WENFEI TANG

Phone: (+1) (734)8812721 ⋄ **Email**: twenfei@umich.edu
LinkedIn: www.linkedin.com/in/twenfei ⋄ **Github**: github.com/twenfei

EDUCATION

University of Michigan, Ann Arbor, MI
M.S. in Computer Science Engineering *Dec 2022*
B.S. in Computer Science with *Honors* and *Distinction*, **Cumulative GPA: 3.825/4** *May 2021*
Course Highlights: Natural Language Processing, Database Management Systems, Data Structures and Algorithms, Web Systems, Introduction to Machine Learning, User Interface Development, Introduction to Computer Security, Electronic Commerce, Compiler Construction, Linear Spaces and Matrix Theory, Differential Equations

SKILLS

Programming Language	C++/C, Python, HTML/CSS, MATLAB, JavaScript
Software & Tools	Visual Studio, Git, shell scripting, Linux, Pytorch, LaTeX, Unity3D

WORK EXPERIENCE

NETSCOUT, Ann Arbor, Michigan *Aug 2021*
Software Engineering Intern - Cybersecurity

- Developed visualization tools for network traffic analysis and cyber threats on web applications at the Cybersecurity Department
- Maintained APIs for mitigations of potential network security problems using Python, React.js, and REST API

University of Michigan, Electrical Engineering and Computer Science Department *Feb 2020 - Present*
Graduate Student Instructor, Instructional Aide - EECS 281: Data Structures and Algorithms

- Help manage a class of around 900 students
- Responsible for teaching lab sections, setting up class projects, writing and grading exams, and holding office hours

CONFERENCE

Wenfei Tang, Sundaresh Ram*, Alexander J. Bell, Cara Spencer, Alexander Buschhaus, Charles R. Hatt, Marina Pasca diMagliano, Stefanie Galban, and Craig J. Galban. "**Detection of Cancer Lesions in Histopathological Lung Images Using a Sparse PCA Network**". Presented at 2021 AACR Conference on Artificial Intelligence, Diagnosis, and Imaging. DOI: 10.1158/1557-3265.ADI21-PO-086. Published March 2021.

PROJECTS

Automated Lung Cancer Lesion Detection on H&E Stained Slides [Code] *July 2019 - Dec 2020*
Galban Lab, Department of Radiology, University of Michigan *Research Assistant*

- Proposed a baseline neural network method called GS-PCANet, which outperforms six other open-source histopathology image classification with the precision of 0.872 and accuracy of 0.908 on a dataset of 67 segmented lung images (∼3000×3000 pixels)
- Independently developed an automated computer-aided tool for detection of potential cancerous regions on lung cancer images
- Project abstract accepted and presented on 2021 American Association for Cancer Research Conference as first author

MFocus, A Web Application for Managing Daily Tasks [Code] *Oct 2020 - Dec 2020*

- Collaborated with another teammate to develop an efficiency tool for managing tasks with HTML, Javascript (Vue.js) and CSS
- Designed an interactive reward system in the app where users can raise an e-pet
- Users can interact with the e-pet, purchase items for their pets, manage tasks and play Spotify music on the app

Mechanism Design for Parking Allocation Problem [Code] *Oct 2019 - Dec 2019*

- Collaborated with another teammate to model the parking allocation problem using a multi-agent system
- Simulated these mechanisms with varied number of agents (up to 2000 agents), number of slots and probabilistic models in Python
- Evaluated three valuation schemes based on the profitability of the mechanism, and the welfare of agents and chose the best pricing system

An Instagram Clone, Dynamic Page Development *Sept 2020 - Dec 2020*

- Developed a static site generator from templates using HTML and Python (Jinja2)
- Implemented the server-side dynamic pages with Flask and SQLite
- Designed client application in JavaScript and used the REST API to achieve client-side dynamic pages in a three-memeber group

Compiler Construction *Jan 2020 - Apr 2020*

- Built a working compiler to transfer Decaf Language (a strongly-typed, object-oriented language with support for inheritance and encapsulation) into MIPS language (an assembly language) in a two-member group
- Developed both the front end and the back end parts of a compiler including parser, scanner, semantic analyzer, code generator and code optimizer