Ata Caner Cetinkaya

Address: London, UK
Phone: +44 7391 904514 | Email: atacanercetinkaya@gmail.com
LinkedIn: linkedin.com/in/atacanercetinkaya | GitHub: github.com/Capataina

Education

University of York *York, UK*

Bachelor of Engineering (BEng) in Computer Science *September 2022 – June 2025*

- Built CNN models for flower classification and facial recognition, MLP for body dysmorphia prediction and RNN for ciphertext indistinguishability attacks, demonstrating proficiency across supervised learning architectures.
- Implemented genetic algorithm for beer quality prediction using DEAP, exploring evolutionary optimisation for regression tasks.
- Analysed network traffic using Wireshark, Nmap, and tcpdump to identify TCP SYN, UDP, and ICMP floods indicative of denial-of-service attacks; proposed security enhancements for banking and fintech infrastructure.
- Led development of a simulation game in Java using libGDX as lead developer in an 8-person team, coordinating architecture, feature implementation, implementation pipelines and automated cross-platform testing.
- Collaborated on a fully multithreaded top-down roguelite in Unity with ECS architecture, focusing on safe concurrency and performance.

Projects

Image Browser — Local-First Pinterest-Style Image Manager [ ]

Rust, Tauri v2, React 19, TypeScript, SQLite, ONNX Runtime, CLIP

- Built local-first desktop application for browsing and organising large image collections with Pinterest-style masonry layout, manual tagging, semantic search, and visual similarity recommendations.
- Implemented recursive filesystem scanner with SQLite database (WAL mode) for concurrent access, thumbnail generation, and Tauri IPC command system for frontend-backend communication.
- Integrated CLIP image and text encoders via ONNX Runtime for semantic search using natural language queries and visual similarity engine with cosine similarity over normalised embeddings.
- Architected fully offline system with asset protocol image loading, tag-based filtering, slideshow mode, and complete user privacy with no cloud dependencies or external services.

Tectra — High-Performance Hybrid Trading Infrastructure [ ]

C++20, FlatBuffers, Lock-Free, Shared Memory, Prometheus

- Designing modular trading infrastructure with market data feed handler, pre-trade risk engine, kill-switch, and deterministic replay for production-grade finance applications.
- Building ITCH decoder with zero-copy parsing, gap detection, and L2 book reconstruction, targeting >1M msgs/sec per core with microsecond-level risk checks.
- Implementing hot-reloadable risk rules (price bands, size limits, throttles) and append-only checksummed journals for post-incident root-cause analysis.
- Architecting dual-plane system with binary fast path for low-latency flow and structured control plane for operator management and circuit-breaker coordination.

Vynapse — Hybrid Neuroevolution & Deep Learning Runtime [ ]

Rust, NEAT, DEAP, PyTorch, TensorFlow

- Building hybrid ML runtime combining NEAT topology mutation, DEAP evolutionary search, and PyTorch-style backpropagation in safe Rust for neuroevolution research.
- Implemented trait-based architecture with genome, fitness, selection, mutation, and crossover abstractions to enable composable evolutionary algorithms.
- Developed task-based fitness evaluation system with matrix operations and activation functions for neural network training via genetic algorithms.
- Designing TensorFlow-inspired graph execution engine to unify dynamic architecture evolution with gradient-based training (Milestone 1 of 10 in progress).

Xyntra — ML Graph Fusion Compiler Pass [ ]

Rust, egg, WGSL, PTX, ONNX, TorchScript

- Designing compiler pass for ONNX and TorchScript graph fusion using e-graph rewriting to identify kernel fusion opportunities and reduce GPU overhead.
- Implemented foundational IR types including type-safe NodeID, TensorShape, OpKind primitives, and graph representation with operation dependencies.
- Building validation framework with cycle detection, shape compatibility checking, and operation constraint verification for fusion legality analysis.
- Planning WGSL and PTX code generation backends with autotuned tile sizes to emit fused GPU kernels (early foundation stage).

Zyphos — HTTP Server & Network Protocol Laboratory [ ]

Rust, TCP/UDP, HTTP/1.1, HTTP/2, WebSockets, QUIC, io_uring, SIMD

- Building modular HTTP server from scratch in safe Rust to explore network protocol implementation, socket programming, and performance patterns from compilers and trading systems.
- Implemented TCP listener with thread-per-connection model, HTTP/1.0 parser with method/path/header extraction, and RFC-compliant response formatting with routing and error handling.
- Designed 30-milestone roadmap spanning raw sockets, thread pools, memory pools, SIMD parser optimisation, io_uring integration, lock-free statistics, rate limiting, and HTTP/2 frame processing.
- Structured as deliberate learning progression through networking fundamentals (TCP state machines, epoll/kqueue event loops, zero-copy buffers, kernel bypass with sendfile/splice).

Nyquestro — Lock-Free Limit Order Book Trading Engine [ ]

Rust, Lock-Free, Matching Engine

- Designing lock-free limit order book engine in safe Rust to explore ultra-low-latency market microstructure without unsafe blocks or OS locks.
- Implemented type-safe primitives for OrderID, price (in cents), quantity, timestamps (nanosecond precision), and order lifecycle status tracking.
- Built Order and PriceLevel abstractions with FIFO queue semantics, partial fill handling, and price-time priority matching rules.
- Planning atomic price buckets, flat-combining allocator, and multi-gateway ingestion with FIX/binary protocols (early foundation stage).

Multi-LLM Debate CLI — AI Orchestration & Knowledge Synthesis Framework [ ]

Python 3.13, Typer, asyncio, Google Gemini API, TOML, Rich/Textual

- Building command-line framework for orchestrating multi-round AI debates across heterogeneous LLM providers (OpenAI, Anthropic, Google, Ollama) with recursive summarisation to surface consensus and divergence.
- Implemented abstract LLM backend interface with async support, provider-agnostic architecture handling API differences, rate limits, and response formats transparently.
- Designed rolling summarisation algorithm maintaining coherent context over unlimited rounds by identifying new arguments, tracking stance changes, and pruning redundancy.
- Planning 12-milestone roadmap including async execution, TOML-based configuration, alignment matrix analytics, local model support, tool integration (web search, citations), and interactive TUI with live monitoring.

AsteroidsAI — Neuroevolution-Based Game AI with Real-Time Visualisation [ ]

Python, Arcade, NEAT, DEAP, GNN, SAC

- Developing simulation for benchmarking AI agents using NEAT, DEAP-GA, DEAP-GP, Evolution Strategies, and GNN with Soft Actor-Critic.
- Comparing evolutionary and gradient-based approaches to real-time decision-making in sparse-reward environments.

Credit Card Fraud Detection [ ]

Python, XGBoost, Pandas, NumPy, scikit-learn, UMAP, Plotly

- Built XGBoost classifier achieving 94% precision, 84% recall, and 89% F1 score on imbalanced fraud dataset

with 0.17% fraud rate across 285,000 transactions.
- Applied UMAP, PCA, and t-SNE for dimensionality reduction to visualise fraud clustering, revealing distinct spatial separation of fraudulent cases.
- Conducted error analysis showing misclassified fraud cases overlap heavily with legitimate transactions, highlighting intrinsic dataset ambiguity.
- Selected XGBoost for superior performance on tabular data, effective imbalance handling, and faster training compared to deep learning alternatives.

My Game Mods — 150K+ Total Downloads [ ]
C#, Java, TypeScript, XML, JSON

- Developed 18+ gameplay mods across RimWorld, Escape from Tarkov (SPT-AKI), Minecraft, and Terraria, achieving an average of 2,000–15,000 downloads per mod with sustained active user engagement.
- Built Passion On Level Up mod for RimWorld using runtime patching with Harmony to dynamically modify pawn skill progression and passion tiers during gameplay without static loading.
- Built real-time health scaling system for Escape from Tarkov using SPT-AKI hooks, adjusting individual body part health values dynamically as player PMC and Scav characters levelled up (12,000 total downloads).
- Created Minecraft furnace interaction mod with block-level runtime editing, enabling flint-and-steel ignition for single-item smelting through custom event handling.
- Extended Terraria's Calamity mod with custom weapon upgrade for Terratomere, implementing custom homing projectile guidance system with trajectory calculations and collision detection.

Neuronika — AI-Powered Personal Knowledge Management Tool [ ] [ ]
TypeScript, React, OpenRouter, Vis.js

- Built AI-enhanced note-taking tool with semantic search, adaptive tagging via LLMs through OpenRouter, and interactive graph visualisation.
- Combined markdown parsing, vector embeddings, and metadata clustering to support non-linear knowledge workflows.

Personal Website — Interactive Portfolio [ ] [ ]
Next.js 15, React 19, TypeScript, Tailwind CSS, Framer Motion, Shadcn UI

- Built interactive portfolio website with quadrant-based navigation, dynamic theming, and field-specific filtering for projects and skills.
- Implemented particle network background with spatial partitioning for O(n) collision detection and mouse-reactive physics simulation.
- Designed component architecture with memoised animations, context-driven accent colours, and accordion-based content organisation.
- Deployed on Vercel with SSR/SSG hybrid rendering for fast load times and SEO optimisation.

Certifications

Google Developer Badges and Certificates
- Machine Learning Crash Course (Issued: July 2025)
- *Skills: Machine Learning, AI, TensorFlow, Linear Algebra, Calculus, Multivariable Calculus, Optimisation*

DeepLearning.AI – Andrew Ng
- MCP: Build Rich-Context AI Apps with Anthropic (Issued: May 2025)
- *Skills: AI, Large Language Models*

HackTheBox – Dr. AITH Cybersecurity Training
- Introduction to Malware Analysis (Issued: July 2025) | Linux Fundamentals (Issued: July 2025)
- *Skills: Reverse Engineering, Linux*

DataCamp Data Science and AI Certificates
- Introduction to ChatGPT (Issued: November 2023) | Intermediate Python (Issued: July 2023) | Understanding Data Science (Issued: July 2023)
- *Skills: NLP, ChatGPT, LLM Operations, Python, NumPy, Pandas, Data Analysis, Data Engineering, Data Science*

CME Group Trading and Finance Education

- Technical Analysis (Issued: November 2023) | Trading and Analysis (Issued: November 2023) | Using Fundamental Analysis When Evaluating Trades (Issued: November 2023)
- *Skills: Technical Analysis, Fundamental Analysis, Strategic Thinking, Problem Solving*

Skills

- **Languages:** Rust, Python, TypeScript, JavaScript, C#, C++, Java
- **Mathematics:** Linear Algebra, Tensor Algebra, Calculus, Probability, Discrete Mathematics, Optimisation
- **Systems & Low-Latency Programming:** Low-Latency Programming, Lock-Free Data Structures, Multithreading, TCP/UDP Sockets, HTTP Protocol Implementation, Safe Concurrency, Async/Await, Linux (NixOS and Arch)
- **Compiler & GPU Tooling:** IR Design, Graph Rewriting, WGSL, PTX, ONNX, TorchScript, Kernel Fusion
- **Desktop Application Development:** Tauri v2, Electron Alternatives, IPC Architecture, Local-First Systems
- **AI & ML Frameworks:** PyTorch, TensorFlow, DEAP, scikit-learn, NEAT, XGBoost, ONNX Runtime, CLIP
- **Scientific Computing & Visualisation:** NumPy, Pandas, SciPy, Matplotlib, Seaborn, Plotly, Vis.js, UMAP, PCA, t-SNE
- **Frontend & Full-Stack Frameworks:** React, Next.js, Vite, Tailwind, Node.js, Supabase
- **Database Systems:** SQLite (WAL Mode), Concurrent Access Patterns, Embedded Databases
- **DevOps & Tooling:** Git, GitHub, CI Pipelines, Testing Frameworks
- **Cybersecurity & Networking:** Wireshark, Nmap, tcpdump, Network Traffic Analysis