

Nengneng Yu

+1 484-995-8987 | ynn1999@umd.edu | College Park, MD

EDUCATION

- **University of Maryland College Park** College Park, MD
Doctor of Philosophy in Computer Science
Advisor: Zaoxing(Alan) Liu
Aug 2023 - Present
- **Boston University** Boston, MA
Bachelor of Science in Computer Engineering
Magna cum Laude, Highest honors in the major
Sep 2019 - May 2023

PUBLICATION & WORKS

- [1] Yajie Zhou*, **Nengneng Yu***, Zaoxing Liu, “Towards Interactive Simulacra of Internet Investigation by Human Researchers”, Hot Topics in Networks (HotNets), 2023
- [2] Yajie Zhou, **Nengneng Yu**, Simiao Zuo, Yue Yu, Haoming Yi, Chao Zhang, Tuo Zhao, Zaoxing Liu, “Fine-Grained, Adaptive Advanced Persistent Threats Detection with SENTINEL”, Under Submission

RESEARCH PROJECTS

- **AI-based Generative Proteomics System for Cancer Detection** Feb 2024 – Present
Mentor: Zaoxing(Alan) Liu, Yufan Wang *Froot Lab, UMD & Johns Hopkins Medicine*
 - Traditional Bio-Informatics methods struggle with high-dimensional, small-sample proteomics data, leading to poor classification accuracy and inability to distinguish between different stages of gastric cancer. Developed an **AI-based generative data system** to enhance detection accuracy and cancer stage differentiation.
 - Built a pipeline utilizing **Diffusion** models to generate synthetic samples, combined with **XGBoost** for downstream analysis to identify key biomarkers. Using **Pytorch** for model implementation. Designed the next-phase AI system to incorporate **distributed computing, transfer learning, and federated learning** for better performance and scalability.
 - Achieved near 100% accuracy in gastric cancer NAT/Tumor classification. Improved CPTAC dataset performance by 10-20%, Korean dataset from 70% to over 90% on weighted accuracy among cancer stages.
- **Accelerating the mapping system for Micro Aerial Vehicle (MAV) navigation** Sep 2023 – Present
Mentor: Peiqing Chen *UMD*
 - Developed a **software caching layer** using **C++** and **OpenMP**, integrated it into the **OctoMap** mapping system to reduce redundant data access and computation, enabling faster map updates and queries.
 - Deployed the optimized OctoMap with caching enhancements into a **ROS-based MAV simulation environment** for testing and validation, improving data handling and responsiveness in a virtual autonomous navigation setup.
 - Achieved up to 45% reduction in end-to-end workflow runtime and increased flight speed by 26%, significantly enhancing real-time mapping performance and navigation efficiency.
- **Interactive Research Agents for Internet Incident Investigation** May 2023 – Nov 2023
Mentor: Zaoxing(Alan) Liu *Froot Lab, UMD*
 - Developed an LLM-based agent to simulate experienced researchers and automate the investigation process, addressing the inefficiencies of traditional manual and time-consuming Internet incident investigations.
 - Built an agent using **Auto-GPT** and **GPT-4**, equipped with autonomous **information retrieval, knowledge memory, and self-learning capabilities**. Tested it on challenging scenarios such as the impact of hypothetical solar storms on networks.
 - Achieved 87.5% consistency in insights compared to human experts, effectively automating complex Internet incident analysis.
- **Advanced Persistent Threat (APT) Detection and Analysis** Feb 2022 – Present
Mentor: Zaoxing(Alan) Liu, Tuo Zhao *Red Hat & Boston University & Georgia Tech*
 - Developed an APT detection system to overcome existing models’ limitations in adapting to new attack types and data scarcity, while improving fine-grained detection precision.
 - Designed a **Multi-Classification-Head Transformer** with a **pre-training** and **fine-tuning** workflow for efficient knowledge transfer. Implemented a flexible data processing pipeline using **Python** and **Pandas**, and applied attention mechanisms to enhance interpretability.
 - Achieved 10%-29% improvement in fine-grained detection accuracy and 25%-50% reduction in training data needs across 8 real-world APT datasets. Enhanced model transparency, aiding in better post-attack investigation.

TECHNICAL SKILLS

Programming languages: C++, C, Python, C#, Java
Web Technologies: HTML, CSS, Flask, React, JavaScript
ML/AI: PyTorch, Numpy, Pandas, Matplotlib
Miscellaneous: Linux, GDB, Git, Shell, MySQL, Latex

SERVICES

- **University of Maryland College Park Department of Computer Science** Aug 2023 - Present
Graduate Teaching Assistant *CMSC414 Network and Security, CMSC250 Discrete Structure*
- **Boston University College of Engineering** Sep 2022 - Dec 2022
Teaching Assistant *EC440 Operating System*
- **Boston University College of Engineering** Jan 2022 - May 2022
Teaching Assistant *EC414 Machine Learning*

COURSE PROJECTS

- **Concurrency Control Schemes for Transaction Processing Systems** Jan 2023 – May 2024
 - Implemented and evaluated six concurrency control schemes, including **Two-Phase Locking (2PL)**, **Optimistic Concurrency Control (OCC)**, and **Multi-Version Concurrency Control (MVCC)**. Developed two versions of 2PL with exclusive and shared locks, serial and parallel validation versions of OCC, and a simplified MVCC with **Serializable Snapshot Isolation (SSI)**.
 - Developed and integrated the concurrency control schemes into a **main-memory key-value store** using **C++** and **thread management** techniques. Built a prototype transaction processing framework with a custom lock manager and multi-threaded execution support.
 - Conducted performance benchmarking with **CMake** and **CTest** to assess throughput and latency across varying transaction lengths and contention levels.
- **eBPF Modularity Project** Sep 2022 – Dec 2022
 - Collaborated with professors from Brown University and IBM engineers to advance research on building a comprehensive and reusable **eBPF module library**.
 - Leveraged **OPENED** tool to analyze and decompose eBPF programs from open-source projects, extracting reusable modules for improved modularity and maintainability.
 - Designed a framework to transform extracted modules into a format compatible with **Bumblebee** tools for generating OCI images. Integrated “glue logic” for seamless module compatibility with **L3AF** and **Polycube**.
- **Basic Unix-like Operating System** Jan 2022 – May 2022
 - Implemented a custom shell capable of executing commands via system calls, redirecting stdin/stdout, handling multiple command pipelines, and supporting background execution with “&”.
 - Developed a subset of the **POSIX threads API** in user mode, enabling multi-threaded execution with round-robin scheduling for effective resource management.
 - Designed and implemented a copy-on-write (COW) thread-local storage (TLS) mechanism to enable data sharing between threads while ensuring isolation of changes, improving memory management at the thread level.