

MD. FAHIM ISLAM

☎ (+880) 1516-566133 ✉ mrislam.fahim@gmail.com
🌐 fahims-portfolio.vercel.app/  fahim  fahim
🏠 Jatrabari, Dhaka-1236, Bangladesh

SUMMARY

AI engineer with hands-on experience building LLM applications, RAG pipelines, agent-based workflows, and conversational AI systems. Worked across retrieval, re-ranking, model integration, evaluation, API development, and streaming interfaces using Python, PyTorch, Hugging Face Transformers, LangChain, LangGraph, FAISS, Ollama, FastAPI, and Next.js. Comfortable taking AI features from model and retrieval logic to backend services and user-facing applications.

EDUCATION

North South University

Bachelor of Science in Computer Science and Engineering, CGPA: 3.52

Sep '21 – May '26

Dhaka, Bangladesh

EXPERIENCE

Machine Intelligence Lab, North South University

Research Assistant (LLMs & Representation Learning)

May '25 – Dec '25

Dhaka, Bangladesh

- Designed hierarchy-aware text representation methods using hyperbolic (Poincaré ball) embeddings, learned token-attention pooling, and multi-segment refinement to support coarse-to-fine semantic retrieval.
- Built a conversational ADHD screening-support system that tracks symptom coverage across dialogue and uses a multi-model pipeline to extract evidence and produce structured assessment outputs.

SKILLS

- **Programming Languages:** Python, JavaScript
- **AI & ML Tools:** PyTorch, Scikit-learn, HuggingFace Transformers, RAG, LLM Fine-tuning, Pruning, Prompt Engineering, LangChain, LangChain.js, LLamaIndex, Ollama
- **Web Development:** MERN Stack, Next.js, FastAPI
- **Databases:** MySQL, MongoDB, Convex
- **Other Expertise:** Git, REST API Development, Data Analysis

PROJECTS

PhyRAG: Offline RAG-Powered Physics Learning Assistant | Python, LangChain, FAISS, Sentence Transformers, Ollama, FastAPI, Next.js

- Built an offline RAG system over a secondary-level physics textbook using two dense retrieval paths, FAISS indexing, candidate merging, cross-encoder re-ranking, relevance filtering, follow-up question reformulation, and local Gemma 2 inference through Ollama.
- Evaluated retrieval and generation on approximately 4,000 textbook-derived question-answer pairs with RAGChecker; mathematical questions reached 0.86 F1 and 0.82 faithfulness in the project's evaluation. Added manual testing with 12 students and documented failure cases including incomplete retrieval and long-conversation degradation.
- Served the RAG pipeline through FastAPI and built a Next.js chat client with per-session conversation state and Server-Sent Events for streamed responses.
- GitHub Link:  [PhyRAG](#)

QuestGen-AI Agent: AI-Powered Exam Question Generator | TypeScript, Next.js 15, LangChain, LangGraph, Convex, PDF.js/pdf-parse, Server Side Event, OpenRouter

- Built a PDF-grounded exam-generation application that accepts source material and user-defined marks, difficulty, and question types, then produces structured MCQ, true/false, short-answer, and essay questions.
- Split generation into five specialized stages: Extractor, Creator, Analysis, Decider, and Formatter so requirements, question generation, quality checks, revision decisions, and final formatting are handled separately.
- Implemented PDF extraction, Convex-backed file storage, OpenRouter model access, institutional header configuration, Markdown output, and real-time Server-Sent Events streaming.
- GitHub Link:  [QuestGen-AI Agent](#)

Hierarchical Reasoning Embedding Model with Hyperbolic Setup | Python, PyTorch, Transformers, Hyperbolic Embeddings, Contrastive Learning

- Built a hierarchy-aware text embedding model that represents each input at multiple semantic levels, from coarse intent to fine-grained meaning, using a frozen Transformer backbone, learned token-attention pooling, and iterative hierarchical refinement.
- Mapped each refinement level into Poincaré space with progressively larger radial scales and trained the representations with a coarse-to-fine hyperbolic NCE objective, allowing semantic granularity to be encoded directly in the embedding geometry.
- Implemented hyperbolic-distance retrieval and an evaluation pipeline supporting 11 MTEB retrieval tasks with Hits@K, Recall@K, NDCG, MRR, and MAP, along with a backbone-only cosine-similarity mode for comparison.
- GitHub Link:  [Hierarchical Reasoning Embedding Model with Hyperbolic Setup](#)

PANDA: Personalized ADHD Neuro Diagnostic Assistant | Next.js, Convex, OpenAI (GPT-4.1, STT/TTS), LLM Evaluation, Conversational AI

- Built a text-and-voice conversational screening-support application that tracks which ADHD symptom indicators have been covered rather than relying on a fixed questionnaire flow.
- Implemented post-session evidence extraction and structured domain-level scoring from conversation transcripts; on a 10-sample evaluation set, reported item alignment of 83.33–94.44%, question coverage of 78.79–85.71%, and response mapping accuracy of 82–86%.
- GitHub Link:  [PANDA](#)

ClearCut Pruning: Efficient Weight Refinement for LLM Compression | Python, PyTorch, Transformers, LLM Compression, N:M Structured Sparsity, Low-Rank SVD

- Implemented a post-training LLM pruning toolkit supporting magnitude, Wanda, SparseGPT, RIA, and an activation-aware ClearCut importance criterion, with both unstructured and hardware-oriented N:M sparsity.
- Added channel reallocation and SVD-based low-rank decomposition for attention projections, with perplexity and zero-shot evaluation across WikiText-2, C4, PTB, BoolQ, RTE, HellaSwag, ARC-Challenge, and MNLI.
- At 50% sparsity on WikiText-2, project experiments reported ClearCut perplexity of 6.78 on LLaMA-2-7B and 11.44 on OPT-6.7B, lower than the SparseGPT, Wanda, and RIA baselines reported in the same experiments.
- GitHub Link:  [ClearCut Pruning for LLM Compression](#)

Traffic Accident Outcome Prediction | Python, Logistic Regression, SVM, Random Forest, XGBoost

- Built a full machine learning pipeline to analyze road traffic accidents in Rawalpindi, Punjab, targeting both injury severity (*Injury Type*) and post-accident outcomes (*Patient Status*).
- Trained and compared classical and ensemble classifiers (Logistic Regression, Decision Tree, SVM, Random Forest, AdaBoost, XGBoost) using Accuracy, Precision, Recall, F1, and confusion matrices; produced diagnostic learning curves for key hyperparameters (solver/max_iter, max_depth, kernel).
- GitHub Link:  [Traffic Accident Analysis](#)

EXTRACURRICULAR

- Represented Machine Intelligence Lab at Bear Summit and National Semiconductor Symposium 2025.
- Participated in the AI, IoT & Robotics Day 2025 seminar at North South University.
- General Member, NSU ACM Student Chapter.

INTERESTS

AI Engineering:

- Building LLM applications with retrieval, tools, and structured workflows.
- RAG pipelines with retrieval, re-ranking, grounding, and evaluation.
- Agent workflows for multi-step AI applications.
- Efficient local LLM inference and model compression.
- Multimodal AI applications using text, vision, and audio.