Sepehr Kazemian

(587) 568-7994
Sepehr.kazemian@gmail.com
Toronto – Canada
linkedin.com/in/sepehrkazemian
github.com/SepehrKazemian

AI-focused Software Engineer with 7+ years of experience building automation systems using LLMs, Azure AI, and custom model deployments. Skilled in designing intelligent workflows, integrating RESTful APIs and third-party AI services, and enabling predictive, autonomous solutions. Strong background in developing data pipelines, microservices, and scalable cloud-native architectures. Experienced in translating business needs into user-centric automation with RAG, LangChain, and Hugging Face tools. Passionate about driving toward agentic AI through innovation, performance optimization, and cross-functional collaboration.

SKILLS

- **LLM Ecosystem:** OpenAI/Gemini API, LangChain, LlamaIndex, LangGraph, Retrieval-Augmented Generation (RAG), Vector DB, Embeddings, RAGAS, Lora, Transformers, Agents
- **ML Tools & Frameworks:** PyTorch, TensorFlow, scikit-learn, Boosting algorithms, Hugging Face, Classic NLP, Time-series modeling, Elasticsearch, MLflow
- **Cloud & Infrastructure:** GCP (Vertex AI, BigQuery, IAM & Security, GKE, Cloud Build, Cloud Run, Artifact Registry), Azure (AML, AKS, Functions, Cognitive Services, DevOps, Monitor), AWS (SageMaker, Lambda, S3, EC2, CloudWatch, IAM, API Gateway)
- **Data Engineering:** dbt, Spark, PostgreSQL, MySQL, BigQuery, Snowflake, Dagster, Pandas, ETL/ELT pipelines, Data Visualization, Databricks, Tableau
- **Software & DevOps:** Python, JavaScript, C, C++, Docker, Kubernetes, Github, Datadog, CI/CD Concepts, Azure DevOps
- **Soft Skills:** Cross-functional communication, end-to-end ownership, leadership, mentorship, analytical thinking, rapid learning, team collaboration, Translating technical ideas to non-technical audiences, Systems thinking, Creativity / innovative mindset
- **LLM Tools & Practices**: Langfuse, LLM-as-a-Judge, Chain-of-Thought Prompting, Hugging Face Inference API, Multi-query Expansion, Document Routing, Hallucination Detection
- **Model Deployment & Observability**: FastAPI, Azure App Service, Azure Functions, Redis Queue, Cron Jobs, Shadow Deployment, Human-in-the-Loop Pipelines
- **Explainable AI & Evaluation**: SHAP, Permutation Importance, RAGAS, Feedback-driven Online Training
- **Document Intelligence**: Tesseract, PyMuPDF, PDFMiner, Rule-based Regex NLP, OCR pipelines
- **Probabilistic Forecasting**: Quantile Regression, Prediction Intervals, Bayesian Neural Nets
- **Synthetic Data & Augmentation**: GANs for Imputation, Custom PII Data Generation
- **Advanced Testing**: Unit Tests, Integration Tests, Nested Cross-Validation, Backtesting
- **Graph & Vector Systems**: Knowledge Graphs, GraphDB, Qdrant, Azure Cognitive Search
- **Explainability Tools**: SHAP, Permutation Importance
- **Probabilistic Modeling**: Quantile Regression, Prediction Intervals, Bayesian NNs
- **Synthetic Data Generation**: GANs + custom PII synthesizer
- **Dashboarding UX**: Streamlit, Interactive forecasting/extraction interfaces
- **Design Patterns & Testing**: Shadow deployment, backtesting, nested cross-validation
- **Automation**: Cron Jobs, Airflow (orchestration), Dockerized training/inference pipelines
- **Version Control & Experiment Tracking**: MLflow, Hugging Face Trainer (for reproducibility)
- **Graph Databases**: Knowledge Graphs, GraphDB (used with LlamaIndex + routing)
- **Streaming & Queues**: Redis Queue (for real-time document pipelines)
- **Regex/NLP Tooling**: Regex engines for structured field extraction, rule-based NLP
- **PDF/Text Parsing**: Tesseract, PyMuPDF, PDFMiner — valuable for OCR/data extraction roles
- **Model Serving & APIs**: FastAPI (you've deployed ML/LLM APIs with it), Azure App Service, Azure Functions
- **Observability & Testing**: Langfuse, Datadog, Unit testing, Integration testing, Shadow deployment
- **Model Feedback Loops**: Human-in-the-loop pipelines, Online training automation (with FastAPI + Cron)
- **LLM Evaluation & Observability**: Langfuse, LLM-as-a-Judge, Chain-of-Thought prompting
- **LLM Deployment**: Hugging Face Inference API, FastAPI (for serving LLMs)
- **LLM Architectures**: Reranking, multi-query expansion, document routing, hallucination detection

- **Prompt Engineering** (you've clearly done it across multiple systems)

PROFESSIONAL EXPERIENCE

Munich Re

Feb 2022 - Present

Senior Applied AI Software Engineer (Reinsurance Industry) Toronto

LLM RAG Pipeline

Goal: To develop a scalable, production-grade Retrieval-Augmented Generation (RAG) system for insurance document summarization and question answering using LLMs, stacked retrievers, and intelligent routing across cloud infrastructure.

- Built a modular in-house RAG pipeline using LangChain, LlamaIndex, and LangGraph for QA, summarization, and retrieval tasks with LLM-as-a-Service capabilities.
- Parsed and embedded content from third-party data sources, storing it with metadata (e.g., page number, topic, chunk ID) in Qdrant, Azure Vector Search, and a graph database.
- Implemented stacked retrievers combining transformer models with FAISS for high-accuracy semantic search; supported multi-query expansion, context filtering, and deduplication to improve precision.
- Designed a dynamic document routing engine to send different content types to optimiz LLMs based on metadata or query type.
- Integrated Chain-of-Thought prompting and RAGAS for robust answer generation and automated output evaluation.
- Used LLM-as-a-judge techniques for hallucination detection, document ranking, and reranking answers based on relevance and groundedness.
- Developed feedback mechanisms for users to rate or correct outputs, which were logged and streamed into an online fine-tuning pipeline for model improvement.
- Deployed the full stack using Kubernetes, FastAPI, and Streamlit on Azure Cloud; served models using Hugging Face Inference endpoints and internal APIs.
- Enabled observability and traceability through Langfuse, Datadog, and Azure Monitor, capturing metrics like latency, success rates, retriever fallbacks, and human corrections.
- Supported batch and real-time ingestion, agent-based function routing, and scalable infrastructure for high-throughput workloads across multiple teams.

Project: Document Classifier

Goal: To build a fast and cost-efficient system for classifying 30 types of insurance and business documents using a layered model pipeline that balances performance, accuracy, and explainability in a real-time production environment.

- Fine-tuned an EfficientNet model on RVL-CDIP and domain-specific data to classify RGB-rendered document images with high speed and cost efficiency.
- Added a fallback RoBERTa model trained on OCRed text using Hugging Face Transformers and Azure OCR, activated only for low-confidence vision predictions.
- Applied a rule-based regex engine as the final layer to enforce classification consistency for edge cases and ambiguous inputs.
- Routed documents dynamically through a layered pipeline (vision → NLP → regex) based on model confidence thresholds.
- Achieved 97% classification accuracy in production across 30 document classes, with real-time and batch input support.
- Deployed the full system in Azure using FastAPI, Azure Blob Storage, App Service, Redis (for job queuing), and AKS for scalable serving.
- Developed a client-facing dashboard for document uploads, correction of OCR errors, and live model feedback, with support for manual review and retriggering classification.
- Enabled online training by collecting corrected labels and integrating them into periodic model updates using PyTorch, Hugging Face Trainer, and scheduled FastAPI Cron jobs.
- Designed the system for long-term maintainability, supporting continual learning, automated feedback loops, and adaptive model evolution based on user interaction.

Project: Impairment Extractor

Goal: Automating the extraction of key medical impairment fields from within insurance documents in various formats (PDF, scanned forms, Word) using a hybrid OCR and NLP pipeline, enabling accurate, real-time data capture for downstream analysis and business workflows.

- Built a robust NLP pipeline to extract structured fields from insurance documents across various formats, including OCR PDFs, digital PDFs, scanned forms, and Word files.
- Integrated Tesseract, PyMuPDF, PDFMiner, and a PDF library with bounding box support to extract text and spatial metadata from documents.
- Applied handcrafted regex rules for early-stage field extraction, later enhanced by a fine-tuned RoBERTa model trained to identify and label key fields with 96% accuracy.
- Developed a dashboard with file upload, text correction, and dynamic field reclassification; supported real-time and batch processing modes.
- Enabled correction of OCR errors through an editable UI, automatically retriggering regex or ML classification for updated outputs.
- Containerized the full pipeline with Docker and deployed on Azure Kubernetes Service (AKS) for scalable, fault-tolerant inference.
- Used Azure Redis for real-time job queuing and background task orchestration across uploads, corrections, and model inference steps.
- Leveraged Azure Blob Storage for managing file input/output, and Azure Functions to trigger processing pipelines upon user actions.
- Hosted the full-stack dashboard on Azure App Service, delivering a seamless real-time interface for business users and reviewers.
- Designed the system for modular extensibility, supporting both rule-based and learning-based field extraction with structured outputs for downstream usage.

Project: Pricing Analytics
Goal: Forecast future healthcare-related costs by modeling client purchasing behavior using historical claims data and economic indicators.

- Collaborated with cross-functional teams at a healthcare insurance group to build predictive models that forecast client spending on medications and services over future months.
- Designed and implemented time series forecasting models including ARIMA, SARIMA, LSTM, RNNs, rolling averages, and ensemble models for robust and adaptive predictions.
- Integrated external economic data (e.g., inflation, CPI, interest rates) via APIs and engineered time-aware features such as lag windows, seasonal indicators, holiday effects, and demographic factors.
- Built and maintained ETL pipelines using SQL, Airflow, Pandas, and dbt to clean, transform, and orchestrate large-scale healthcare data into modeling-ready formats.
- Visualized forecasts and SHAP-based model explainability in interactive dashboards using Power BI, Tableau, Streamlit, and Plotly Dash to support business decision-making and pricing strategy.
- Deployed models to production with automated retraining and monitoring using MLOps practices, including CI/CD (GitHub Actions), containerization (Docker), and cloud orchestration.
- Evaluated model performance using RMSE and R^2; delivered fully automated pipelines and dashboards used in real-time planning across the organization.

Project: Data Anonymization
Goal: A production-ready pipeline that anonymizes semi-structured medical JSON data by detecting and redacting personal identifiable information (PII), ensuring compliance, preserving structure integrity, and supporting both batch and real-time workflows.

- Engineered a recursive parser using BFS/DFS traversal to convert nested JSON into tree structures for flexible and fast inspection across variable schemas.
- Created a regex-based engine to detect and replace sensitive data in both field names and values, including names, addresses, birthdates, and device identifiers.
- Applied per-document randomized substitution to ensure anonymized consistency within each file, storing mappings for audit traceability.
- Trained and integrated a lightweight BERT-based classifier to identify context-dependent PII based on the path-to-value input, enabling accurate redaction beyond rule-based logic.
- Reconstructed anonymized documents into valid, well-formed JSON after transformation, preserving structure and format fidelity.

- Built a user-facing dashboard enabling both real-time and batch uploads for anonymization, offering fast performance and low-latency inference.
- Deployed the full pipeline on Azure, supporting both small-scale uploads and high-throughput batch processing of large document sets.
- Validated final outputs with automated and manual testing to ensure accuracy and JSON integrity across edge cases.
- Collaborated directly with stakeholders and internal clients to refine PII detection logic, compliance requirements, and product usability.

AltaML **May 2020 – Feb 2022**

Machine Learning Engineer (Startup FinTech Industry) Toronto

- Led multiple ML and data science initiatives across retail and fintech domains, including inventory optimization, financial forecasting, data imputation, and investment modeling.
- Designed and deployed an inventory optimization system to minimize cost, stockout rates, and overstock waste, accounting for product expiration, lead times, and seasonal trends across regional branches.
- Built probabilistic forecasting models using XGBoost, Bayesian neural networks, and quantile regression; incorporated prediction intervals and ensemble confidence bounds.
- Implemented shadow deployment to compare model recommendations against business-as-usual decisions, improving ordering efficiency with measurable KPIs.
- Collaborated closely with C-level stakeholders and domain experts to define relevant features and success metrics; communicated technical insights through interactive dashboards and executive-level reports.
- Built ETL workflows and cleaned noisy SQL + NoSQL datasets (MongoDB) for time series and transactional inputs, integrated Airflow for pipeline orchestration.
- Conducted feature selection and model explainability using SHAP and permutation importance techniques to improve transparency and stakeholder trust.
- Developed GAN-based data imputation pipelines to reconstruct missing segments of financial records for clients with corrupted or lost databases.
- Delivered long-term asset performance modeling and trend classification using explainable ML methods on financial time series, focusing on regression, risk analysis, and directional prediction.
- Applied nested time-series cross-validation and backtesting strategies to simulate realistic financial scenarios and validate model robustness.
- Used Docker to containerize pipelines; integrated MLflow for experiment tracking and Azure AutoML/H2O.ai for benchmark comparison and AutoML exploration.
- Built interactive dashboards for model insights and forecasting, supporting custom time horizon selection, lag adjustment, and confidence output overlays.
- Developed comprehensive unit and integration test suites across all production models and data pipelines to ensure reliability and reproducibility.

Telus **May 2018 – May 2019**

Software Engineer (Telecommunication Industry) Toronto

- Designed and implemented end-to-end automation tools for hardware testing of routers and modems, reducing testing cycles from a full week to under a day with zero manual effort.
- Built decision-support systems using tree-based models and Bayesian heuristics to optimize channel and frequency test planning, minimizing exhaustive sweeps.
- Conducted signal strength modeling and noise analysis across frequency bands to identify weak points and environmental interference patterns.
- Automated firmware update processes across test devices, enabling consistent rollouts and validation during manufacturing and QA stages.
- Developed software for programmable router chipsets, integrating embedded logic to support custom configurations pre-deployment.

- Created scalable test orchestration workflows with robust logging, error detection, and support for concurrent device testing.
- Used CI/CD pipelines (e.g., GitHub Actions or Azure DevOps) to deploy and maintain testing tools in an efficient and trackable manner.
- Collaborated with cross-functional hardware and QA teams to translate engineering protocols into scalable testing systems.

Information & Communication Technology Center
April 2015 – April 2016

Software Engineer (Tech Company) Toronto

- Developed SDN applications using OpenFlow with Floodlight and ONIX controllers, programming in Java, Python, and C++ across testbed and hybrid production environments.
- Designed programmable routing policies for adaptive channel switching, frequency optimization, and bandwidth management based on live network conditions.
- Implemented QoS-aware traffic control modules to dynamically prioritize application traffic, enhancing video streaming and data transfer efficiency.
- Built real-time network monitoring tools to track latency, throughput, and utilization across programmable routers and dynamically trigger rerouting logic.
- Conducted experimental deployments simulating fault tolerance, dynamic failover, and load balancing strategies in controlled environments.
- Evaluated performance gains and configuration trade-offs across multiple controller frameworks, contributing to architectural decisions in network control logic.
- Focused on delivering reusable, modular code and test-driven workflows across a distributed control and data plane environment.

Freelance
April 2015 – April 2016

Software Engineer (Tech Company) Toronto

- Built and deployed custom full-stack web applications for small businesses, including scheduling, booking, and reservation systems tailored to specific business needs.
- Designed and maintained web servers and backend infrastructure using PHP, Flask, and Node.js, with both relational (MySQL) and non-relational (MongoDB) databases.
- Created secure user authentication systems with role-based access and integrated admin dashboards for content and operations management.
- Led end-to-end client engagements, including requirements gathering, system architecture, development, deployment, and post-delivery support.
- Independently managed project hosting, deployment, and debugging on self-managed servers and shared hosting environments.

EDUCATION

M.Sc. in Computer Science
2018 - 2020

- *University of Alberta* - GPA – 3.8/4
- Applied AI in Software Engineering

B.Sc. in Software Engineer
2012 – 2017

- Tehran Polyechnique – GPA – 3.8