Md. Saidur Rahman (Pulok)

Dhaka, Bangladesh saidur.pulok@gmail.com +880 1621 299536
linkedin.com/in/saidurpulok github.com/saidurpulok saidurpulok.github.io

Research Interests

Efficient LLM inference and serving, cost-aware model routing, resource-aware ML systems, and adaptive inference-time compute.

I am particularly interested in scheduling, memory and compute management, heterogeneous execution, and deploying learning systems under resource constraints, including embodied / physical AI.

Education

Bangladesh University of Engineering and Technology (BUET) **2019–2024**
B.Sc. in Mechanical Engineering *Dhaka, Bangladesh*

- **Undergraduate Thesis:** *Integrating Machine Learning for Enhanced Detection and Localization of Gas Pipeline Leaks.*
- **Human-Following Shopping Cart:** Built a cart that recognizes its owner's face and follows them in any direction, using an ESP32 camera module, Arduino motor control and ultrasonic distance sensing.
- **Heat Exchanger Design:** Designed and fabricated a shell-and-tube heat exchanger, performing thermal design calculations and validating the design using HTRI.

GRE (2025): Quantitative 165 | Verbal 151 | Analytical Writing 4.0

Research and Systems Projects

AxiomServe: Experimental LLM Serving Runtime **2026–Present**
Independent ML Systems Project

- Built a minimal LLM serving runtime for Qwen2.5-0.5B-Instruct on an NVIDIA RTX 5070 Laptop GPU, implementing explicit batched prompt prefill and token-by-token decode without relying on `model.generate()`.
- Implemented per-request KV-cache reuse, sequential and static-batched execution, continuous batching, request admission, FCFS and shortest-prompt-first scheduling, streaming, cancellation, and bounded resource controls.
- Built reproducible GPU benchmarks and observability for throughput, time-to-first-token, batch occupancy, cache usage, and request-level latency.
- On a 192-request workload, continuous batching cut p95 time-to-first-token to 119 ms, versus 503 ms for static batching and 3.48 s sequential, while static batching kept the highest throughput.

routewise: A Cost–Quality Benchmark for LLM Routing **2026**
Independent ML Research Project

- Built an offline benchmark of learned LLM routers on per-question HELM results (8.5k prompts, 22 models), where each router estimates every model's probability of answering correctly and picks the model that maximizes expected accuracy minus a cost penalty.
- Compared task-aware, kNN and logistic-regression routers using 5-fold cross-validation stratified by task, 95% bootstrap confidence intervals, best two-model mixes and an oracle as reference points, and corrected for output tokens that HELM fails to record.
- At equal accuracy, routing cost 48% less than the best cheap model (DeepSeek-V3) on 2024 models and 25% less than the best model (Qwen3) on 2025 models, and beat every single model by 3–4.5 points.

VeriSearch: Verifier-Guided Reasoning Under Fixed Inference Budgets **2026–Present**
Independent ML Research Project

- Found that strong verifier performance came largely from answer-format shortcuts: a trivial answer-presence classifier matched its accuracy, and its AUROC fell to 0.47 on traces with extracted answers.
- Built an end-to-end GSM8K reasoning evaluation framework using Qwen2.5-Math-1.5B-Instruct with stochastic candidate generation, self-consistency, learned verifier reranking, adaptive early stopping, and fixed generated-token budgets.
- Fine-tuned a DistilRoBERTa verifier using leakage-safe problem-level splits and evaluated ranking quality, calibration, AUROC, AUPRC, Brier score, bootstrap confidence intervals, and compute-normalized policy performance.
- On a 48-problem held-out pilot, adaptive stopping used 35% fewer generated tokens than full verifier reranking for about 1 point lower accuracy.

Publications

M. S. Rahman, F. Khandoker, S. S. Arnov, and M. A. Islam. "Integrating Machine Learning for Enhanced Detection and Localization of Gas Pipeline Leaks." *IUT International Conference on Core Engineering and Technology (ICCET)*, 2024.

Industry Experience

Machine Learning Engineer I, Optimizely **Apr 2026–Present**
Dhaka, Bangladesh

- Designed a 46-task, 13-category benchmark (medium / hard) to evaluate Optimizely's AI desktop app, Mark (formerly Opal), against third-party AI tools across real work tasks.
- Built an on-device LiteLLM routing proxy whose decision model sends each request to a model sized for the task, cutting spend from defaulting to expensive models; used daily by the team.
- Designed an Argo Workflows ML pipeline on EKS and productionized 2 ML pipelines end to end on Snowflake data; experiments tracked in MLflow and SageMaker.
- Ran QLoRA fine-tuning R&D on Qwen2.5 (0.5B / 1.5B) for customer-data recall: good on facts, weak on numbers.

Machine Learning Engineer, ACI PLC **Aug 2025–Apr 2026**
Dhaka, Bangladesh

- Productionized and deployed an on-premise face-recognition attendance system replacing fingerprint check-in for 50+ employees: FastAPI and Redis Streams pipeline over RTSP cameras, InsightFace on ONNX / CUDA, Qdrant.
- Built and launched AgroGPT, a RAG platform for precision agriculture used by farmers, agri-businesses, and researchers: Airflow pipelines for weather and market data, FastAPI services, Prometheus / Grafana monitoring.

Jr. Software Engineer, Brick Line Technology Ltd. **Mar 2025–Jul 2025**
Dhaka, Bangladesh

- Built Python backend services for Zeda, a US clinical trial management system (CTMS) used by several US clinics: FastAPI, AWS RDS, DynamoDB, S3, Lambda.
- Implemented MFA with AWS Cognito and Twilio, and shipped several Generative AI features.

Technical Skills

Programming:	Python, C/C++, TypeScript, Git
Machine Learning:	PyTorch, TensorFlow, Scikit-learn, Hugging Face Transformers, OpenCV, ONNX
Systems / Backend:	Linux, FastAPI, Docker, Redis, PostgreSQL, Qdrant, REST APIs
Cloud / Infrastructure:	AWS, GCP, Kubernetes / EKS, Snowflake, GitHub Actions
Research / Evaluation:	Experimental design, reproducible benchmarking, bootstrap confidence intervals, calibration analysis, performance analysis, ANSYS simulation