

# Sheharyar Khalid

sheharyar@virginia.edu | LinkedIn: [sheharyar-khalid](#) | GitHub: [Sheharyar-Khalid](#) | +1 (434) 760-7643

## EDUCATION

**University of Virginia (UVA)**— *Masters in Computer Science* - **GPA: 4.0/4.0** Aug 2023 – May 2025

- **Coursework:** Computer Networks, Economics of Distributed Systems, Graph ML, HCI, Machine Learning, Network Security

**Lahore University of Management Science (LUMS)** — *Bachelors in Computer Science* Aug 2019 – June 2023

- **Coursework:** Algorithms, AI, Data Structures, Data Science, Object Oriented Programming, OS, Software Engineering

## SKILLS AND EXPERTISE

**Programming Languages:** C, C++, Python, Javascript, HTML, CSS, Ruby (basic), Haskell, Go, Bash, MATLAB

**Frameworks:** Jupyter, Numpy, Pandas, Scikit-learn, Keras, PyTorch, Ruby on Rails (basic) Tensorflow, Express, NodeJS, ReactJS

**Tools & Other:** Jira, MongoDB, SQL, Redis, AWS, Git, Docker, Elasticsearch, Automated Testing, JSON, Spark, Agile, CI/CD, Splunk

**Soft Skills:** Strategic Thinking, Problem Solving, Team Work, Presentation and Technical Writing, Communication

## WORK EXPERIENCE

**Research Software Engineer - LAVA Lab (UVA)** Mar 2024 – Present

- Led a cross-functional team to design and implement a high-performance C++ engine for graph matching on streaming log data, achieving sub-millisecond inference times and reducing attacker dwell time by **99%** while managing **100 Gbps** network traffic.
- Developed a novel bloom-filter-based engine in C++ leveraging hashmaps (Boost), parallel processing, and garbage collection for high performance and minimal memory overhead.

**Research Software Engineer - LAVA Lab (UVA)** Aug 2023 – Mar 2024

- Implemented FPGA-based hardware acceleration for SIEM (Elasticsearch, Splunk) workloads for optimizing compute-intensive operations through hardware modules (Pipelined Architectures, Parallel Processing) achieving **400%** throughput improvement.
- Conducted rigorous performance profiling and bottleneck analysis of Elasticsearch (Python, Bash scripting, Docker) to automate the setup and testing processes.
- Improved detection latency to under 1 millisecond, saving enterprises millions of dollars in potential losses.

**Software Developer — Internet Security and Privacy Lab** May 2022 – Aug 2023

- Developed a distributed real-time graph-matching engine for streaming log data using Wazuh, Kibana, and Logstash achieving a **30% improvement** in graph-building and matching performance.
- Used Docker for containerization to ensure modularity, scalability, and fault tolerance and Logstash and Kibana to preprocess and enrich log data and visualize graph-matching results.

## SELECTED PROJECTS

**Community Discussion Social Forum | Node, React, MongoDB, AWS, Express, Selenium, Postman** [Github Link](#)

- Built a community discussion and social forum for LUMS to connect peers, organize events, submit course requests, discuss topics, and buy/sell items on an integrated marketplace.
- Implemented JWT, and developed REST APIs with Express.js and MongoDB, ensuring efficient backend operations.
- Designed and tested APIs using Postman and conducted end-to-end testing using Selenium.

**Video Conferencing Using L4S | TCP/IP, Linux, Open vSwitches, L4S, Video Conferencing** [Github Link](#)

- Implemented and evaluated L4S technology (Linux, TCP/IP) to optimize video conferencing performance across real-world network conditions achieving a **10x** reduction in latency and a **5x** decrease in jitter.
- Deployed a network testbed (Linux network namespaces, Open vSwitches) to evaluate L4S performance across multiple concurrent connections and analyzed key metrics including latency, throughput, and jitter.

**Explainable Graph-Based Host Intrusion Detection | GNNs, GNNExplainer, MITRE ATT&CK, LLMs**

- Implemented GNNExplainer by enhancing the IDS data pipeline for graph data, and adapting ML models for compatibility.
- Mapped detected intrusion patterns to MITRE ATT&CK tactics and techniques using Large Language Models finetuned on prior security data, improving alert clarity by **30%** and overall efficiency by **20%**.