Tarek Shiblee Shawkat

sstarek93@gmail.com | 070-7526-5128 | linkedin.com/in/sstarek | sstarek.xyz | Yokohama, Japan

PROFESSIONAL SUMMARY

AI/ML and AIOps Engineer with 7+ years of experience building production-grade machine learning systems, from data pipelines and model development to agentic AI and cloud-native deployment. Experienced working within cross-functional teams to translate research concepts into reliable, real-world AI solutions.

TECHNICAL SKILLS

Programming Languages: Python, SQL, C

ML Frameworks: TensorFlow, Keras, PyTorch, scikit-learn, XGBoost

NLP: BERT, Word2Vec, NLTK

Generative AI & AI Agents: LLM Integration, Hugging Face Transformers, LangChain, MCP/FastMCP

Data Processing: Pandas, NumPy, Dask

Visualization: Matplotlib, Seaborn, Power BI

Web Frameworks & APIs: FastAPI, Flask

ML Infrastructure: Docker, GitHub Actions (CI/CD), MLflow, Kubernetes, Terraform (IaC), AWS (S3, EC2, Bedrock, Lambda, Step Functions, CloudWatch, OpenSearch, S3 Vectors), OpenTelemetry

Others: ChromaDB, Librosa, OpenCV, MySQL

PROFESSIONAL EXPERIENCE

Woven by Toyota (via Robert Half) - Tokyo, Japan

AI/ML & AIOps Engineer (Contractor) | Feb 2026 - Present

Designed and deployed production-grade, event-driven AIOps systems on AWS - owned the AI roadmap; identifying operational pain points across cloud and network operations, and replacing manual workflows with autonomous AI pipelines

Technologies: Python, NumPy, Pandas, statsmodels, AWS, ChromaDB, scikit-learn, Docker, LangChain, LangGraph, FastMCP, FastAPI, Terraform, GitHub Actions, Kubernetes

AI-Driven Root Cause Analysis (RCA)

- Designed a multi-step Lambda pipeline orchestrated by AWS Step Functions to automatically analyze AWS incidents reported in Slack - validating incoming requests, retrieving and analyzing CloudWatch logs with an LLM
- Engineered chained LLM calls via Bedrock to parse incident text into structured data and analyze CloudWatch logs, returning a root cause report with confidence scoring; addressed hallucination and formatting issues through prompt iteration and regex-based validation
- Designed and tested log deduplication using TF-IDF and clustering: reduced input token consumption by up to 96% with no loss in analysis quality (20,000 tokens to 853 tokens on a real test case)
- Fully deployed to production on AWS and integrated with Slack - incidents are automatically analyzed and root cause posted to the originating Slack thread in real time
- Extended the RCA pipeline into a fully agentic system using LangChain and LangGraph; enabling autonomous tool selection, stateful orchestration, and dynamic MCP tool binding for flexible incident investigation
- Designed and deployed the agentic system as two production microservices, a FastAPI inference layer and a custom MCP tool server, containerized with Docker and orchestrated with Kubernetes, with prompt injection safeguards implemented at the MCP layer

AI-Driven Alert Refinement

- Designed and deployed an AI-powered alert classification system that replaced manual review of ~30 daily Slack alerts with an automated triage pipeline, classifying each alert by severity - Critical, Warn, Info

- Designed a tiered classification architecture combining nearest-neighbor retrieval, regex pattern matching, and RAG-based LLM escalation based on confidence scoring - with a fail-safe default to Critical
- Built a text normalization pipeline to clean raw alert text before embedding, removing noise that would otherwise degrade retrieval quality
- Generated alert embeddings via Amazon Bedrock and indexed them into ChromaDB locally and S3 Vectors for AWS deployment, migrating from Amazon OpenSearch
- Achieved 97% accuracy, 0.89 macro F1 score, and 100% recall on Critical alerts on the test set
- Deployed the inference system on AWS Lambda; provisioned infrastructure with Terraform and automated testing and deployment via GitHub Actions CI/CD pipeline
- Configured CloudWatch error-rate alarms on deployed Lambda functions to monitor inference pipeline health in production; automatically triggering alerts when error thresholds are breached, with notifications routed via SNS

SLO Violation Prediction

- Designed a daily forecasting pipeline using Holt-Winters with weekly seasonality to project month-end SLO (service level objective) violation risk
- Implemented Monte Carlo confidence scoring to estimate the probability of month-end SLO breach, providing engineers with a risk signal alongside each prediction
- Validated on synthetic data modeled after real production patterns - 100% classification accuracy, average violation lead-time of 19 days, zero persistent false alarms
- Containerized in Docker with a model-agnostic architecture, allowing the forecasting component to be upgraded without redesigning the pipeline

Career Break

Upskilling & Job Search | May 2025 - Jan 2026

Following company-wide restructuring, focused on active job search alongside independent upskilling. Built and published a hands-on LLM uncertainty quantification project on GitHub, exploring confidence calibration and error detection signals across math reasoning and factual QA benchmarks

Hiperdyne Corporation - Tokyo, Japan

AI Engineer (Full Time) | Nov 2018 - Apr 2025

Project: *Deep12 - Music Analysis AI, Sony Computer Science Laboratories*

Contributed to R&D and developed AI-powered music analysis functionalities across search, detection, and predictive tasks, deployed as a service in the Sony Music Publishing library. Mentored new hires on project architecture, ML workflows, and engineering best practices

Technologies: Python, NumPy, Pandas, PyTorch, TensorFlow, Keras, scikit-learn, Librosa, BERT, SentenceTransformers, BERTopic, XGBoost, Docker, MLflow, AWS, GitHub Actions, pytest, NLTK, Flask, Git

ML & Model Development

- Conducted time-series analysis and applied signal processing techniques (fundamental-frequency analysis, noise filtering) for feature extraction; performed feature engineering and selection to design task-specific inputs and improve model effectiveness
- Designed and trained CNN, LSTM, and Transformer-based models, implementing architecture from academic papers after literature review, and achieved results comparable to published benchmarks
- Applied traditional ML models, including XGBoost and One-vs-Rest, to non-time-series data, utilizing ensemble learning techniques and achieving 75-82% accuracy across classification tasks
- Optimized large-scale data processing pipelines using GPU-accelerated PyTorch, implementing parallelized execution and improving runtime by 4-5x (75-80%)
- Addressed dataset imbalances through targeted augmentation strategies including synthetic data generation and resampling, improving model robustness and generalization

Generative AI & NLP

- Built a multi-stage NLP pipeline for text classification, progressing from TF-IDF & Naive Bayes to Word2Vec & Random Forest to fine-tuned BERT, moving toward deeper contextual understanding at each stage
- Leveraged AWS Bedrock to generate synthetic text for minority classes, applying few-shot prompting, cosine similarity deduplication, BERTopic-based off-theme filtering, and retry logic with checkpointing for resilient generation at scale
- Benchmarked all pipeline stages against the augmented dataset - fine-tuned BERT outperformed the next-best model by ~8%

MLOps & Infrastructure

- Containerized training and inference pipelines using Docker to ensure reproducibility and stable experimentation environments
- Designed and maintained CI pipelines using GitHub Actions, running inside project Docker containers for environment consistency, with automated unit testing and model validation checks on every pull request - improving code reliability and catching regressions early across the team
- Utilized AWS EC2 (G4dn instances) and S3 to provision scalable compute and storage resources for model training, experimentation, and benchmarking
- Tracked experiments, hyperparameters, and evaluation metrics using MLflow, and managed model versioning and lifecycle promotion through the MLflow Model Registry

EDUCATION

Bachelor of Science in Computer Science and Engineering

Ahsanullah University of Science and Technology | Dhaka, Bangladesh | 2017

LANGUAGES

English: Native level/Bilingual

Bengali: Native

Japanese: Daily conversation