Emilian Joseph Bowry

07831799619 **emil.bowry@icloud.com** https://github.com/emilbowry

Education

Trinity College, University of Cambridge
BA, Engineering Tripos, 2020-2025

Note: Degree awarded unclassified due to medical intermission preventing completion of Part IIB

Judge Business School, University of Cambridge
Accelerate Cambridge, August 2022-July 2023

Experience

Software Developer AI Compatible
Remote August 2025 - Current

- **Regulatory Analysis Engine:**
  - Utilises Large Language Models for initial data extraction: Defining the output form using an OpenAPI schema object for consistently returned format for processing.
  - Improving and refining the Hader et al's methodology [DOI: 10.1007/s00607-024-01331-9] for a generative, and automatable process to gain more nuanced insight into privacy policy and reduce the amount of API calls.
  - Unsupervised semantic equivalence detection model utilises Baysian Inference, Topology, Linear Algebra, NLP techniques and Non-linear Systems analysis. Part of my system derived an analog for Random k Conditional Nearest Neighbours, a recently published technique for the classification of high-dimensional data.
- **Redeveloping full-stack website:**
  - **Frontend:**
    - Creating a dynamic background image generator to programmatically generate complex, 'tilable' background textures entirely on the client-side
    - Creating a geometry engine to compose mathematically definable UI styles, including the capability to have seamless non-uniform backgrounds over disconnected, and unconventionally shaped UI components.
    - Generalisng the CSSTypes library to be able to create more complex, type-safe style objects.
    - Simulating a toroidal topology for a scroll-bar to define well ordered cyclical UI wheel elements.
    - Direct style-sheet injection to handle high-performance style changes, by prevent re-rendering by modifying css rules and not the elements.
    - PDF generator that implements the 'ISO 32000' PDF specification to generate PDFs from java objects and/or json
    - Contextual telemetry analysis, creates interaction based profiles utilising site-entry points, device and location data to profile user events.
  - **Backend:**
    - Managing cloud infrastructure, including virtual machines and networking.
    - Devising and prototyping HTTP and API servers from scratch using Rust, to learn the fundamentals

* Managing backend API with express.js, Prisma, PostgreSQL, mongoDB.
* Handling authentication, payment processing and third-party APIs.

Co-founder Luucid.tech
Cambridge August 2022 - October 2023

- Created novel electrochemical and material mechanisms for detecting spiking agents in beverages.
- Determined product-market fit and commercial viability of scientific research.
- Sponsorship by University of Cambridge's startup incubator.

Software Development and Business Analysis Intern Atomic Media
Nottingham April 2022 - August 2022

- Built an anomaly detection system that analysis of fuel levels in a vehicle fleet to infer when there may have been an incident of fuel theft.
- Analysed new business opportunities and ventures, writing insight articles.
- Led skill days, which taught developers the low-end networking implementations of the tools they use: https://github.com/emilbowry/NetworkProgrammingLesson
- Organised the weekly cyber-security brief about emerging threats and vulnerabilities.

Published and Open Source Software

Plotting Tools: https://github.com/emilbowry/Plots
https://pypi.org/project/plottingtools-emilbowry

Extension of the python Plotly library to make 4+ dimensional correlations intuitive to the human eye, using metaprogramming techniques to create a robust and adaptable framework.

Code Editor: https://github.com/emilbowry/editor

A fork of Microsoft VSCode that:

- Improves supply chain security by removing telemetry "at the source" rather than just blocking the URL (like alternative's like 'Codium')
- Removes LLM, MCP and agetic AI integrations and bloat.
- Adds new features, like a persistent homepage, and cross-codebase note taking system

AST Debug Logger: https://github.com/emilbowry/AST-Debugger

A debugging tool that intercepts python code before execution to toggle any 'debug' flags, even from orphans and disconnected nested code. It also intercepts and saves a logs.

Algorithmic Trading resource: https://github.com/emilbowry/algorithmicTrading

A teaching resource that develops and implements knowledge gained in my signals processing, statistics, systems and other engineering courses, as well as other information I learnt from Dexter's Notes of the Mathematics Tripos. This includes appllications of:

- Analysis tools like: K-means clustering, linear regression
- Stastical tests like Augmented Dicky-Fuller, Variance Ratio, Generalised Autoregressive Conditional Heteroskedasticity, and Stationarity Analysis
- Signal Processing techniques including Fourier, Wavelet and Hilbert transforms, continuous and discrete filters.

- Optimisation techniques, it covers: linear programming, non-linear programming and convex optimisation
- Other general tools like PCA, semantic analysis, HMMs and inference tools, Metropolis-Hastings Algorithm, Decision Boundaries
- Backtesting engine

Projects and Additional Experience

Phasor Average Estimator: An estimator that models events as a phasor. Each observation adds a new phasor to the model, allowing us to estimate where we expect to see events in the future. Modelling an event as a phasor allows us to:

- Be robust to noise and jitter: Interference of the phasors allows this to cancel out in probability
- Handle missed signals: The phasors still process regardless of observing a signal or not.
- React instantly to periodicity changes: Most estimators would require some evidence accumulation period before it changes period regime, this just spins a new phasor.
- Handle false positives: A new phasor adds to the bank, adding a different frequency component, however until repeated signals are observed, this does not counteract the dominant frequencies.
- Make reasonable predictions arbitrarily far into the future, given the events we have seen.
- Does not require any exogenous and somewhat arbitrary values like damping ratio since the energy can be bound.
- Scale-Invariant: Can handle signals in arbitrary time-frames and amplitudes

The result was a Zero-Lag, Non-Parametric, Recursive Bayesian Estimator to solve issues with hardware jitter and Inter-Symbol Interference (ISI), and sensor saturation in a Molecular Communication system.

Neural Data Analysis: Built an simulation framework for Lateral Intraparietal Cortex (LIP) neuron impulses, in order to analyse stochastic models The core objective was to evaluate and test different statistical models for varying models. The framework included:

- Utilised Stochastic Differential Equations (SDEs) as a model for Neuron Impulses 'step/ramp models'
- Assesed HMM approximations of the SDEs
- Developed a ETL (Extract, Transform, Load) process for scientific data, instead of directly using the lab's given methods.
- Normalised methods to run statistical analysis on these simulations, i.e Peri-Stimulus Time Histogram, Fano-Factor.
- Centralise transformation between different data formats, like ones appropriate for histograms, spike rasters and the ability to apply smoothing and other functions to the data.
- Used Baysian Inference to fit model parameters, and quantify model mismatch and brittleness.
- Graphing capability to easily evaluate each model and the variations it has due to different parameters, including 3D graphs and mutatable plots with sliders, showing how the surface changes as we vary parameters

iCloud Find-my messaging service: A system to piggyback on Apple's "Find my iPhone" API to remotely communicate between devices without knowledge of any identifiers like IP addresses (Side-Channel Analysis). This involved reverse engineering the protocol to analyse packet payloads/and database structures without documentation.

Automated Notes Reasoning: A system that makes inferences about my course notes, given their titles. Using techniques derived from Category Theory to deriving understand from syntax and structure to verify and generate notes, infer logical gaps, and missing objects.

- Utilised Universal Properties (specifically Categorical Products) to programmatically define the existence of intersectional notes. This allowed the system to strictly enforce schema completeness by identifying missing 'product' nodes between disjoint topics.
- Modelled the relationship between Note Syntax (Grammar) and Semantic Concepts as an Adjunction. This allowed for bidirectional consistency checking: ensuring every valid grammatical title mapped to a logical object, and every logical gap could be described by a generated title.
- Utilises the note structure entirely by how they relate to these 'Type Functors' (Definitions, Equations, Methods), leading to understanding of the small note category.

The implementation necessitated reverse engineering parts of the 'Obsidian Markdown Editor', in order to have a better integration than was offered in the public API and documentation; parser design: Writing a Lexer/Parser for the custom note title grammar and knowledge graph construction: Automating the generation and verification between objects based on synr rules.

Module Type Objects: Built a parellel object system using modules as the core components in python to allow for more flexible and better controlled attribiutes.

Awards and Achievements

Goldman Sachs: Awarded a scholarship and Engineering Spring week.

Imperial College London: Awarded the President's Scholarship to Imperial College London, given to the top 112 candidates that demonstrated the "highest academic excellence at interview".

Referees available upon request