Huanting Wang Curriculum Vitae

Email: schwa@leeds.ac.uk Personal Website: https://huantwang.github.io/

Education

10/2021 – Present	University of Leeds, United Kingdom	*Ph.D. in computer science* • **3 Publications (1 under review)** • **School of Computing Full Scholarship (2 places)**
09/2018 – 07/2021	Northwest University, China (Tier 1A)	*MSc in Software Engineering* • **4 Publications + 5 patents** • **3x First Class Scholarships (top 5% students), during 2018, 2019 and 2020**
09/2014 – 07/2018	Chang'an University, China (Tier 1A)	*BSc in Software Engineering*

Professional Experience

08/2022 - 08/2023	Huawei 2012 Labs *Research Intern*	*Research Intern in Security group*
08/2021 – 11/2021	Alibaba DAMO Academic *Research Intern*	*Research Intern in LLM group*
07/2019 – 12/2019	Ant Group *Software Engineer*	*Software Engineer Intern in Security group*

Selected Publications

[1] *Combining Structured Static Code Information and Dynamic Symbolic Traces for Software Vulnerability Prediction,*
H. Wang, Z. Tang, S. Chen, Jie. Wang, Y. Liu, H. Fang, C. Xia, Z. Wang,
Conditional accept of the International Conference on Software Engineering (ICSE), 2024
Premier ACM conference in Software Engineering (CORE A*)

[2] *Automating reinforcement learning architecture design for code optimization,*
H. Wang, Z. Tang, C. Zhang, J. Zhao, C. Cummins, H. Leather, Z. Wang,
Proceedings of the 31st ACM SIGPLAN International Conference on Compiler Construction (CC), 2022
Premier ACM conference in parallel computing (CORE A)
This work has received over 100 stars on GitHub and was featured on Nature.

[3] *Combining Graph-based Learning with Automated Data Collection for Code Vulnerability Detection,*
H. Wang, G. Ye, Z. Tang, S.H. Tan, S. Huang, D. Fang, Y. Feng, L. Bian, Z. Wang**,**
IEEE Transactions on Information Forensics and Security (TIFS), 2021
Flagship journal in computer security (CORE A*), ranking #2 of the top publication list in the "Computer Security & Cryptography" category according to Google Scholar's metric.
This work was featured on several online media and received over 130 citations.

[4] Deep Program Structure Modeling Through Multi-Relational Graph-based Learning,
G. Ye, Z. Tang, **H. Wang(Lead student author** first two authors are supervisors), S. Huang, Z. Wang
ACM Conference on Parallel Architectures and Compilation Techniques (PACT), 2020
Premier ACM conference in parallel computing (CORE A)

Programming Skills

Machine/Deep/Reinforcement/ Learning; Multi-task Learning; Software Security; Program Optimization;

Python; C++; Pytorch/Tensorflow;

Awards

2023	MITACS Globalink **Research Award**
2021	School of Computing **Full Scholarship** of University of Leeds (2 places)
2018, 2019, 2020	Scholarship for **First-Class Honors** at Northwest University (top 5%)

Research Experience

My research experience in the area of **vulnerability detection** and **compiler optimization** through the use of **machine learning techniques**. I have participated in the following projects during my MSc and Ph.D. study.

- **Robust Machine Learning for Program Modeling** **April 2022 to now**

In recent years, machine learning has emerged as a powerful tool for assisting code analysis and optimization tasks. Machine learning models can be vulnerable to changes in the deployment environment. Even slight alterations in hardware or application workloads can severely impact their accuracy.

We proposed a Python anomaly detection framework using conformal prediction to identify low-confidence model predictions. We applied our system to 11 representative machine learning models, covering optimization tasks such as heterogeneous device mapping, GPU thread coarsening, loop vectorization, and source-code level bug detection. our system successfully identifies 90% (up to 100%) of mispredicted test samples and enhances prediction performance in operational environments through incremental learning.

This work has led to one paper under review.

- **Hybrid Learning-based Software Vulnerability Prediction** **December 2021 to June 2023**

Deep Learning (DL) is increasingly employed for software bug and vulnerability detection, extracting program representations from static code sources such as code texts. DL may face challenges from complex code structures, redundant statements, and extensive execution paths, potentially reducing the performance.

We proposed use DL to learn program presentations by combining static source code information and dynamic program execution traces. By implementing a focused symbolic execution solution, we brings the benefits of static and dynamic code features while reducing the expensive symbolic execution overhead. We has successfully uncovered more than 100 unique vulnerabilities and yielding 36 new, unique CVE IDs and also outperforms 14 prior methods by providing higher accuracy and lower false positive rates.

This work has led to one paper published in ***ACM ICSE 2024 [1]****. Collaboration with* ***Huawei 2012 lab****.*

- **Automatic Reinforcement Learning Model Architecture Design** **July 2020 to Apr 2022**

While programmers apply reinforcement learning (RL) to their domain, the first step is to design the RL architecture for their tasks. However, expertise creates a barrier between programmers and RL.

We proposed an open-source framework for automating RL architecture search, simplifying RL integration into compilers. We applied it to four optimization problems: image pipelines, neural network code generation, code size reduction, and superoptimization. Experimental results demonstrate its superiority, improving performance and accelerating deployment-stage search by an average of 1.75x (up to 100x).

This work has led to one paper published in ***ACM CC 2022 [2]****. Collaboration with* ***Meta AI research lab****.*

- **Large Language Model** **August 2021 - November 2021**

My work involves creating and implementing machine learning models, tools, and infrastructure. I ensure their seamless integration into Alibaba's pre-trained model ecosystem.

- **Deep Program Structure Modeling using Graph Neural Networks** **June 2019 to January 2021**

Deep learning is promising for code-related tasks like compiler optimization. An important factor is having the right representation to characterize the model input for the given task. Existing approaches in the area typically treat the program structure as a sequential sequence but fail to capitalize on the rich semantics of data and control flow information, for which graphs are a proven representation structure.

We introduced a novel Graph Neural Networks (GNNs) approach to learn valuable code representations from program graphs, distinguishing diverse code relationships, including data and control flow, critical for downstream tasks. We apply our approach to four tasks: device mapping, thread coarsening, loop vectorization, and vulnerability detection, across various programming languages. Experimental results consistently show its superiority over competing methods.

This work has led to 3 papers published in ***IEEE TIFS 2021 [3], ACM PACT 2020 [4] and JISA 2023****.*
Collaboration with ***Ant group****.*