

# Soumyajit Karforma

☎ +1 (289)-892-4623 ✉ [skarform@uwaterloo.ca](mailto:skarform@uwaterloo.ca) 🔗 [linkedin.com/in/soumyajitkarforma](https://www.linkedin.com/in/soumyajitkarforma) 🐙 [github.com/SoumyajitProjects](https://github.com/SoumyajitProjects)

## Education

### University of Waterloo

September 2021 – April 2026

*B.A.Sc. in Computer Engineering (Hons.)*

*Waterloo, ON*

- **Relevant Coursework:** Data Structures & Algorithms, Operating Systems, Systems Programming & Concurrency, Compilers, Networks, Databases, Embedded Microprocessor Systems, Discrete Math, Probability & Statistics

## Technical Skills

**Languages:** Python, Java, C/C++, TypeScript, SQL, Lua

**Libraries/Frameworks:** NumPy, Pandas, FastAPI, Node, React, TensorFlow, OpenCV, ROS2

**Tools/Technologies:** AWS, PostgreSQL, MongoDB, Ansible, Jenkins, Kubernetes, Docker, Splunk

## Experience

### Sun Life Financial Inc.

May 2025 – Present

*Software Engineering Intern*

*Toronto, ON*

- Optimized a production-grade LLM chatbot using AWS Bedrock (Claude 3.5), Lambda, and pgvector-enabled PostgreSQL to deliver support across **10+** internal systems using retrieval-augmented generation (RAG).
- Designed an S3 ingestion pipeline with Python and AWS Glue to extract documentation from PDF/HTML Confluence pages, **improving LLM answer accuracy by 30%**
- Engineered **20+** prompt variations for frequently accessed APIs and evaluated their responses against existing Confluence documentation, resulting in a **22% improvement** in retrieval consistency compared to the baseline prompts
- Implemented Splunk dashboards with custom SPL queries to monitor RAG chatbot metrics—query volume, latency, and error rates—enabling proactive incident detection and reducing troubleshooting time by **40%**

### Pearl Sullivan Engineering Ideas Clinic

January 2025 – April 2025

*Robotics Software Engineering Intern*

*Waterloo, ON*

- Built an AI navigation system for Automated ground vehicles using ROS2, Python, and OpenCV that achieved **90% collision avoidance** in dynamic environments
- Implemented stop sign detection using OpenCV and trained the model on **20+ hours** of real-world driving footage, improving rule-based compliance by **95%** in simulation
- Designed and containerized automated ground vehicle software environment with Docker, reducing setup errors by **80%** and enabling consistent multi-platform testing

### Sun Life Financial Inc.

January 2024 – April 2024

*Software Engineering Intern*

*Toronto, ON*

- Developed an automated incident notification pipeline using Microsoft Teams APIs, Ansible, YAML, and PowerShell to deliver over **100 real-time alerts daily**, improving response time and visibility for infrastructure incidents.
- Containerized patching and deployment workflows using Docker and integrated them with Kubernetes and Jenkins CI/CD pipelines, reducing rollout failures by **40%** and accelerating deployment times by **2x**
- Automated OS patching on **30+** MongoDB servers using Ansible and PostgreSQL, decreasing setup time from over **2 hours to under 30 minutes** per server by removing manual interventions

### Wind River Systems

September 2022 – December 2022

*Software Engineering Intern*

*Alameda, CA*

- Replaced Python with Lua for runtime scripting in VxWorks RTOS, reducing system response latency by **30%** due to lower memory overhead and faster interpreter performance
- Fine-tuned real-time scheduling logic for VxWorks, minimizing task starvation and improving high-priority execution throughput by **15%** in baseline simulations
- Redesigned the C-based test suite for VxWorks by introducing parallel test execution and removing redundant validations, reducing build times by **30%** and memory usage by **25%** compared to the original version

### Pearl Sullivan Engineering Ideas Clinic

January 2022 – April 2022

*Software Engineering Intern*

*Waterloo, ON*

- Designed TensorFlow and PyTorch inference pipelines for autonomous vehicle perception, increasing object recognition accuracy by **20%** and reducing latency by **35%** compared to the initial model
- Built an Ubuntu-based simulation environment to emulate NVIDIA Jetson Nano using virtual sensors, reducing hardware usage during prototyping by **50%**