Dinh Khai Nguyen

+84 37838454 nguyendinhkhaiqt2005@gmail.com ngwkhai.com github.com/ngwkhai Hanoi, Vietnam

SUMMARY

Based in Vietnam, I am an AI researcher and AI engineer with a strong passion for artificial intelligence and data-driven technologies. My academic background in computer science and hands-on experience with projects in machine learning, deep learning, natural language processing, and financial technology have equipped me with both theoretical knowledge and practical skills. I have worked on projects ranging from image recognition and fraud detection to building intelligent chatbots and real-time systems. This journey has given me valuable expertise in data analysis, programming, and problem-solving. I am deeply motivated to create AI solutions that not only advance technology but also deliver real-world impact. Innovation, continuous learning, and pushing the boundaries of what AI can achieve are at the core of my work, and I aspire to build impactful projects that shape the future of intelligent systems.

EDUCATION

VNU University of Engineering and Technology Aug 2023 - Present
Bachelor of Artificial Intelligence GPA: 3.61/4

EXPERIENCE

VinSmart Future

AI Application Engineer Nov 2026 - Present
- Take on in-depth AI data work and own its quality control, ensuring every dataset meets the required standard before it is put to use.
- Support the management of a twelve-person team, helping keep its work on schedule and up to standard.

Junior AI Application Engineer Sep 2026 - Nov 2026
- Carried out data quality control, reviewing data against the required standard and making sure it met requirements before it was put to use.
- Supported the day-to-day management of a six-person team.

AI Intern May 2026 - Aug 2026
- Built the data collection and quality-control pipeline for a travel dataset: crawling and ingestion through to the verification pass that decides what is fit to keep.

Viettel Networks Sep 2025 - Jan 2026
AI Intern
- Research on generating structured project layouts from natural-language specifications, addressing the structural blindness large language models show on hierarchical output.
- Built the training set and the evaluation metric for the task, and fine-tuned compact open models with LoRA and a loss function aligned to that metric.

RESEARCH

NLP Lab Jun 2025 - Present
Member
- Participated in research and development activities in natural language processing, including exploring AI models, experimenting with text data, and contributing ideas to group projects.

PROJECTS

ResearchMap AI 2026
Solo — product, backend, AI pipeline, frontend, deployment
Turns a one-line description of a research field into a map of its literature: topics, trends and gaps. An agent answers questions about the field, citing only papers it actually retrieved.
Live demo: researchmap.ngwkhai.com · Case study: ngwkhai.com/en/projects/researchmap
- Built the full product solo: a FastAPI backend on PostgreSQL with pgvector, a React frontend, sign-in with Google OAuth and TOTP, plans with quotas, and an admin panel.
- Designed the analysis pipeline. An LLM plans the OpenAlex queries, papers are embedded and clustered with KMeans, an LLM names each topic, and research gaps are computed from topics that share no papers.
- Built an agentic RAG chat on OpenAI tool calling, with 7 tools and at most 4 rounds. Its citations come only from papers the tools returned, and it sits behind a relevance floor measured on real data.
- Benchmarked it with four harnesses and an LLM judge, and shipped it to Cloud Run through GitHub Actions and Workload Identity Federation.

Key results: Cited papers verified real **36/36** · Retrieval precision@5 **0.867** · Cost per answer **$0.00032** (gpt-5.4-nano, measured)
Technologies: Python, FastAPI, PostgreSQL, pgvector, scikit-learn, OpenAI API, React, TypeScript, Cloud Run, GitHub Actions.

TaxMate AI 2026
Team lead of four — scope, integration and releases; Risk core, API and auth, legal retrieval, GCP deployment
A tax-compliance assistant for Vietnamese online sellers. It answers legal questions with a citation on every claim and reconciles sales against declared revenue. Built by a team of four that Khai led.
Live demo: taxmate.ngwkhai.com · Case study: ngwkhai.com/en/projects/taxmate
- Led a four-person team building four cores in one monorepo: Legal RAG, Risk, Graph and quantum Optimization. Set scope, budget and releases, reviewed each core and integrated them into one product.
- Built the Risk core. It imports CSVs and reconciles sales, bank deposits and declared revenue through four versioned rules, and returns `insufficient_data` instead of a misleading score when data is missing.
- Rebuilt the legal assistant's retrieval with BM25 over Vietnamese syllable pairs, and made every claim cite only chunks that were actually retrieved. This raised document recall@5 from 32 to 52 of 56 test questions.
- Built multi-tenant auth on Supabase, and deployed dev and production on Cloud Run behind a load balancer and CDN through GitHub Actions.

Key results: Legal recall@5 **93%** (52 of 56 questions, up from 32) · Legal documents indexed **13** (1,118 chunks) · Cores integrated **4**
Technologies: Python, FastAPI, PostgreSQL, Supabase Auth, OpenAI API, React, TypeScript, Cloud Run, GitHub Actions.

Detoura 2026

Solo — architecture, rules engine, protocol, tournament service, search algorithms, deployment
A realtime two-player board game built as a platform. It has a deterministic rules core, one set of typed schemas that validates, documents and tests the API, and a separate tournament service for bots.
Live demo: detoura.ngwkhai.com · Case study: ngwkhai.com/en/projects/detoura
- Designed a nine-package TypeScript monorepo around a deterministic rules core with no dependencies, so the same code runs in the browser, on the server and in replays.
- Made the platform game-agnostic: rules plug into one typed interface, and the server treats a match as opaque state.
- Used one set of Zod schemas to validate every message and to generate the OpenAPI and WebSocket spec, with a contract-drift suite that runs against the live deployment.
- Built an authoritative WebSocket server with rate limiting and a server-owned clock, a Postgres tournament service, and a Cloud Run pipeline with no stored credentials.

Key results: Workspace packages **9** · Interfaces under contract **5** · Protocol messages **14**
Technologies: TypeScript, Zod, ws, React, Vite, PostgreSQL, Docker, Cloud Run, GitHub Actions.

Bacera 2026

Team of two — Pancake POS plugin, shop and checkout, payments, search, demo and infrastructure (theme, auth and workshops by the other developer)
An online shop and workshop booking site for a Hanoi pottery studio, built on WordPress with Pancake POS as the system of record. Its public demo replays recorded POS traffic.
Live demo: bacera.ngwkhai.com · Source: github.com/ngwkhai/bacera-pancake-integration · Case study: ngwkhai.com/en/projects/bacera
- Built a WordPress plugin that integrates Pancake POS through six modules, so products, stock and orders live in the POS and the site never keeps a second inventory.
- Kept the API key on the server. Checkout calls nonce-checked REST routes that talk to the POS, and product images are proxied under the studio's own domain.
- Verified bank transfer, card and PayPal payments with payment intents that expire, plus a status endpoint that turns the POS's inconsistent "paid" signals into one answer.
- Turned the client project into a safe public demo. It replays recorded POS responses, cannot reach the real POS from its container, and resets every night on Compute Engine.

Key results: Integration modules **6** · Custom SQL tables **15** · POS endpoints replayed **12**
Technologies: PHP 8.3, WordPress, REST API, Tailwind CSS, Alpine.js, MariaDB, Docker, Caddy, Compute Engine.

Flow Center 2026

Solo, for a client — full stack, AI features, deployment
Booking, a self-assessment and a chat assistant for a clinic that treats period pain with traditional medicine. The AI stays strictly inside clinician-approved content.
Live demo: flowcenter.ngwkhai.com · Case study: ngwkhai.com/en/projects/flowcenter
- Built the clinic's site solo on Next.js 16, PostgreSQL and Drizzle, with booking and a staff workflow, a self-assessment, a chat assistant, auth and an admin area, deployed on Vercel and Neon.
- Made booking safe under concurrency with advisory locks and state transitions enforced in SQL. A test sends eight parallel bookings at two seats.
- Kept the AI inside approved content. Fixed rules score the assessment, warning signs stop it, and the model may only rephrase the clinic's template, falling back to that template on any failure.
- Built a RAG chat assistant with hybrid Vietnamese search: accent-folded keywords plus pgvector, fused with RRF. It reads data, but books only when the visitor presses the button.

Key results: Parallel bookings in the race test **8** (for 2 seats, none overbooked) · Warning signs that stop the analysis **10** · Days a conversation is kept **90**
Technologies: Next.js 16, TypeScript, PostgreSQL, Drizzle ORM, pgvector, OpenAI API, Vercel, Neon.

AI Pipeline Mastery 2026

Solo — platform architecture, GPU orchestration, build tooling, content production
A course built as software. A Python pipeline generates the course material and executes it as a test, and a Next.js platform runs learners' code either in the browser or on a pool of remote GPUs.
Case study: ngwkhai.com/en/projects/ai-pipeline-mastery
- Built a Python toolchain that generates slide decks, documents and exercises from source. So far it has produced 383 slides and 129 figures rendered from real data.
- Enforced quality by execution. Every reference solution must pass, every starter must fail as expected, and generated files are rebuilt and compared byte for byte before each release.
- Built the learner platform on Next.js 16 with server actions and no internal API layer. The browser never touches the data store, and files are served as signed URLs.
- Designed a GPU job queue for a serverless host. It leases accounts from a pool, tracks jobs through polling and a scheduled sweep, and can run end to end in a local simulation mode.

Key results: Slides generated **383** · Figures rendered **129** · Execution lanes **2** (in-browser and remote GPU)
Technologies: Next.js 16, Firebase Auth, Firestore, Cloudflare R2, Pyodide, Kaggle API, Tailwind v4.

Classroom Trash Detection 2026

Full system — distributed architecture, client, inference service, runtime optimisation, deployment
A computer-vision system split across a network. A stateful client handles the camera and the dashboard, and a stateless FastAPI service runs YOLOv8 on a GPU.
Source: github.com/ngwkhai/AI_Server · Case study: ngwkhai.com/en/projects/ai-server
- Refactored a single-process app into a stateful client and a stateless GPU inference service, connected by one authenticated HTTP contract.
- Cut network and GPU load by downscaling frames to 640 px and rescaling the returned coordinates, and by running inference on sampled frames with cached results in between.
- Added a local fallback for when the remote service is down, and a status API that reports which backend served each request and how long it took.
- Exported the model to half-precision ONNX, loaded it at startup, and deployed the same FastAPI app to a notebook GPU, a Docker image and a serverless GPU.

Key results: Model formats **3** (PyTorch, ONNX, TensorRT) · Upload width **640 px** · Deploy targets **3**
Technologies: Python, YOLOv8, ONNX Runtime, FastAPI, Flask, OpenCV, SQLite, Docker, Modal.

Neural Machine Translation 2025

Dataset construction, model implementation, evaluation design

Built a 12.5M-pair dataset for Vietnamese diacritic restoration and implemented RNN, GRU, LSTM and Transformer seq2seq models from scratch in PyTorch.

Source: github.com/ngwkhai/Neural-Machine-Translation · Case study: ngwkhai.com/en/projects/neural-machine-translation

- Built the first large-scale Vietnamese diacritic restoration dataset (12.5M sentence pairs without → with diacritics).
- Implemented Seq2Seq models (RNN, GRU, LSTM, Transformer) from scratch using PyTorch.
- Designed and conducted comprehensive model evaluation with BLEU and ChrF++ metrics, achieving BLEU 81.31 and ChrF++ 88.57 with Transformer, significantly outperforming RNN and GRU baselines.
- Contributed across data construction, model development, and performance evaluation, leading to a robust prototype system for Vietnamese text restoration.

Key results: BLEU **81.31** (Transformer) · ChrF++ **88.57** (Transformer) · Sentence pairs **12.5M** (without → with diacritics)

Technologies: PyTorch, Transformer, LSTM, GRU, RNN, sacreBLEU.

Optimization for AI Inference Engines on GPUs 2025

Benchmarking methodology, profiling, analysis

Optimising GPU inference for ResNet50 and DistilBERT with FP16, INT8, TorchScript/torch.compile and ONNX.

Paper: PDF · Case study: ngwkhai.com/en/projects/gpu-inference-optimization

- The project focuses on optimizing AI inference on GPUs for ResNet50 and DistilBERT through FP16, INT8, TorchScript/torch.compile, and ONNX.
- Results show that FP16 provides significant improvement, especially for DistilBERT: throughput increased from 305 → 1202 samples/s at batch size 8 (≈4×).
- INT8 (TensorRT) achieves 2–5× speedup but requires careful calibration to maintain accuracy.
- Graph optimization yields a 1.2–1.8× speedup, while CUDA profiling identifies the main bottlenecks in GEMM and ReLU kernels.
- The project demonstrates that mixed-precision and quantization are the most effective approaches for large-scale inference optimization.

Key results: FP16 throughput **≈4×** (DistilBERT, 305 → 1202 samples/s at batch size 8) · INT8 speedup **2–5×** (TensorRT, needs careful calibration) · Graph optimisation **1.2–1.8×**

Technologies: TensorRT, ONNX Runtime, TorchScript, torch.compile, Nsight, CUDA.

License Plate Recognition System 2025

System design, API and database layer, deployment

A real-time license plate recognition system: a fine-tuned YOLOv5 model with OCR, a FastAPI and PostgreSQL backend and a React frontend, deployed with Docker.

Source: github.com/ngwkhai/License-Plate-Recognition · Case study: ngwkhai.com/en/projects/license-plate-recognition

- Developed a real-time license plate recognition system by integrating a fine-tuned YOLOv5 model with OCR for text extraction.
- Focused on system development: designed the pipeline, built REST APIs with FastAPI, managed data in PostgreSQL, and created a ReactJS-based frontend. Deployed the system with Docker for scalability and ease of integration.
- Achieved real-time processing at ~12 FPS with reliable recognition accuracy.
- Enhanced traffic management efficiency and enabled citizens to conveniently check their traffic violation records online.

Key results: Throughput **~12 FPS** (real-time processing)

Technologies: YOLOv5, OCR, FastAPI, PostgreSQL, React, Docker.

Quantitative Trading System 2025

End-to-end — feature engineering, model, training pipeline

A quantitative trading system built from scratch in PyTorch: a Transformer that predicts BUY/SELL/HOLD signals from financial indicators and historical trading data.

Source: github.com/ngwkhai/Quantitative_trading_system · Case study: ngwkhai.com/en/projects/quantitative-trading-system

- A quantitative trading system built from scratch in PyTorch using a Transformer model to predict BUY/SELL/HOLD signals from financial indicators and historical trading data, with a processing pipeline of 3.4M training samples and 351K development samples.
- The system incorporates features such as log-volume, technical indicators (MACD, RSI, Bollinger Bands, ATR), and Time2Vec with two-step normalization.
- Achieved Validation Accuracy of 92.75% and Test Accuracy of 92.50%.

Key results: Validation accuracy **92.75%** · Test accuracy **92.50%** · Training samples **3.4M** (plus 351K development samples)

Technologies: PyTorch, Transformer, Time2Vec, MACD, RSI, Bollinger Bands, ATR.

Credit Card Fraud Detection 2024

Preprocessing, resampling, model comparison, evaluation

Fraud detection on the Kaggle credit card dataset, where 0.172% of transactions are fraud, comparing four classifiers with SMOTE and ADASYN resampling.

Source: github.com/ngwkhai/Card-fraud-detection · Case study: ngwkhai.com/en/projects/credit-card-fraud-detection

- Conducted fraud detection on the Kaggle dataset (284,807 transactions, only 492 fraud → 0.172%).
- Applied SMOTE and ADASYN to handle severe class imbalance.
- Implemented and compared Logistic Regression, Decision Tree, Random Forest, and XGBoost.
- Achieved ROC-AUC 0.9817 and Recall 0.735 with XGBoost on imbalanced data.
- Improved Recall to 0.816 with ADASYN while maintaining high ROC-AUC (0.9744), outperforming other models.
- Contributed to data preprocessing, resampling, model development, and evaluation, delivering a robust prototype for financial fraud detection.

Key results: ROC-AUC **0.9817** (XGBoost on imbalanced data) · Recall **0.816** (with ADASYN, up from 0.735) · Positive rate **0.172%** (492 frauds in 284,807)

Technologies: XGBoost, Random Forest, Decision Tree, Logistic Regression, SMOTE, ADASYN.

Large-Scale Text Sentiment Classification 2024

Pipeline implementation and comparative benchmarking

Sentiment classification of 1.6M tweets (Sentiment140) with Naive Bayes and SVM, run on Apache Hadoop and Apache Spark.

Source: github.com/ngwkhai/Text-Sentiment-Classification-Hadoop-Spark · Case study: ngwkhai.com/en/projects/text-sentiment-classification

- This project explored text mining, sentiment analysis, and parallel/distributed computing with Apache Hadoop and Apache Spark.
- The system was implemented on the Sentiment140 dataset (1.6M tweets) using Naive Bayes and SVM.
- Results showed that Naive Bayes achieved 75% accuracy with faster processing speed, while SVM reached higher accuracy (80%–82%) but required longer training time on Hadoop.
- Spark outperformed Hadoop in execution time and scalability for large-scale data.

Key results: Dataset **1.6M** (tweets, Sentiment140) · SVM accuracy **80–82%** (longer training time on Hadoop) · Naive Bayes accuracy **75%** (faster processing)

Technologies: Apache Hadoop, Apache Spark, Naive Bayes, SVM, Sentiment140.

YOLOv1 With CBAM From Scratch 2024

Full implementation — dataset, loss function, attention module, training

YOLOv1 rebuilt from scratch in PyTorch, with its own dataset class and loss function, and CBAM attention added to the feature extractor.
Case study: ngwkhai.com/en/projects/yolov1-cbam

- This project rebuilt the YOLOv1 model from scratch using PyTorch, where I implemented FruitDataset and YoloLoss, and integrated CBAM (Convolutional Block Attention Module) to enhance feature extraction, with a total of 269,168,434 parameters.
- The model employed the ReduceLROnPlateau scheduler to optimize the training process.
- Training results showed a Train mAP of 0.8829 and a Best Validation mAP of 0.6994.
- These results demonstrate that CBAM provides significant improvements in object detection performance.

Key results: Train mAP **0.8829** · Best val mAP **0.6994** · Parameters **269.2M** (269,168,434)
Technologies: PyTorch, YOLOv1, CBAM, ReduceLROnPlateau.

SKILLS

Programming	Python, C/C++, Java, SQL, TypeScript
Machine Learning	PyTorch, Transformer, Seq2Seq, YOLOv5 / YOLOv1, CBAM, XGBoost, SMOTE / ADASYN, scikit-learn
NLP	Vietnamese NLP, Diacritic restoration, BLEU / ChrF++, Sentiment classification
Systems & Deployment	FastAPI, Docker, PostgreSQL, React, Apache Spark, Apache Hadoop
Inference & Performance	TensorRT, ONNX Runtime, TorchScript, torch.compile, FP16 / INT8, CUDA profiling
Languages	Vietnamese: Native, English: Professional

HONORS & AWARDS

Third Prize, Student Scientific Research Award — Artificial Intelligence, VNU University of Engineering and Technology	2026
Third Prize, Student Track — Humanitarian Logistics Hackathon	2026
Academic Encouragement Scholarship, University of Engineering and Technology	2024
Second Prize, Provincial Excellent Student Contest in Physics, Thai Nguyen Province	2023
Third Prize, Provincial Excellent Student Contest in Informatics, Thai Nguyen Province	2021

LEADERSHIP

Institute for Artificial Intelligence Sep 2024 - Present

Class Vice-President & Executive Committee Member

- Served as Class Vice-President and an active member of the Executive Committee, coordinating student projects, organizing meetings, and contributing to strategic planning within the institute.