Mobile App Testing Service. Defence Science and Technology Agency, Singapore (DSTA) March 2018, Singapore (Invited by DSTA)

PROFESSIONAL ACTIVITIES

Program Committee Member

- The 42nd International Conference on Software Engineering (ICSE), SEIP, 2020. (CCF-A)
- IEEE International Conference on Software Testing, Verification and Validation (ICST), Industry Track, 2019-2020. (CCF-C)
- International Symposium on Theoretical Aspects of Software Engineering (TASE), 2019-2020
- National Software Application Conference (NASAC), China, 2019. (CCF-C)
- International Workshop on Advances in Mobile App Analysis (A-Mobile), 2019-2020
- International Workshop on User Interface Test Automation and Testing Techniques for Event Based Software (INTUITESTBEDS), 2019
- International Workshop on Testing: Academia-Industry Collaboration, Practice and Research Techniques (TAIC-PART), 2019-2020

Invited Journal Reviewer

- Automated Software Engineering Journal (ASE), CCF-B
- Information and Software Technology (IST), CCF-B
- Software Quality Journal (SQJ), CCF-B
- Journal of Software: Evolution and Process, CCF-B
- IEEE Access

External Reviewer (selected)

- Joint meeting of the European Software Engineering Conference and the ACMSIGSOFT Symposium on the Foundations of Software Engineering (FSE), 2019
- International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS) 2018, 2019
- International Conference onSoftware Engineering (ICSE), 2019
- IEEE/ACM International Conference on Automated Software Engineering (ASE), 2018
- ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA), 2018
- IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 2019
- IEEE Transaction on Reliability, Frontiers of Computer Science, Journal of Computer Science and Technology (JCST)

LANGUAGES

Proficient in English and Mandrin.