# XUEQI YANG

Homepage, Github, Linkedin, ResearchGate

+1-571-392-0734 ⋄ xyang37@ncsu.edu

## EDUCATION

**PhD in Computer Science** — Aug 2018 - Present
North Carolina State University, Advisor: Dr. Tim Menzies
Research interests: **Software Engineering, Static Code Analysis, Data Mining and Deep Learning**
Coursework: Automated SE | Algorithm | Networking | Spatial Temporal Data Mining | Natural Language Processing

**Bachelor in Information Management and Information System** — Sep 2014 - July 2018
Dongbei University of Finance and Economics, China, GPA: 90/100
Coursework: C | Java | Data Structure | Data Mining | Database | Web Design | Operation Research

## SKILLS AND STRENGTHS

| | |
|---|---|
| **Languages** | Python, C, Java, Bash, JavaScript, MATLAB, SQL, ASP.Net |
| **Tools** | PyTorch, Keras, TensorFlow, Scikit-learn, LaTex |
| **Others** | Operation Research, Statistics |

## WORK EXPERIENCE

**Microsoft Research** — May 2022 - Aug 2022
Research Intern in Cloud and Infrastructure Security Group — Seattle, WA
Mentor: Mariusz Jakubowski
Address Transformer Token-length limitation with **Learned Token Pruning** algorithm and **Longformer** to detect vulnerability in command line dataset with long input sequences.

Explore recent advancements in natural language processing (**CodeBERT** and other Transformer-based models) in helping to structure and analyze peta-scale security data in cloud environments.

## SELECTED PROJECTS

**Security Bug Detection and Localization** — May 2021 - Feb 2022
*NSF funded project in the RAISE lab* — *Raleigh, NC*

· Propose a **lexer** to resolve the feature extraction in security vulnerability analysis in open-source C/C++ projects.
· Leverage **placeholders** in token embedding in source code to address the out-of-vocabulary (OOV) issue.
· Utilize an attention mechanism, **CAM (Class Activation Mapping)**, to interpret and localize the vulnerability in source code.

**Detection for Static Defects with Incrementally Active Learning** — July 2020 - Dec 2020
*NSF funded project in the RAISE lab* — *Raleigh, NC*

· Test the Linux mainline at source tree level with **coccinelle**, a program matching and transformation engine which provides the language SmPL (Semantic Patch Language) for specifying desired matches and transformations in C code.
· Implement **feature extractors** from warning messages and patches generated from coccinelle with **TF-IDF and code2vec embedding methods**.

**Simpler Hyperparameter Optimization for Software Analytics** — May 2020 - Sep 2020
*NSF funded project in the RAISE lab* — *Raleigh, NC*

· Apply a **simpler hyperparameter optimization** (DODGE, using a technique called $\epsilon$-domination) to 120 SE datasets to find the optimal control settings for data miners.
· Avoid the high training overhead by **evaluating and ranking the parameter space** in comparison with traditional optimizers, either Differential evolution (DE) or Genetic algorithm (GA).
· Implement **Box-counting methods** to estimate the intrinsic dimensionality of SE datasets and standard Machine learning datasets (UCI).

### Detection for Static Defects with DNN Models
*NSF funded project in the RAISE lab*

Sep 2019 - Jan 2020

Raleigh, NC

· Implement **deep neural networks** in Keras and PyTorch with static defect artifacts to predict real defects to act on.
· Utilize **regularisers** to avoid DNN models from overfitting and lower the running overhead.
· Use Box-counting methods to explore the **intrinsic dimension** of SE data and match the complexity of machine learning algorithms with the datasets it handles.

### Static Warnings Analysis using active learning
*NSF funded project in the RAISE lab*

Jan 2019 - Aug 2019

Raleigh, NC

· Identify actionable static warnings of nine Java projects generated by FindBugs with **incrementally active learning** and machine learning algorithms to achieve higher recall with lower cost by reducing false alarms.
· Utilize different **sampling approaches** (random sampling, uncertainty sampling and certainty sampling) to query warnings suggested by active learning algorithm.
· Interact the system with a human oracle to update the system.

### Multi-task Learning for Evaluating Peer Assessments
*Coursework project*

Sep 2020 - Dec 2020

Raleigh, NC

· Leverage a benchmark **language representation model** (BERT, Deep Bidirectional Transformers) in multi-task learning to automatically evaluate peer feedback comments. Utilize **oversampling method** (at data-level and algorithm-level) to avoid the data imbalance issue. Use **Subword Tokenization** method, WordPiece which splits a text into subwords, to address the out-of-vocabulary (OOV) problem in NLP. And compare the performance of multi-task model with baseline method, single-task model.
· Implement **word2vec** (CBOW and Skip-grams) and **doc2vec** (Doc2vec and Part-of-speech tagging) models in Python 3 on Sentimental Analysis Dataset and Question Answering Dataset. Compare performance of the proposed methods with baseline methods (TF-IDF and BOW) in individual projects.

### Spatial Temporal Object Change Detection and Localization
*Coursework project*

Jan 2020 - May 2020

Raleigh, NC

· Utilize **Mask R-CNN** implemented with PyTorch for satellite image change detection and localization.
· **Assess building damage** from satellite imagery with a variety of disaster events and different damage extents.

### SmartWeather App with Agile Development
*Coursework project*

Jan 2020 - May 2020

Raleigh, NC

· Implement SmartWeather App in C# with Xamarin and Visual Studio.
· Follow the **Scrum process** to iterate and manage software development.
· Use Architecture Diagram, Context Diagram and Quality Attribute Scenarios in software design.
· Utilize **Fuzzy logic controller** to convert a crisp input value into a fuzzy set with a predetermined lower and upper bound of impreciseness.

## PUBLICATIONS

[1] Xueqi Yang, Mariusz Jakubowski and Tim Menzies, Sparsity Attention in Transformer Model for Long Sequence Analysis to Detect Security Vulnerability in C/C++ Open-source Projects, **(under submission)**.

[2] Rahul Yedida, Hong Jin Kang, Huy Tu, Xueqi Yang, David Lo, Tim Menzies, How to Find Actionable Static Analysis Warnings, **Transactions on Software Engineering, (accepted)**, 2023.

[3] Xueqi Yang, Jianfeng Chen, Rahul Yedida, Zhe Yu and Tim Menzies, Learning to Recognize Actionable Static Code Warnings (is Intrinsically Easy), **Empirical Software Engineering, (accepted)**, 2021, **International Conference on Software Engineering Journal-First, (accepted)**, 2022.

[4] Xueqi Yang, Zhe Yu, Junjie Wang and Tim Menzies, Understanding Static Code Warnings: an Incremental AI Approach, **Expert Systems with Applications (accepted)**, 2021.

[5] Amritanshu Agrawal, Xueqi Yang, Rishabh Agrawal, Xipeng Shen and Tim Menzies, Simpler Hyperparameter Optimization for Software Analytics: Why, How, When?, **Transactions on Software Engineering (accepted)**, 2021.

[6] Jian Luo, Xueqi Yang, Ye Tian and Wenwen Yu, Corporate and Personal Credit Scoring via Fuzzy Non-kernal SVM with Fuzzy within-class Scatter, **Journal of Industrial and Management Optimization (accepted)**, 2017.