HEMANTH KOPPAKA
SR. SOFTWARE ENGINEER- AI
LinkedIn | +1(978)237-4932 | hemanthdpk@gmail.com | Portfolio

PROFESSIONAL SUMMARY

Senior AI Engineer with 7+ years of experience building production-grade software systems across healthcare, financial services, and fintech domains. Specializes in designing and deploying agentic AI workflows for regulated enterprise environments, with deep expertise in retrieval-augmented generation, multi-agent orchestration, and human-in-the-loop systems that meet clinical and compliance requirements. Proven track record of translating complex business and regulatory constraints including CMS prior authorization mandates and HIPAA-aligned data governance into reliable, auditable AI pipelines operating at scale. Equally fluent in backend engineering, data infrastructure, and applied machine learning, with hands-on experience taking AI systems from prototype through production across cloud-native architectures. Known for building systems where AI autonomy and human oversight are deliberately balanced, with quality signals, override tracking, and compliance reporting embedded by design rather than added as afterthoughts.

TECHNICAL SKILLS

AI & GenAI: LangChain, LangGraph, Azure OpenAI, Azure AI Foundry, GPT-4, Amazon Bedrock, Hugging Face Transformers, RAG Pipelines, Multi-Agent Workflows, MCP Servers, Prompt Engineering, Langfuse, Pydantic Output Validation
ML & NLP: PyTorch, TensorFlow, Scikit-learn, spaCy, RoBERTa, SageMaker, Vertex AI, MLflow, Fraud Detection, Churn Prediction, Underwriting Models
Languages: Python, Java, SQL, PySpark, Scala, JavaScript, TypeScript
Backend & APIs: FastAPI, Django REST Framework, Flask, Spring Boot, GraphQL, Celery, AsyncIO, JWT/OAuth2, RBAC
Frontend: React.js (Redux, Tailwind CSS), Angular (NgRx), Vue.js, Next.js, WebSockets
Data Engineering: Apache Spark, Databricks, Delta Lake, Kafka, Airflow, NiFi, Hadoop, Hive, AWS Glue, ETL/ELT Pipelines
Databases & Search: PostgreSQL, MySQL, Oracle, MongoDB, DynamoDB, Aurora, Cassandra, Pinecone, FAISS, OpenSearch, Elasticsearch, Azure Cognitive Search
Caching & Messaging: Redis, RabbitMQ, Kafka, IBM MQ, AWS Kinesis, Azure Service Bus, AWS SQS/SNS
Cloud: AWS (EKS, Lambda, S3, RDS, SageMaker, Step Functions, Glue, API Gateway) · Azure (AKS, AI Foundry, ADLS Gen2, Key Vault, API Management) · GCP (GKE, Cloud Run, Cloud SQL)
DevOps & IaC: Docker, Kubernetes, Helm, Terraform, GitHub Actions, Jenkins, Azure DevOps, Harness, Trivy, Blue-Green & Canary Deployments
Observability & Security: Prometheus, Grafana, ELK Stack, OpenTelemetry, Langfuse, HIPAA, HITRUST, IAM, Secrets Manager, Azure Key Vault
BI & Visualization: Power BI, Looker, QuickSight, Recharts

Professional Experience

Client: Cigna, Hartford, CT **July 2024 - Present**

Senior Software Engineer - AI

Responsibilities:

- Developed a human-in-the-loop agentic prior authorization system using LangGraph StateGraph with typed state schemas across 8 nodes as intent classification, member context retrieval, clinical document extraction, policy grounding, LLM clinical reasoning, confidence-based routing, nurse review handoff, and audit output, with conditional edges routing cases between autonomous approval and nurse review based on confidence thresholds.
- Designed two parallel context streams feeding every prior auth recommendation which are live transactional data covering member eligibility, plan details, claims history, and prior authorization history consumed from governed FastAPI backend APIs, combined with clinical policy grounding retrieved from Azure AI Search RAG pipelines over coverage criteria documents, medical necessity guidelines, and plan benefit documents and both streams assembled in LangGraph state before any LLM reasoning occurred.
- Integrated Azure Document Intelligence to extract and normalize structured clinical facts from unstructured physician narrative notes, prior authorization forms, and supporting clinical attachments stored in Azure Blob Storage, converting

free-text medical documentation into structured state fields so the clinical reasoning node always operated on clean labeled data rather than raw PDF content.

- Implemented Pydantic-based structured output validation at two distinct layers, FastAPI input layer validating incoming prior auth submission structure before graph invocation, and LangGraph node layer validating LLM clinical reasoning responses against strict JSON schemas before state updates ensuring no unstructured or hallucinated data reached nurse review or downstream audit systems.
- Built Python FastAPI backend services exposing member eligibility verification, claims history, benefit coverage details, and prior authorization history as secure schema-validated APIs consumed by LangGraph member context nodes at runtime, with completeness and consistency validation before data passed into workflow state catching upstream admin data entry errors before they corrupted recommendations.
- Developed RAG pipelines over clinical policy documents and medical necessity guidelines using document chunking, embedding-based indexing, and hybrid retrieval in Azure AI Search, with plan-specific filters ensuring members only retrieved policies matching their specific plan.
- Implemented human-in-the-loop interrupt nodes suspending workflow state at clinical decision checkpoints a regulatory requirement under CMS prior authorization rules - packaging member context, retrieved policy content, confidence scores, and LLM reasoning into structured decision-ready handoffs for clinical nurse reviewers, with workflow resumption triggered by nurse approval or denial recorded back into state.
- Persisted active workflow state and audit-ready processing records in Azure SQL during initial pilot rollout across a limited nurse cohort, with architecture designed for Cosmos DB migration as concurrent suspended case volume and 72-hour CMS compliance windows demanded distributed state management at scale with permanent decision history, nurse override records, and compliance reporting remained in Azure SQL throughout.
- Tracked nurse override rates as a primary system quality signal cases where nurses disagreed with high-confidence recommendations surfaced RAG content staleness, upstream data quality issues, or clinical reasoning prompt weaknesses, feeding directly into retrieval quality evaluation and prompt refinement cycles before broader nurse cohort rollout.
- Implemented Redis caching for frequently accessed member eligibility and plan detail API responses, reducing redundant governed API calls across concurrent prior auth cases and improving workflow throughput during peak submission windows without compromising data freshness for clinical decisions.
- Integrated asynchronous workflow events using Azure Service Bus and Event Hubs so audit record writes, downstream reporting updates, and operational notifications ran completely decoupled from LangGraph orchestration preventing batch processing dependencies from blocking active clinical workflows during peak prior auth volumes.
- Secured all service interactions through Azure API Management with OAuth2 authentication, role-based access control separating nurse, admin, and system identities, rate limiting, API versioning, and full audit logging - meeting HIPAA-aligned enterprise compliance requirements across clinical nurse access, governed backend API consumption, and AI workflow interactions.
- Containerized backend and AI workflow services with Docker, deployed to AKS using Helm-based pipelines with autoscaling and controlled rollout strategies, with trace-level observability through Langfuse covering retrieval quality scores, LLM reasoning consistency, tool call behavior, confidence score distributions, and end-to-end workflow latency across production prior authorization cases.

Client: Hearst, New York, NY ***October 2023 – July 2024***

Senior Software Engineer

Responsibilities:

- Built scalable Python FastAPI and Django REST Framework backend services exposing ratings, issuer metadata, outlook changes, and portfolio metrics with strong request validation, pagination, and standardized error handling for high-volume financial datasets.

- Integrated SageMaker-hosted ML inference endpoints into FastAPI microservices, enabling real-time credit-risk scoring, risk flag generation, and model-driven outputs to be consumed through governed backend APIs inside underwriting and analytics applications.
- Designed PostgreSQL-centered data models with optimized schema design, indexing strategies, and complex SQL queries (CTEs, window functions, joins) to support issuer history lookups, portfolio drilldowns, and rating movement analysis at scale.
- Built asynchronous FastAPI services for compute-intensive financial simulations (rate shocks, spread movements) using NumPy and Pandas, enabling interactive dashboards to request backend computations without blocking core application flows.
- Integrated Redis caching and Elasticsearch to accelerate API response times, reduce database load on repeated lookup patterns, and support fast issuer and instrument discovery across large financial datasets.
- Secured backend platforms using Amazon Cognito, OIDC, and JWT-based role-based access control, ensuring governed access for analysts, investors, and regulatory users with full auditability.
- Contributed to Databricks-based pipeline modernization by aligning backend service consumption with curated Delta Lake outputs, supporting reliable ML feature availability for downstream scoring workflows.
- Containerized backend services with Docker, deployed on AWS EKS, and supported CI/CD through GitHub Actions and Terraform, with OpenTelemetry-based observability across distributed service flows.

TransUnion, Stamford, CT **Sep 2019 – July 2023**

Software Engineer

Responsibilities:

- Built Django REST Framework and Flask APIs powering borrower onboarding, eligibility checks, and KYC flows, supporting multi-step progress saving, resume-later workflows, and partner integrations with consistently low latency under high transaction volumes.
- Developed Angular SPAs with NgRx/Redux state management and integrated RESTful APIs and WebSockets for real-time onboarding status updates, reducing UI blocking and improving responsiveness across the application.
- Implemented async background processing using Celery, RQ, and RabbitMQ (with DLQ, retry, and TTL policies) for document verification, bureau callbacks, and KYC event processing , keeping workflows stable under traffic spikes without blocking the frontend.
- Processed onboarding and credit events through Apache Kafka topics enabling reliable asynchronous communication between microservices handling KYC validation, credit bureau callbacks, and partner notifications across distributed systems.
- Managed relational data using SQLAlchemy ORM with Aurora PostgreSQL and Oracle, applying schema tuning, indexing strategies, and optimized query patterns to support high-throughput transactional workloads and compliance reporting.
- Built PySpark and Python ETL pipelines to move, transform, and reconcile onboarding and credit application data across S3, PostgreSQL, and MSSQL, implementing validation checks for missing, delayed, and duplicate records in a Hadoop/Hive environment.
- Stored document metadata and verification results in DynamoDB for low-latency lookups and leveraged OpenSearch for fast full-text search across case records, supporting both operational and compliance use cases.
- Implemented serverless event-driven workflows using AWS Lambda, Step Functions, and Glue to orchestrate document verification, bureau calls, and KYC tasks, containerizing Python services with Docker and deploying on ECS Fargate with GitHub Actions and Jenkins CI/CD pipelines.
- Processed over 100TB of structured and unstructured claims data using Apache Spark and Hadoop to extract fraud indicators, increasing fraud detection accuracy by 20% and reducing claim alert latency by 50% through Kafka and Airflow-based streaming architecture.
- Developed deep learning models in PyTorch and TensorFlow for customer churn prediction and underwriting score calibration, improving decision precision by 18% and serving model outputs through FastAPI and GraphQL APIs into underwriting and claims platforms.

- Applied NLP techniques using Hugging Face Transformers, spaCy, and RoBERTa to analyze service chats and customer feedback, enhancing claim resolution quality and driving measurable improvements in customer satisfaction metrics.
- Integrated AI-driven risk scoring and premium adjustment recommendations into policy systems via FastAPI and Flask, using Airflow-orchestrated pipelines and MLflow experiment tracking for reproducible, auditable insurance fraud detection workflows.
- Provisioned cloud infrastructure using Terraform and scaled model inference environments on Kubernetes, reducing AWS compute costs by 25% during peak hours and automating SageMaker and Vertex AI model deployments through Jenkins and GitHub Actions CI/CD.
- Built real-time portfolio and risk dashboards in Power BI and Looker using model-driven insights, accelerating financial reviews and risk reporting cycles by 30% for internal underwriting and operations teams.
- Secured APIs using AWS Cognito, OAuth2, JWT scopes, and least-privilege IAM policies with secrets managed via SSM and Secrets Manager, embedding Prometheus and ELK stack observability across pipeline stages to reduce downtime and accelerate anomaly response.

Arete IT Services, India ***2018 May – July 2019***

Software Developer

Responsibilities:

- Developed backend modules using Python (Django REST Framework) and Java (Spring Boot), deploying scalable microservices on AWS EC2, Elastic Beanstalk, and Kubernetes (EKS).
- Built RESTful APIs with secure authentication (JWT/OAuth2), API versioning, and efficient JSON structures for cloud-native access**.**
- Engineered real-time features such as order management, workflow execution, and streaming analytics using advanced algorithms, multithreading, and event-driven Java modules.
- Integrated backend logic with AWS RDS (MySQL/PostgreSQL), applying schema normalization, indexing, and stored procedure optimization for transactional and analytics-heavy workloads.
- Implemented CI/CD pipelines with Jenkins managing PyTest/UnitTest executions, Docker image builds, and zero-downtime deployments.
- Followed TDD practices, proactively monitoring test results in Jenkins and collaborating with QA for regression-free releases.
- Developed predictive models with PyTorch and applied optimization algorithms (SciPy linear programming) for logistics forecasting and cost minimization.
- Leveraged OOP principles, design patterns, and reusable utility tools to maintain clean, modular codebases across Python, Java microservices, and AWS Lambda.
- Collaborated with cross-functional teams across finance, logistics, and retail domains, conducting sprint demos, and aligning backend implementations with enterprise cloud standards to ensure scalability, reliability, and visibility.

Personal Projects and Research Work

AGORA - Autonomous AI Research Platform | agora-production-4b08.up.railway.app

- Built a multi-agent AI research platform orchestrating 4 specialized agents (Claude Sonnet, Claude Haiku, GPT-4o-mini) on unsolved math and CS problems with hand-coded agent orchestration in TypeScript with role-based turn management, cross-session memory persistence, and real-time SSE streaming
- Deployed on multi-cloud infrastructure Railway (GCP) for persistent backend with Docker/Nixpacks CI/CD pipeline, Supabase PostgreSQL (AWS) for structured conversation and agent memory storage, automatic daily scheduling with zero human intervention
- Designed a 4-agent research pipeline (Question Framing to Evidence Retrieval to Hypothesis Generation to Critic/Verifier) with individual persistent memory per agent, live Tavily web search, and multi-session problem continuity across days
- Built end-to-end in ~1,800 lines of TypeScript using Next.js App Router, Anthropic and OpenAI SDKs, PostgreSQL, and Server-Sent Events for real-time streaming and no agent frameworks used.

Development and Validation of Unsupervised Machine Learning Clustering Techniques | (link of the paper)

- Built a scalable clustering pipeline using K-Means, DBSCAN, Agglomerative Clustering, and GMM to segment social media users by sentiment and engagement; applied NLP and graph-based network analysis to reveal community patterns.

- Evaluated models on accuracy efficiency; work published in Springer Nature.

FAI-Enhanced TRIX Momentum Trading Dashboard | (GitHub Link)

- Developed an end-to-end AI-driven trading platform using CrewAI to coordinate autonomous agents for market signal generation and trade decision support.
- Developed modular LangChain agents for trend analysis, risk scoring, and sentiment evaluation, and orchestrated them through CrewAI to generate actionable BUY/SELL/HOLD signals based on the TRIX momentum indicator.
- Built an interactive Streamlit dashboard for analyzing real-time and historical stock data, configuring TRIX parameters, and tracking key performance metrics such as Sharpe ratio, total return, and maximum drawdown.

EDUCATION

- ***Masters in CS from University of Massachusetts, Lowell***
- ***Bachelors in Computer Science from KL University ,Vijayawada, Ind***

CERTIFICATIONS

- ***Microsoft Certified: Azure AI Fundamentals***
- ***Microsoft Certified: Azure Fundamentals***
- ***Databricks Generative AI Fundamentals***
- ***AWS Certified Solutions Architect - Associate***
- ***Multicloud Network Associate***