Research Statement

Michael Fu

As software becomes increasingly integral to daily life, the impact of cybersecurity breaches, such as the recent Medicare and Optus incidents in Australia, is more profound. Ensuring secure codebases is critical, yet manual inspection of millions of lines of code is impractical. While static analysis tools offer automation, their rule-based nature limits their scope. Leveraging advances in deep learning (DL) and large language models (LLMs), my PhD research focuses on developing AI-driven methods for automating security workflows during software development. We designed approaches that accurately identify line-level vulnerabilities, classify their types, estimate severity, and suggest repairs. These methods are integrated into a VSCode extension, providing real-time detection, explanation, and remediation within the developer's IDE. My future work will focus on two key areas: (1) AI4E—extending AI-driven security to all phases of DevSecOps, beyond just software development; and (2) SE4AI—addressing AI safety issues in AI-driven software products. My research tackles significant real-world problems and delivers practical solutions to maximize impact in both the software engineering and AI communities.

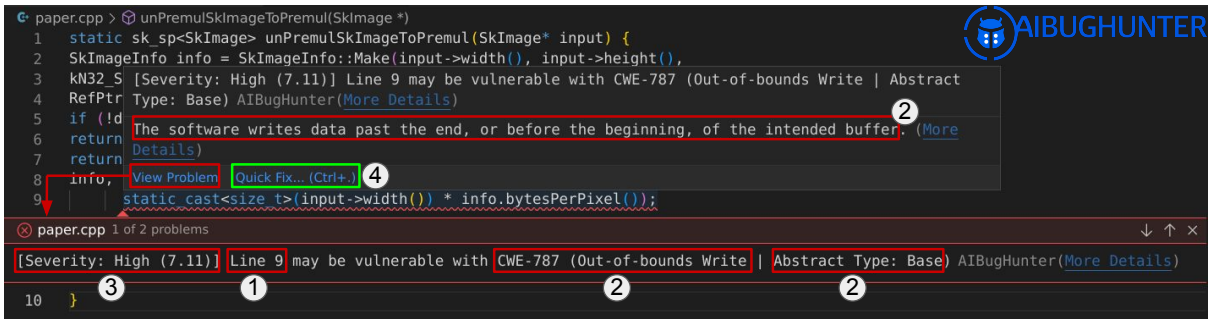

Figure 1: AIBugHunter: A practical AI-driven C/C++ security tool in VSCode.

Research Overview

My PhD research addresses pressing real-world cybersecurity challenges by developing cutting-edge DL models. My research works have been rigorously peer-reviewed and presented at leading software engineering conferences such as ICSE, FSE, ASE, and MSR, as well as published in top-tier journals including TSE, TOSEM, and EMSE. The impact of my research is substantial, evidenced by over 500 citations accrued within just over two years of commencing my PhD studies. Additionally, the proposed security tool has garnered nearly 1,000 downloads and secured competitive funding, including a Google Cloud research credit valued at 7,000 AUD. These achievements underscore the significant contributions and practical relevance of my research. Below, we provide a detailed overview of selected research projects.

Dectect Security Vulnerabilities

Problem Statement: Previous DL-based approaches focus on coarse-grained detection at the file or function level. Such predictions often include many lines that require manual inspection by security experts, requiring considerable effort.
Approach: We proposed two novel line-level vulnerability detection methods. The first utilizes pre-trained language models (LMs) and their self-attention mechanisms to infer line-level vulnerabilities [5]. The second employs optimal transport (OT) theory to quantize and learn vulnerability patterns from training data, leveraging these patterns during testing for precise line-level detection [1].
Results: Both methods were evaluated on datasets of vulnerable source code from real-world projects, showing significant performance improvements over previous studies.
Publications and Research Impact: Our first paper [5], accepted at the CORE A software engineering venue MSR 2022, has garnered 172 citations to date. The second paper [1] is under review at the top software engineering journal, TSE. Our work on line-level vulnerability detection has been featured in multiple media channels, including Australian Cybersecurity Magazine, Gizmodo Australia, Monash University, and the Information Age of the Australian Computer Society.

Explaining Detected Vulnerabilities

Problem Statement: Most prior works focus on detection without providing explanations for the types of vulnerabilities (i.e., CWE-IDs), which hinders security experts from effectively prioritizing their security backlogs and creating patches. Additionally, vulnerability data is highly imbalanced, with common vulnerabilities like buffer overflows having many training samples, while rare types have few samples, posing a significant data imbalance challenge.
Approach: To mitigate data imbalance, we split the data based on the hierarchical structure of CWE-IDs and learned expert models where each expert specializes in classifying specific CWE-IDs with high accuracy. Using a knowledge distillation framework, we transferred the expertise of these accurate models to a student model, enabling it to generalize across all CWE-IDs in our benchmark dataset [3].
Results: Our approach was evaluated on a benchmark of real-world vulnerability data with 91 different CWE-IDs, demonstrating superior performance over previous methods and effectively addressing data imbalance, as shown by our ablation study.
Publications and Research Impact: This work [3] was published in the top software engineering journal TSE in 2023 and has received 19 citations to date. It represents a significant advancement in addressing the data imbalance problem for CWE-ID classification, paving the way for more effective vulnerability explanation methods.

Repair Vulnerabilities Automatically

Problem Statement: Previous works leverage RNN-based repair models, which are suboptimal for handling comprehensive long sequences due to the sequential nature of RNNs. Additionally, vulnerabilities often exist in a few lines of code (LOC), but current repair approaches do not focus specifically on these critical areas, leading to less efficient repairs.
Approach: To address the limitations of RNN-based models, we explored the use of transformer-based pre-trained language models (LMs) for vulnerability repair, demon-

strating that self-attention mechanisms learn better global dependencies in source code [7]. Inspired by Vision Transformers (ViT), we proposed a vulnerability masking mechanism compatible with pre-trained LMs, allowing the decoder to focus on vulnerable code areas during repair, thereby enhancing accuracy [2].
Results: Our pre-trained LMs significantly outperformed RNN-based models in vulnerability repair tasks. The proposed vulnerability masking mechanism further improved the performance of LMs in repairing vulnerabilities.
Publications and Research Impact: The first paper [7] was accepted at the top software engineering conference FSE 2022 and has received 113 citations to date. The second paper [2] was published in the top software engineering journal TOSEM in 2023 and has received 9 citations to date.

Practical AI-Driven Security Tool in VSCode IDE

Problem Statement: In popular integrated development environments (IDEs) such as VSCode, existing security tools for C/C++ primarily rely on static analysis methods with predefined rules. Despite the development of various DL-based models for security analysis, there is a notable gap in integrating these models into practical tools that are readily accessible to developers and security experts.
Approach: To address this gap, we developed AIBugHunter, an AI-driven security extension for VSCode [6]. This tool supports C/C++ and automates several critical functions: ① detecting and locating vulnerable lines of code, ② providing detailed explanations of vulnerabilities via the 'view problem' button, ③ estimating the severity of detected vulnerabilities, and ④ suggesting potential repairs through the 'quick fix' button, as illustrated in Figure 1. Additionally, AIBugHunter is designed as a plug-and-play solution, allowing for easy updates to the backend DL model to accommodate other programming languages and incorporate advancements in model architectures.
Results: While the foundational models were evaluated in our previous research [7, 5, 1, 2, 3], we conducted further surveys and user experiments specifically for AIBugHunter. The results indicated high levels of satisfaction among software practitioners, with our tool significantly reducing the time required for vulnerability analysis from 15 minutes to 3-4 minutes.
Publications and Research Impact: This work was published in EMSE in 2023 and has accumulated 21 citations to date. The tool has garnered nearly 1,000 downloads from the VSCode marketplace. It has been showcased at university events such as Monash Open Day in 2022, 2023, and 2024, where it received considerable attention from the general audience. Additionally, it was presented at industry-related events like Monash TechFutureFest in 2022, generating significant interest in potential industry collaborations focused on AI for cybersecurity.

Future Research Directions

AI for SE - DevSecOps Security Automation. My PhD project focused on enhancing security automation within the software development phase of DevSecOps. In my future career, I plan to broaden this scope by developing AI-driven security automation solutions for other phases of DevSecOps, including planning, building, deploying, testing, and monitoring, to achieve a wider impact. To lay the groundwork for this ambitious

goal, I have conducted a systematic literature review [4], analyzing existing AI-driven approaches in DevSecOps, identifying their limitations, and uncovering promising research opportunities. In summary, one of my primary research directions is to advance AI for DevSecOps by proposing novel AI-driven solutions from both software engineering and security perspectives.

SE for AI - AI Safety. Another key research direction has emerged from my industry collaboration with Transurban, a road operation company in Australia. This work [10] highlighted significant challenges when applying AI in industrial contexts, particularly from a software engineering perspective. Our collaborative efforts, focused on engineering a retrieval-augmented generation (RAG) virtual assistant, led to the identification of eight critical AI safety challenges, especially in workflows involving large language models (LLMs). These challenges form the foundation of my research in SE for AI, where I aim to address the safety and reliability concerns of AI applications in industry.

In summary, my future research will be driven by two intertwined directions: AI for Software Engineering and Software Engineering for AI. Through the development of AI-driven security automation for DevSecOps and the exploration of AI safety within industrial applications, I aim to contribute significantly to both domains. By addressing these challenges, my research will not only advance the state of the art but also ensure that AI technologies are developed and deployed with the highest standards of security, safety, and reliability.

Research Career Plan

To support and expand my research initiatives, I plan to secure additional funding through various channels, including an application for the DECRA (Discovery Early Career Researcher Award) 2026. I also intend to actively pursue grant sponsorships from leading AI companies, such as the OpenAI research access program [9] and the NVIDIA academic grant program [8], as these resources are crucial for the computationally intensive aspects of my research. By diversifying funding sources and collaborating with industry leaders, I aim to enhance the scope and impact of my work, enabling cutting-edge advancements in the field. I will also seek to broaden my professional network by collaborating with experienced researchers, supervising students, and engaging with the broader academic community. Additionally, I plan to contribute to the field by joining editorial boards of top software engineering journals and participating as a program committee member in leading conferences. This involvement will help me stay at the forefront of research developments and offer my expertise to the community. Furthermore, I will actively seek industry collaborations to ensure that my research remains closely aligned with real-world challenges and has a tangible impact.

Summary

In summary, my PhD research has made significant contributions to software security automation through advanced DL models for vulnerability detection, explanation, and repair. Moving forward, I will focus on AI for DevSecOps security automation and AI safety, covering both AI for SE and SE for AI.

References

[1] Michael Fu, Trung Le, Van Nguyen, Chakkrit Tantithamthavorn, and Dinh Phung. Learning to quantize vulnerability patterns and match to locate statement-level vulnerabilities. *arXiv preprint arXiv:2306.06109*, 2023.

[2] Michael Fu, Van Nguyen, Chakkrit Tantithamthavorn, Dinh Phung, and Trung Le. Vision transformer inspired automated vulnerability repair. *ACM Transactions on Software Engineering and Methodology*, 33(3):1–29, 2024.

[3] Michael Fu, Van Nguyen, Chakkrit Kla Tantithamthavorn, Trung Le, and Dinh Phung. Vulexplainer: A transformer-based hierarchical distillation for explaining vulnerability types. *IEEE Transactions on Software Engineering*, 2023.

[4] Michael Fu, Jirat Pasuksmit, and Chakkrit Tantithamthavorn. Ai for devsecops: A landscape and future opportunities. *arXiv preprint arXiv:2404.04839*, 2024.

[5] Michael Fu and Chakkrit Tantithamthavorn. Linevul: A transformer-based line-level vulnerability prediction. In *Proceedings of the 19th International Conference on Mining Software Repositories*, pages 608–620, 2022.

[6] Michael Fu, Chakkrit Tantithamthavorn, Trung Le, Yuki Kume, Van Nguyen, Dinh Phung, and John Grundy. Aibughunter: A practical tool for predicting, classifying and repairing software vulnerabilities. *Empirical Software Engineering*, 29(1):4, 2024.

[7] Michael Fu, Chakkrit Tantithamthavorn, Trung Le, Van Nguyen, and Dinh Phung. Vulrepair: a t5-based automated software vulnerability repair. In *Proceedings of the 30th ACM joint european software engineering conference and symposium on the foundations of software engineering*, pages 935–947, 2022.

[8] NVIDIA. Nvidia academic grant program. `https://www.nvidia.com/en-us/industries/higher-education-research/academic-grant-program/`, 2024.

[9] OpenAI. Openai researcher access program application. `https://openai.com/form/researcher-access-program/`, 2024.

[10] Rui Yang, Michael Fu, Chakkrit Tantithamthavorn, Chetan Arora, Lisa Vandenhurk, and Joey Chua. Ragva: Engineering retrieval augmented generation-based virtual assistants in practice. *Under Review at 39th IEEE/ACM International Conference on Automated Software Engineering (ASE 2024) Industry Track*, 2024.