

MEHBUBUL HASAN AL-QUVI

alquvi.m@northeastern.edu
Website | LinkedIn | Google Scholar | GitHub

EDUCATION

Northeastern University

Boston, USA

PhD in Computer Science (GPA: 4.00/4.00)

Sep 2025 – May 2030 (Anticipated)

- **Research Interest:** Performance Debugging, System Reliability, Program Analysis, Compilers, Operating System.

Bangladesh University of Engineering and Technology (BUET)

Dhaka, Bangladesh

BSc in Computer Science and Engineering (GPA: 3.84/4.00)

Apr 2019 – Jul 2024

RELATED COURSEWORK

Compiler Construction, Operating Systems Implementation, Computer Architecture, Data Structures and Algorithms Analysis, Advanced Algorithm Engineering.

TECHNICAL SKILLS

Programming Language: Proficient in C, C++, Java, Assembly, Bash, Python

Systems, Frameworks, and Tools: LLVM, QEMU, OpenSBI, GDB, eBPF, Intel Pin, perf, Linux Kernel, Git, SSH

PUBLICATIONS

1. Shamit Fatin, Mehbubul Hasan Al-Quvi, Haz Sameen Shahgir, Sukarna Barua, Anindya Iqbal, [and 4 others]. LELANTE: LEveraging LLM for Automated ANDroid TESting. In *EASE'25*. Pages 701-707. [\[PDF\]](#)

WORK EXPERIENCE

Graduate Research Assistant at Northeastern University

Sep 2025 – Present

Systems Research Group (SRG). Supervisor: [Prof. Xiang \(Jenny\) Ren](#)

Systems Research Intern at University of Michigan

Jul 2024 – Dec 2024

Ordered Systems Lab (OrderLab). Supervisor: [Prof. Ryan Huang](#)

Research Assistant at Bangladesh University of Engineering and Technology (BUET)

Jun 2023 – Aug 2025

Applied Machine Learning Lab. Supervisor: [Prof. Anindya Iqbal](#)

PROJECT EXPERIENCE

Performance Bug Detection and Root Cause Localization in Production Systems

Sep 2025 – Present

Role: *Project Lead*, Supervisor: [Prof. Xiang \(Jenny\) Ren](#)

- Building online performance debugging techniques that automatically identify the root causes of performance issues in production environments with minimal overhead.
- Developing techniques to automatically determine where to place performance metrics in large software systems .
- Applying sampling-based profiling (e.g., perf), static and dynamic binary instrumentation (e.g., Intel Pin, uprobes) to diagnose performance regressions and latency anomalies.

A Framework for Enhancing the Flexibility and Runtime Safety of eBPF Programs

Jul 2024 – Dec 2024

Role: *Co-Developer*, Supervisor: [Prof. Ryan Huang](#)

- Addressed key limitations of the eBPF verifier, including false rejections of valid programs due to overly conservative static checks, and false negatives from safety violations undetected by static-only analysis.
- Contributed to the framework's custom Intermediate Representation (IR) for BPF program transformation and instrumentation.

- Wrote IR-level passes that convert specific verifier rejections into runtime checks, allowing valid but rejected programs to run safely.
- Co-authored a paper on the framework, currently under submission to **EuroSys 2027**.

CRFS: A Crash-Resilient File System for egos-2k | [GitHub](#)

Feb 2026 – Apr 2026

Role: *Developer*, Supervisor: [Prof. Cheng Tan](#)

- This work extends the [egos-2k](#) operating system with a full set of file system APIs and two complementary approaches to file system reliability.
- Implemented a file system checker that scans the on-disk structure at boot time to detect and repair inconsistencies such as orphaned inodes, corrupted bitmaps, and broken directory entries.
- Implemented write-ahead journaling to ensure atomic multi-block updates, so a crash mid-operation leaves the file system in a recoverable state.

LELANTE: A Framework for Script-Free Android Test Execution

Feb 2024 –

Role: *Co-Developer*, Supervisor: [Prof. Anindya Iqbal](#) at BUET and [Kallol Kumar Pal](#) at Samsung Research, Bangladesh

- Built an automation agentic framework that executes natural-language test cases directly on Android devices, removing the need for hand-written Appium scripts and their maintenance overhead as UIs evolve.
- Designed an iterative action-generation pipeline using LLM reasoning to plan and execute GUI actions in a closed loop against live device state.
- Applied model distillation to reduce inference cost and improve scalability, achieving a 73% execution success rate across 390 test cases spanning 10 Android applications.
- Co-authored a paper on this work, published at **EASE 2025**.

QLox: An Optimized Compiler for the Lox Programming Language

Aug 2026 – Present

Role: *Independent Developer*

- Building a bytecode compiler for the Lox programming language, compiling source code to bytecode and executing it on a custom stack-based VM.
- Implementing a mark-and-sweep garbage collector to manage memory for the VM's heap-allocated objects.
- Plans to support core language features including closures, classes with single inheritance, and first-class functions.

QCC: A Tiny Compiler for the C Programming Language | [GitHub](#)

Jul 2024 – Nov 2024

Role: *Independent Developer*

- Built a compiler for a subset of C that generates 8086 assembly code, using Flex and YACC.
- Handwrote a lexer and an SLR(1) parser in C++ for the full ANSI C grammar as standalone tools.

Virtual Memory Management for xv6

Dec 2022 – Dec 2022

Role: *Developer*, Supervisor: [Prof. Tahmid Hasan](#)

- Extended the [xv6](#) operating system with page table-based virtual memory support.
- Implemented copy-on-write fork and page swapping.

Custom 4-Bit MIPS Microarchitecture and Hardware Implementation | [GitHub](#)

Sep 2022 – Dec 2022

Role: *Project Lead*, Supervisor: [Prof. Tahmid Hasan](#)

- Designed a custom 4-bit MIPS-lite ISA with modified opcode and register allocations
- Built the datapath in Logisim to simulate instruction execution.
- Built a physical hardware prototype to emulate the CPU's control logic.
- Wrote an assembler that converts assembly code into raw binary machine code.
- Tested arithmetic operations and control flow by executing the assembled binary programs on the CPU.

ACTIVITIES

Student Volunteer, VLDB 2026

2026

Competitive Programmer, National Programming Contests, BUET CSE

2020 – 2024

Director of Technology, BUET Cyber-Security Club

2023 – 2024