

ZHE YU

Department of Software Engineering, RIT, Rochester, NY, 14623
(919)-949-1382 ◊ <https://zhe-yu.github.io> ◊ zxyvse@rit.edu

I believe the future of AI is not replacing humans, but, rather, better supporting humans with automated intelligences. Hence, my research focuses on the creation of “human-in-the-loop” machine learning environments.

RESEARCH INTEREST

Software Engineering, Machine Learning, Information Retrieval, Human-AI Collaboration

EDUCATION

| | |
|-------------------------------------------------------------------------------|---------------------|
| PhD in Computer Science North Carolina State University | Aug 2015 - May 2020 |
| MS in Control Science and Engineering Shanghai Jiao Tong University | Sep 2011 - Mar 2014 |
| BS in Automation Shanghai Jiao Tong University | Sep 2007 - May 2011 |

EMPLOYMENT

| | |
|-------------------------------------------------------------------------------------------------------------------|---------------------|
| Assistant Professor Department of Software Engineering Rochester Institute of Technology, Rochester, NY | Aug 2020 - Present |
| Graduate Research Assistant Department of Computer Science North Carolina State University, Raleigh, NC | Aug 2015 - May 2020 |
| Summer Intern Traffic Estimation for Ads Google, Los Angeles, CA | May 2019 - Aug 2019 |
| Summer Intern Data Engine Machine Learning Google, Mountain View, CA | May 2018 - Aug 2018 |
| Summer Intern LexisNexis, Raleigh, NC | May 2017 - Aug 2017 |
| Summer Intern LexisNexis, Raleigh, NC | May 2016 - Aug 2016 |

SYNERGISTIC ACTIVITIES

Conference Committee Member: ICSE 2022, MSR 2021-2022, ICSME 2021-2022, SCAM 2020-2021 NIER track
Conference Review: CSCW 2020, ICML 2022-2023, NeurIPS 2022
Workshop (Co-)Organizer: Fairware'23 at ICSE'23
Journal Review: IEEE Transactions on Software Engineering (TSE), ACM Transactions on Software Engineering and Methodology (TOSEM), Empirical Software Engineering (EMSE), Information and Software Technology (IST), Journal of Systems & Software (JSS), Automated Software Engineering (ASE), Machine Learning, IEEE Access
Guest Editor: JSS Special Issue: Over the Horizon: Limits and Breakthroughs in Software Fairness

GRANTS

| | |
|------------------------------------------------------------------------------------|----------------------------------|
| NSF CRII: SHF: Testing Fairness in Human Decisions with Algorithmic Bias | Jun 2023 - May 2025 \$174,999 |
|------------------------------------------------------------------------------------|----------------------------------|

TEACHING

DSCI-633: Foundation of Data Science
SWEN-352: Software Testing

RECENT RESEARCH PROJECTS

| | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|
| Machine Learning Fairness in Regression. <i>Rochester Institute of Technology</i> | Oct 2021 - Present |
| <ul style="list-style-type: none">While fairness in classification problems has been intensively explored, fairness in regression is still under-explored. | |

- This project aims to explore ML fairness in regression problems such as
 1. How to better measure fairness in regression?
 2. How to reduce bias in regression problems?
 3. Is there a generalized methodology for both classification and regression fairness problems?

Human Ethical Bias Detection with Machine Learning

Aug 2020 - Present

Rochester Institute of Technology

- Instead of trying to reduce bias in machine learning software, this project aims to reduce bias from its source—the human decisions.
- That is, this work proposes to utilize the machine learning bias inherited from the training data (human decisions) as an indicator for detecting human bias.
- While it is difficult to directly test whether a human has bias, with current research on machine learning bias, it is now easy to test, in large scale at low cost, whether a machine learning software has bias.

Trace Link Prediction with Dual-Encoders

Aug 2021 - Present

Rochester Institute of Technology

- Trace link prediction is a crucial problem to facilitate the maintenance of software projects.
- This work proposes to customize the dual-encoder architecture for trace link prediction, especially in the inference strategy and the loss function for training.
- The designed dual-encoders will be able to encode different software artifacts into a shared encoding space. In that shared encoding space, artifacts connected by trace links would be closer to each other than the unconnected ones.

Few-shot Active Learning for Better Information Retrieval

Aug 2020 - Present

Rochester Institute of Technology

- This project aims to improve efficiency of information retrieval with a combination of few-shot learning and active learning.
- When a deep neural network model learns to retrieve different relevant information (of the same type, e.g. literature reviews of different topics), it optimizes its inner structure so that it will learn faster and better to retrieve new, unseen information.
- Active learning is also important in retrieving new, unseen information by (1) continuously learning from human decisions on which is relevant and (2) suggesting what should be reviewed next based on the learned model.

Better self-admitted technical debt identification with adaptive CNN.

Aug 2020 - Present

Rochester Institute of Technology

- Incorporate a CNN model in the framework of a human-in-the-loop self-admitted technical debt identification system.
- Explore ways to efficiently update the CNN model with incrementally obtained labels.
- Make the prediction model of CNN adaptive to the target software project.

Total Recall and Software Engineering

Aug 2015 - May 2020

North Carolina State University

- Apply machine learning algorithms to support human retrieve all desired information from big data with less effort, a class of information retrieval problem called total recall.
- Developed an active learning based framework—FASTREAD—to support fast selection of primary studies in systematic reviews and all the total recall problems.
 - Validated in simulations, FASTREAD was usually able to find 95% of relevant studies by asking humans to review 10% of the candidates, which outperformed the prior state of the art total recall solutions.
 - FASTREAD accurately estimated the total number of relevant studies in the candidates and provided a reliable stopping rule for high target recalls, e.g. 90%, 95%, or 99%.
 - FASTREAD suggested which labels should be double checked to correct human errors. By double checking 50% of the labeled studies, 96% of the human errors could be covered.
- A tool has been developed to implement FASTREAD at <https://github.com/fastread/src>.
- Same idea applied to solve other software engineering problems such as software security vulnerability prediction and test case prioritization.

Test Case Prioritization for Automated UI Testing

Sep 2018 - Apr 2019

Cooperation project with LexisNexis Legal & Professional

- Conducted a systematic literature review on test case prioritization researches, using the FASTREAD tool.
 - Validated by 6 graduate students, 90% of the relevant studies were found by reviewing 6% of the candidates with FASTREAD targeting at 90% recall.
- Proposed a novel test case prioritization framework by adapting FASTREAD to the automated UI testing problem.
- Improved performance by 9% (measured in APFDc) using the proposed framework.

Social Network of US Public Companies

Feb 2018 - Dec 2018

Cooperation project with LexisNexis Legal & Professional

- Extracted board of directors from 10-K filings by rule-based named entity recognition.
- Connected companies with mutual board of directors (find connected components in the graph).
- Found that 40% US public companies were fully connected with each other while the rest were isolated ones.
- Validated that 70% of the US top 500 companies were fully connected. This suggested that companies connected with others are more likely to succeed.

Scalable FASTREAD on HPCC Systems

Feb 2017 - Dec 2017

Cooperation project with LexisNexis Legal & Professional

- Implemented FASTREAD tool on HPCC Systems for high scalability.
- Enabled multi-users to work on the same project in parallel.

Youtube eCPM Seasonality in TEA

May 2019 - Aug 2019

Internship at Google (TEA: Traffic Estimation for Ads)

- Analyzed which features are significantly correlated to outliers (when forecasts were way off from actuals).
 - Enabled null hypothesis tests on scalar features.
 - Built a new feature for the internal validation tool to analyze outlier features in drilldown pages.
- Added seasonality predictions to the current TEA forecasts
 - Improved seasonality predictions—time series analysis on previous years to predict the curve in next year.
 - Validated the overall TEA forecasting performance improvement with seasonality predictions via A/B testing.

KIWI: Knowledge In Web Images

May 2018 - Aug 2018

Internship at Google (DEML: Data Engine Machine Learning)

- Mined image-entity pairs in web images with alt text, image url, etc. Trained a model to measure the image-entity pair quality and filter out low-quality pairs.
 - Trained a dual encode model, between entity and image starburst.
 - Designed and tried different metrics to evaluate the model performance.
 - Added a feature to dual encoder framework to support dense feature.

Legal Document Headnotes Generation and Classification

May 2017 - Aug 2017

Internship at LexisNexis Legal & Professional

- Developed a text summarization framework for generating “headnote” of more than 1 million legal documents.
- Designed a scalable classification scheme with doc2vec to categorize documents into specific legal topics.
- Demonstrated that the above framework can reduce document review time by $\geq 50\%$ according to user surveys.

Improve Legal Document Retrieval Efficiency of DiscoveryIQ

May 2016 - Aug 2016

Internship at LexisNexis Legal & Professional

- Created a sandbox for prototyping new DiscoveryIQ features.
- Developed new features to “open the black box” of DiscoveryIQ.
- Incorporate new features into the current DiscoveryIQ product.

SELECTED PUBLICATIONS

[1] Zhe Yu and Xiaoyin Xi. “A Pilot Study on Detecting Unfairness in Human Decisions With Machine Learning Algorithmic Bias Detection.” arXiv preprint arXiv:2112.11279.

[2] Zhe Yu, Joymallya Chakraborty, and Tim Menzies. “Fairer Machine Learning Software on Multiple Sensitive Attributes With Data Preprocessing.” arXiv preprint arXiv:2107.08310.

[3] Zhe Yu, Jeffrey C. Carver, Gregg Rothermel, and Tim Menzies. “Assessing expert system-assisted literature reviews with a case study.” Expert Systems with Applications (2022): 116958.

- [4] Abdullah AlOmar, Eman, Jiaqian Liu, Kenneth Addo, Mohamed Wiem Mkaouer, Christian Newman, Ali Ouni, and Zhe Yu. "On the Documentation of Refactoring Types." *Automated Software Engineering Journal*. 2021.
- [5] Zhe Yu, Fahmid Morshed Fahid, Huy Tu, and Tim Menzies. "Identifying Self-Admitted Technical Debts with Jitterbug: A Two-step Approach." *IEEE Transactions on Software Engineering*.
- [6] Yang, Xueqi, Zhe Yu, Junjie Wang, and Tim Menzies. "Understanding static code warnings: An incremental AI approach." *Expert Systems with Applications* 167 (2021): 114134.
- [7] Yang, Xueqi, Jianfeng Chen, Rahul Yedida, Zhe Yu, and Tim Menzies. "Learning to recognize actionable static code warnings (is intrinsically easy)." *Empirical Software Engineering* 26, no. 3 (2021): 1-24.
- [8] Chakraborty, Joymallya, Suvodeep Majumder, Zhe Yu, and Tim Menzies. "Fairway: A Way to Build Fair ML Software." In *Proceedings of ESEC/FSE 2020*.
- [9] Agrawal, Amritanshu, Tim Menzies, Leandro L. Minku, Markus Wagner, and Zhe Yu. "Better software analytics via DUO: Data mining algorithms using/used-by optimizers." *Empirical Software Engineering* 25, no. 3 (2020): 2099-2136.
- [10] Huy Tu, Zhe Yu, Tim Menzies. 2020. "Better Data Labelling with EMBLEM (and how that Impacts Defect Prediction)." *IEEE Transactions on Software Engineering*.
- [11] Zhe Yu, Christopher Theisen, Laurie Williams, Tim Menzies. 2019. "Improving Vulnerability Inspection Efficiency Using Active Learning." *IEEE Transactions on Software Engineering*.
- [12] Zhe Yu, Fahmid M. Fahid, Tim Menzies, Gregg Rothermel, Kyle Patrick, Snehit Cherian. 2019. "TERMINATOR: Better Automated UI Test Case Prioritization." In *Proceedings of ESEC/FSE'19, Software Engineering in Practice*, 883-894. <http://doi.acm.org/10.1145/3338906.3340448>
- [13] Zhe Yu and Tim Menzies. 2019. "FAST2: An intelligent assistant for finding relevant papers." *Expert Systems with Applications*. 120: 57-71. <https://www.sciencedirect.com/science/article/pii/S0957417418307413>
- [14] Zhe Yu, Nicholas A. Kraft, and Tim Menzies. 2018. "Finding Better Active Learners for Faster Literature Reviews." *Empirical Software Engineering*. 23(6): 3161-3186. <https://link.springer.com/article/10.1007/s10664-017-9587-0>
- [15] Zhe Yu and Tim Menzies. 2018. "Total recall, language processing, and software engineering." In *Proceedings of NL4SE Workshop 2018*, 10-13. <https://dl.acm.org/citation.cfm?id=3283818>
- [16] Vivek Nair, Zhe Yu, Tim Menzies, Norbert Siegmund, and Sven Apel. 2018. "Finding faster configurations using flash." *IEEE Transactions on Software Engineering*.
- [17] Vivek Nair, Amrit Agrawal, Jianfeng Chen, Wei Fu, George Mathew, Tim Menzies, Leandro Minku, Markus Wagner, and Zhe Yu. 2018. "Data-Driven Search-based Software Engineering." *The Mining Software Repositories (MSR)*.
- [18] Agrawal, Amritanshu, Tim Menzies, Leandro L. Minku, Markus Wagner, and Zhe Yu. "Better software analytics via DUO": Data mining algorithms using/used-by optimizers." *Empirical Software Engineering* 25, no. 3 (2020): 2099-2136.