TEMITAYO ADEFEMI

Google Scholar ◇ Github ◇ LinkedIn

RESEARCH INTERESTS

I am interested in parallel computing, high-performance computing, computer architecture, hardware & software co-design, compilers, dynamic neural networks, data structures & algorithms, embedded & operating systems, theoretical computer science, machine-learning systems, affective computing, robotics, distributed systems, performance analysis and optimization, quantum computing

EDUCATION

University of Edinburgh *Sept 2026*
MSc in High Performance Computing with Data Science [1] [2] [3] [4] [5] [6] [7]
Grade: **Merit**
Related courses: Threaded Programming, Message Passing Interface, Programming Skills, Parallel Design Patterns

Dissertation: **"Juliet: Per-Sample Conditional Branching for Efficient Convolutional Networks"** - Advised by Dr. Adam Carter

Aston University *Sept 2019*
MSc in Business and Management
Grade: **Distinction**
Related courses: Marketing Management, International Business Strategy, Accounting for Management, People and World Organization.

Dissertation: **"The Relationship between Academic Motivation and the Big Five Personality model in Aston University Students"** - Advised by Dr. Uwe Napiersky - 73%

Coventry University *Sept 2018*
BA in Business Administration
Grade: **Second Class Upper**
Related courses: Quantitative Methods for Business, People Management, Supply Chain and Operations Management, Introduction to Financial Services

PREPRINTS

[1] **Temitayo Adefemi**, "A conceptual framework for trie augmented neural networks (tanns)," 2024. [Online]. Available: `https://arxiv.org/abs/2406.10270`.

[2] **Temitayo Adefemi**, "Cellular automata as a network topology," 2024, Cited by 3. [Online]. Available: `https://arxiv.org/abs/2407.05048`.

[3] **Temitayo Adefemi**, "Analysis of the matrix multiplication algorithm on the cirrus supercomputer," 2024, Cited by 1. [Online]. Available: `https://arxiv.org/abs/2408.15384`.

[4] **Temitayo Adefemi**, "What every computer scientist needs to know about parallelization," 2025, Cited by 6. [Online]. Available: `https://arxiv.org/abs/2504.03647`.

[5] **Temitayo Adefemi**, "The entropy of parallel systems," 2025. [Online]. Available: `https://arxiv.org/abs/2509.12256`.

PUBLICATIONS

[6] **Temitayo Adefemi**, "When do MPI derived datatypes help? a multi-node, cross-implementation study of communication-bound stencil halo exchange," *Concurrency and Computation: Practice and Experience*, 2026.

[7] **Temitayo Adefemi**, "Comparative performance of MPI, OpenMP, and hybrid parallelism for a 3d jacobi solver on ARCHER2," *Performance Modeling, Benchmarking and Simulation*, 2026.

RESEARCH EXPERIENCE

Do MPI Derived Datatypes Actually Help? — October 2025

- Investigated whether MPI *derived datatypes* improve performance over manual packing and unpacking across three real-world, two-dimensional stencil applications: a CFD Jacobi solver, Conway's Game of Life, and a lattice-based image-reconstruction kernel.
- Implemented two functionally equivalent versions of each application: a BASIC version using MPI basic datatypes with explicit packing into scratch buffers, and a DDT version using `MPI_Type_vector`, `MPI_Type_create_subarray`, and `MPI_Type_create_resized` to describe non-contiguous halo regions.
- Ensured strict API parity and compared three communication semantics: non-blocking point-to-point communication (`MPI_Irecv`, `MPI_Isend`, and `MPI_Waitall`), neighborhood collectives (`MPI_Neighbor_allt` and MPI-4 persistent operations (`*_init`). This isolated the effect of datatype representation from algorithmic differences.
- Benchmarked all variants under strong and weak scaling from 1 to 16 ranks on a single ARCHER2 node across four MPI implementations: MPICH, Open MPI, Intel MPI, and MVAPICH2. Validated that the BASIC and DDT versions produced bitwise-identical halo data.
- Found that no single strategy was universally optimal: DDTs were the fastest in some cases and among the slowest in others. Performance depended heavily on the MPI implementation, problem size, communication semantics, and application, demonstrating that performance portability is not guaranteed and that practitioners should profile both BASIC and DDT designs on their target systems.

Juliet Architecture — October 2025

- Proposed a trie-structured, sample-wise dynamic architecture featuring learned per-node routing and online growth-and-pruning mechanisms.
- Developed practical, training-variance-triggered splitting and exponential moving average (EMA)-based pruning methods to stabilize topology learning under standard supervised-loss functions.
- Conducted an empirical study using ResNet-18, EfficientNet-B0, and DenseNet-121 on CIFAR-10 across ARCHER2 and EIDF, together with an ImageNet-1K validation experiment using ResNet-101 on H100 GPUs. Included FLOP and latency profiling and sensitivity analyses for growth and pruning hyperparameters.
- Analyzed utilization and routing-overhead factors that contribute to the gap between theoretical and realized speedups and developed corresponding deployment guidelines.
- Achieved 27.1 Top-1 accuracy points per GFLOP at ImageNet scale with the best-performing architecture, outperforming SkipNet, ConvNet-AIG, and BlockDrop.

Entropy of Parallel Systems — September 2025

- Developed a graph-theoretic mathematical model using logarithmic measures to quantify the entropy of parallel computing systems.
- Calculated the entropy of the ten highest-ranked supercomputers on the TOP500 list.

- Empirically validated the model on the ten systems, identifying a strong negative correlation between system entropy and LINPACK performance ($r = -0.7832$, $p = 0.0077$). The results suggest that lower-entropy systems may be more computationally efficient.

Trie-Augmented Neural Networks (TANNs) **June 2024**

- Integrated trie data structures with neural networks to improve model interpretability and computational efficiency.
- Embedded specialized miniature neural networks within trie nodes, achieving comparable or slightly improved performance on text-classification tasks while providing a more transparent decision-making process.

“

PROFESSIONAL EXPERIENCE

Mansa December 2022 - November 2023
Senior Software Engineer

· Enhanced application scalability by integrating efficient data processing flows.
· Contributed to significant revenue growth by engineering robust features and optimizing existing infrastructure.
· Developed analytics pipelines in Python and C-based tooling to manage real-time data, improving diagnostic insights.
· Implemented secure protocols to streamline user authentication and reduce system overhead.

Union Maritime Agency Limited January 2021 - July 2022
Senior Backend Engineer

· Developed server-side logic in Python and Golang with an emphasis on performance and scalability.
· Employed modern communication protocols to enhance data transfer efficiency across microservices.
· Led DevOps initiatives to optimize cloud infrastructure and reduce operational costs.
· Coordinated cross-functional teams to ensure seamless project execution and deployment.

PROJECTS & LIBRARIES

Delta-Encoded Tensor Library

· Implemented a high-performance C++ tensor abstraction featuring delta encoding, cache-aware memory layouts, and alignment-aware stride computation to reduce memory footprint and improve access efficiency.

Juliet

· Developed a Python library for creating dynamic neural networks using the Juliet architecture (trie backbone, branch selector, and grow/prune mechanisms).

Distributed Computing Framework

· Designed and implemented a C framework to distribute large datasets across multiple processes for parallel computation using Armadillo and MPI.

HydraDB

· Engineered a high-performance in-memory database in C, primarily focusing on concurrency and low-latency data access.

Kokoca

· Created a word-processing application with paraphrasing, referencing, and multi-format support; launched to **200** trial users.

Prime Market

· Created a parimutuel prediction market which allows users to stake money on real-world events and receive payouts when the market has been resolved using Typescript

OPEN SOURCE CONTRIBUTIONS

Kivy: Contributed cross-platform UI components for diverse applications.
Plyer: Enhanced API modules to improve data handling and integration with Kivy-based apps.

HONORS, AWARDS & ACHIEVEMENTS

Awarded the **EPCC Dissertation Scholarship**	*Summer 2025*
Top 5% of Class (Aston University)	*Fall 2019*
Became a member of **Turing Engineering Pool**	*Summer 2021*
Became a member of **Mercor Talent Pool**	*Spring 2025*
Became the **Youngest Member of Andela Talent Network** (Joined at 21)	*Summer 2021*

SELECTED ACADEMIC PROJECTS

Parallel Cellular Automata using MPI (Report & Repository) Jan 2024 - June 2024
Dr David Henty – *University of Edinburgh* INFD11011: Message Passing Interface

· Wrote a C program which required dividing cellular automata array cells across multiple processes, which required handling an even or odd number of processes and an even or odd number of cellular automata array cells. The rationale for our design choices and the second was to analyse the profiling data.
· Achieved an A3 mark of 72% (Distinction) overall for the module.

Parallel Adaptive Quadrature using OpenMP (Report & Repository) Sept 2023 - Dec 2023
Dr Mark Bull – *University of Edinburgh* INFD11012: Threaded Programming

· Designed and wrote two separate parallel C programs which solved the adaptive quadrature algorithm; one used a recursive method while the other used a LIFO (Last in first out) method and profiled the performance for both strong scaling, weak scaling and on the Cirrus supercomputer at the University of Edinburgh
· Profiled the performance for OpenMP schedules which includes static, dynamic and guided on two separate loops with unique structures and wrote a detailed report discussing and analyzing the performance
· Achieved a B mark of 69% (Merit) overall for the module.

Predator and Prey Simulation (Report & Repository) Sept 2023 - Dec 2023
Dr Bianca Prodan – *University of Edinburgh* INFD11016: Programming Skills

· Optimised and refactored a predator and prey simulation in python which modelled the growth and decline of mice and rats using differential equations
· Wrote a report discussing the memory footprint and timing profiling results of the simulation code
· Achieved a B mark of 64% (Merit) overall for the module.

Optimising a 3D body simulation program (Report & Repository) Sept 2024 - Dec 2024
Dr Adrian Jackson – *University of Edinburgh* EPCD11009: Performance Programming

· Wrote a report using different compiler flags on a 3D body simulation and documenting the results
· Hand optimised the program by changing numerous lines of code and profiled the results on the Cirrus supercomputer at the University of Edinburgh
· Achieved a B mark of 61% (Merit) overall for the module.

ACADEMIC REVIEWING

Transactions on Parallel and Distributed Systems (TPDS)
Journal of Parallel and Distributed Computing (JPDC)
Parallel Computing (PC)

SKILLS/HOBBIES

Programming Languages	Python, C/C++, CMake, Git, Linux, MPI, OpenMP, Slurm, Rust, Zig, CUDA
Machine Learning Tools	PyTorch, TensorFlow, Scikit-learn, Pandas, NumPy
Hobbies	Reading, Writing, Chess, Thinking, Competitive Programming, Music, Maths