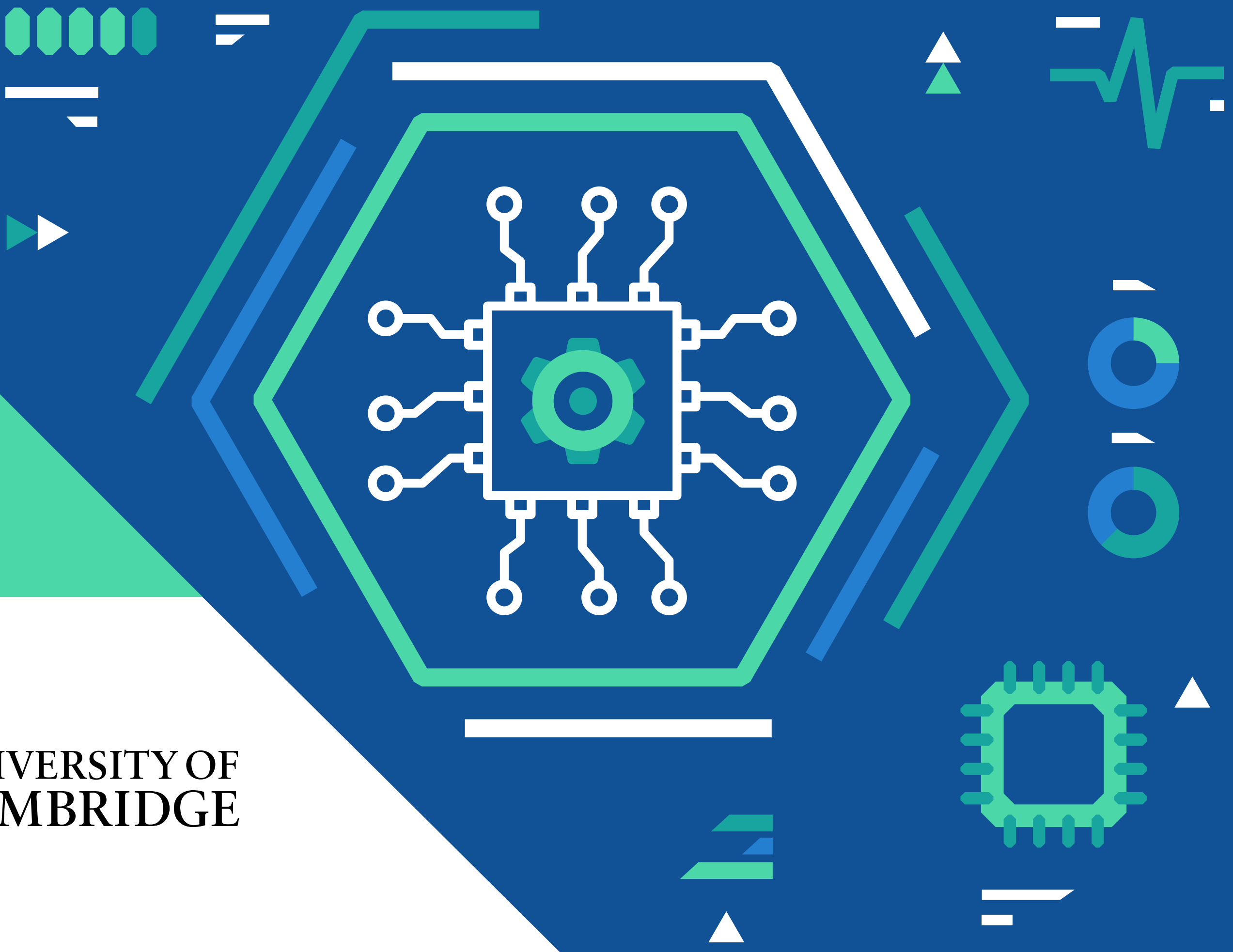
Machine learning from innovation to deployment

A strategic research agenda for AutoAI



UNIVERSITY OF CAMBRIDGE

designbysoapbox.com

Table of contents

Summary

While excitement about the potential of artificial intelligence (AI) technologies continues to build, a gap is emerging between our aspirations for the benefits of AI and our ability to deploy these technologies to tackle real-world challenges. A significant proportion of attempted AI deployments fail, highlighting a suite of practical issues that arise when trying to integrate AI into real-world systems – from data management and use, to model performance, to user experience. Such failures not only hold back the economic potential of AI, they also expose individuals, communities and societies to new forms of harm.

Achieving the full potential of AI – and its benefits for society and the economy – requires the ability to safely and effectively deploy AI systems at scale. As those deploying AI technologies seek to tackle more sophisticated tasks, AI systems are becoming larger and more complex. This complexity gives rise to challenges in the interpretation, explanation, accuracy, and fairness of AI systems. These challenges lie behind many of the failures we see today. They stem from the connection between system complexity and new types of technical and intellectual debt – our ability to deploy complex decision-making systems has increased, but further work is now needed to manage performance in deployment and scrutinise how these systems operate in practice.

Tackling these issues requires fresh approaches to system design that can manage the complex interactions that arise between machine learning components in a decision-making system. AutoAI offers this new perspective. It proposes a pathway to connecting machine learning and AI system design through AI-assisted techniques for system monitoring and maintenance.

The AutoAI Programme at Cambridge Computer Lab scales our ability to deploy safe and reliable AI solutions, driving innovation in machine learning architectures and techniques for deploying, maintaining, understanding, and redeploying AI systems. By investigating how AI systems can be decomposed into their component parts, how data availability and use can be more effectively managed in the development of AI, and how performance in deployment can be monitored, AutoAI will develop a new AI design and engineering paradigm.

Experience of embedding machine learning in large-scale industrial applications has identified integration of machine learning components in established systems as a major point of friction. Better integration requires research advances to develop (1) more sophisticated machine learning models that can be integrated with the wider system; (2) new software architectures and programming techniques to solve technical challenges at scale; (3) a shift towards data-oriented technical infrastructures and organisational practices; and (4) wider organisational, legal and policy interventions to influence AI adoption. AutoAI is addressing the entire pipeline of AI system development, from data acquisition to decision making. In so doing, it is developing policies that enable trustworthy access to data, building effective research collaborations that embed stakeholder needs in research design, and promoting an open culture of AI and software engineering.

This document sets out a strategic research agenda for the AutoAI Programme. It describes the core research themes and activities that the Programme supports, and provides a roadmap for the development of AI-assisted system design and monitoring tools.

AutoAI research agenda

Data-oriented architectures

- Understand basic principles of data-oriented systems and format a common vocabulary around such systems;
- Identify and lay-out the basic components of data-oriented architectures in streaming format;
- Introduce hypothetical streams for automated model declaration; and
- Integrate graphical structures for monitoring data flow.

Data readiness and provenance

- Identify the actions needed to help make organisations 'data ready';
- Develop data maturity assessments to promote common understandings of data quality issues;
- Design tools to automate record matching and data deduplication; and
- Automate tools for data validity assessment.

Data governance and stewardship

- Investigate what failure modes emerge from the deployment of machine learning, and what points of intervention can be leveraged to correct these failures;
- Progress discussions around personal data sharing, with a particular focus on the development of data trusts;
- Create AI methods that are responsive to the evolving regulatory and policy environment; and
- Investigate the role that AutoAI can play in helping ensure the FITness of an AI system and support compliance with regulatory requirements.

System design and continual system meta-learning

- Design 'shadow systems' that can be used to explore counterfactual explanations about how a system works and test different deployment strategies;
- Investigate how AI-enabled monitoring can help users understand performance in deployment, the reasons for any performance changes, and optimise system functionality (for example, optimising resource usage through targeted training on less data); and
- Integrate concepts from control theory and continual learning to develop novel approaches to maintaining optimal system performance.

End-to-end system optimisation

- Investigate how requirements form machine learning models and algorithms change when they are embedded in a wider system, and what metrics are needed to characterise whether a model is correct or useful;
- Develop methods to represent uncertainties in hierarchical and multi-component systems;
- Develop techniques for counter-factual emulation, exploring what features of a system should be emulated, how, and the interactions between emulation and end-to-end learning; and
- Connect emulation and Bayesian Systems Optimisation for monitoring of whole-system performance.

Information dynamics, infrastructure, and programming at scale

- Simulate feedback loops using emulation and data-oriented architectures;
- Monitor instabilities in information dynamics and propose interventions to resolve them; and
- Identify the causes of socially destructive attractors in dynamical systems.

1

AI for real-world challenges

Creating safe and reliable AI systems

In 1987, more than a decade after the invention of the personal computer, economist Robert Solow concluded that "You can see the computer age everywhere but in the productivity statistics".[1]

Solow's paradox described the disconnect between the apparent technological strength of the computing revolution and continuing stagnation in the US economy. Despite the promised benefits of the personal computing revolution, the ability of individuals and organisations to deploy available computing technologies lagged the technical capabilities on offer.

These lags between invention and wider social or economic benefit can be seen throughout the history of innovation. From the steam engine to microprocessors, history shows that the pathway from innovation to deployment and diffusion involves not only technological innovation but wider changes to individual behaviours, social structures, organisational cultures, and national policies, as human activities reorganise around technology.[2]

Today, artificial intelligence (AI) technologies offer the possibility of a new wave of automation. AI-enabled decision-making systems can automate a growing range of processes, taking on increasingly complex tasks. In contrast to previous waves of technology change, the apparent promise of AI is that it will be the first generation of automation in which machines will adapt to human needs, rather than requiring humans to make adjustments for new machines.[3]

However, while excitement about the potential of AI continues to grow, practical experience of deploying these technologies in service of real-world problems highlights a disconnect between the field's potential and its ability to deliver real-world benefits. A significant proportion of attempted deployments fail, exposing implementation challenges across all stages of the deployment workflow – from data management, to model

[1] Solow, R. (1987) We'd Better Watch Out, New York Times Book Review, 12 July 1987

[2] Royal Society (2018) The impact of AI on work: implications for individuals, communities and society, available at https://royalsociety.org/topics-policy/projects/ai-and-work

[3] Lawrence, N.D. (2020) The Great AI Fallacy, http://inverseprobability.com/talks/notes/the-great-ai-fallacy.html

training and verification, to user experience.[4] Such failures not only hold back the economic potential of AI, they also expose individuals, communities and societies to new forms of harm.[5]

Despite these failures, ambitions for AI continue to grow, as organisations seek to deploy AI technologies to tackle more sophisticated tasks. In the process, AI systems are becoming larger and more complex. As their complexity increases, a gap has emerged between the ability to rapidly deploy an AI system and the ability of those working with, or affected by, the system to interrogate and understand how it works. This has implications for the safety and efficacy of AI systems 'in the wild'.

Achieving the full potential of AI – and its benefits for society and the economy – requires the ability to deploy AI systems safely and effectively at scale. This, in turn, demands fresh approaches to systems design, combining insights from cutting-edge machine learning research with experiences of real-world deployment. AutoAI is our response to these challenges.

This document

'AutoAI' is a collection of tools and methods that will facilitate automated monitoring and adjustment of AI systems deployed in real-world contexts. Many of the core concepts and approaches that underpin its development have a long history in computer science and associated fields. AutoAI extends, integrates, and innovates with these concepts to deliver a new approach to AI system design.

This document introduces the rationale behind the AutoAI Programme. It sets out those limitations in current AI systems that AutoAI seeks to address, it introduces the design features that AutoAI seeks to embed in the development of AI systems, and it highlights some of the Programme's current areas of research interest. To help contextualise these developments, it also introduces use cases that explore how AutoAI might contribute to real-world AI systems.

Much of the recent excitement surrounding AI has been sparked by advances in machine learning. Machine learning is a technology that enables computer systems to learn from data. It combines data with mathematical models through computation to make predictions. In this document, the term 'machine learning' is used to describe the models used as a sub-component of a wider AI system. Those models produce a prediction or decision that contributes to a larger output. Such units, as discussed later, are often integrated into wider systems – of other AI technologies, classical software, other mathematical models, organisational processes, and human interactions – the collation of which we refer to as as 'AI systems'.

[4] Paleyes, A., Urma, R.G. and Lawrence, N.D. (2022) Challenges in Deploying Machine Learning: a Survey of Case Studies. ACM Comput. Surv. https://doi.org/10.1145/3533378

[5] Partnership on AI (2020) When AI Systems Fail: Introducing the AI Incident Database, www.partnershiponai.org/aiincidentdatabase

2

AutoAI

A new paradigm for
AI design and deployment

Advances in machine learning over the past decade have produced a suite of algorithms that can achieve high levels of performance when trained to carry out specific tasks.

Those tasks are, however, restricted: they are tightly defined and constricted in ways that do not readily map onto the level of sophistication required for many real-world activities. Consequently, any AI solution in deployment will normally take the form of interacting components, each component performing a specific task as part of a wider AI system that completes a more complex activity. In a supply chain, for example, machine learning models might be responsible for generating predictions about demand and supply of different goods. These models might feed into operations research (OR) models that make decisions around optimal stock levels given those predictions. Together these components match expected supply to expected demand, while considering constraints such as available warehouse space and customer service targets. With more data available from different sources, the complexity of these composite systems increases, as new components are added.

When tackling more sophisticated tasks, such as autonomous vehicle control, developers design more complex AI systems. This complexity introduces new challenges that must be addressed to ensure the safe and effective operation of AI in deployment. These challenges emerge in the form of technical and intellectual debt.

The concept of 'technical debt' has its roots in the development of software systems in the 1990s.[6] It describes the challenge of building systems that are maintainable in production, without requiring significant additional labour.[7] Technical debt arises from the tension between the desire to deploy a system quickly and the time taken to design code that is robust in deployment – rapid deployment can meet a near-term need, but can result in systems that require amending later to correct for unforeseen issues.

[6] Cunningham, W. (1992) The WyCash Portfolio Management System, OOPSLA '92 Experience Report, http://c2.com/doc/oopsla92.html

[7] Lawrence, N.D. (2020) AutoAI and Machine Learning Systems Design, https://inverseprobability.com/talks/notes/auto-ai-and-machine-learning-systems-design.html

Technical debt can be accrued in different ways in an AI system. The ways in which machine learning components interact with relevant data, the connections between these components, or changes to the external environment can all contribute to maintenance problems that must be addressed if the AI system is to continue to perform effectively.[8]

As technical debt accrues, it becomes more difficult to *maintain* the software system. Intellectual debt is a related but different concept. When intellectual debt accrues, it becomes more difficult to *explain* important aspects of the system, including how it works.[9] As the complexity of AI systems in deployment has increased, methods to explain how these systems work have not kept pace. The complexity of today's AI systems leaves their workings opaque to most users. While their performance can be validated using statistical tools, answering questions about why the system has produced a particular result is increasingly challenging. Intellectual debt accrues due to the complexity of the interactions between data, algorithmic components, and the wider environment that the AI system operates within.

Resolving intellectual debt has become important as AI is increasingly integrated in systems that influence daily life or routine activities. Recent years have seen a range of examples of AI system failures contributing to safety incidents, discrimination, or other forms of harm to individuals, groups, and organisations. Technologists and policymakers alike require novel strategies to prevent these harms emerging, and ways of scrutinising how AI systems are operating.

The field of software systems design has already developed new ways of thinking in response to the challenges of technical debt. Similarly, today's AI systems demand a new technical ecosystem to ensure their effective performance in deployment that is maintainable and explainable. While recognising that technical interventions alone are not sufficient to address concerns around governance or accountability of AI systems, AutoAI offers a response to this demand for a new paradigm in machine learning systems design. Through AI-assisted design and monitoring of AI systems, AutoAI can help ensure that AI solutions perform robustly, safely and accurately in their deployed environment.

Creating this new paradigm requires fresh approaches to systems and machine learning engineering across the entire pipeline of AI system development, from systems design to systems deployment, including specific tasks related to the lifecycle of machine learning components (for example, data acquisition and decision-making). It requires technical interventions

[8] Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.F., and Dennison, D. (2015) Hidden Technical Debt in Machine Learning Systems. In Advances in Neural Information Processing Systems 28, edited by Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M. and Garnett, R. 2503–11. Curran Associates, Inc. http://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems.pdf

[9] Zittrain, J. (2019) Intellectual Debt: With Great Power Comes Great Ignorance. https://medium.com/berkman-klein-center/from-technical-debt-to-intellectual-debt-in-ai-e05ac56a502c

to boost performance and ensure that AI systems operate in harmony with social values and regulatory requirements, while ensuring that these interventions operate in a wider context that considers the availability, quality and ethical use of data.

This report presents a research agenda for AutoAI. It proposes a way of designing technical architectures to support safe and effective AI systems, and identifies research areas where innovations in systems design and machine learning can create more effective tools for deploying, maintaining and understanding AI.

3

Putting systems at the heart of AI deployment

The three D’s of AI systems design

AI systems adapt to the context in which they are deployed. It is the evolving nature of these systems that demands new approaches to model development, with the aim of ensuring that AI systems operate reliably under changeable real-world conditions.

To respond to these conditions effectively, system design must take into account three features:

- decomposition of the real-world challenge into separate tasks that are addressable with a machine learning solution – ***decomposition***;
- collection, curation and stewardship of appropriate data – ***data***; and
- monitoring and maintenance of performance and quality in deployment – ***deployment***.

The sections that follow consider the role of each of these '3 D's of system design' – decomposability, data, deployment – in achieving safe and reliable AI solutions.

Decomposability

Identifying points of automation

Any repetitive task is a candidate for automation, but many of the repetitive tasks performed by humans are more complex than any individual algorithm can replace. The selection of which elements of a task to automate and how to use machine learning for such automation are important design choices that have significant downstream consequences for the performance of an AI system. The design process requires the overarching goal to be broken down into smaller, automatable, components and the designer to consider how the performance of each component is dependent on those upstream of it, before determining how individual components might be automated through machine learning. This is the challenge of decomposition in AI systems design.

Task selection is influenced by both what technical capabilities are available and the overall system objectives, taking into account:

- the extent to which the inputs and outputs of each component can be defined and represented mathematically, making them amenable to automation;
- what machine learning algorithms are available, and whether these are suitable for the tasks at hand;
- how different components interact with each other and influence the overall performance of the system;
- how the proposed task split affects user requirements, for example in relation to accuracy, interpretability, or fairness; and
- what data is available for use in the system.

Managing interactions between components

While the process of automation requires decomposition of the system into individual tasks, the performance of each of those individual components is dependent on those upstream of it. These interlinkages lead to co-evolution of systems, as upstream errors are compensated by downstream corrections. In logistics and supply chain, for example, a website might show an initial plan for delivery times, computed when an item is viewed, based on waiting times for that item; however, when an order is placed the constraints on waiting times may be replaced by constraints around cost. Such sub-systems might make inconsistent decisions, and part of the role of system design is to monitor and control the extent of the inconsistency.

End-to-end learning is one approach to managing this dynamic. Instead of optimising the performance of each component independently, then fitting them together, end-to-end learning seeks to produce the optimal outcome for the overall system, using machine learning techniques to adjust operational parameters across the entire decision pipeline. Another alternative is to replace the entire system with a single machine learning model, using methods such as deep reinforcement learning.

These approaches can improve performance, but have implications for how users can manage the resulting system. Learning across the system reduces its decomposability – it becomes more difficult to assess the performance of individual components – which in turn affects the extent to which human users can interrogate which parts of the system played what role in delivering an outcome. As a result, the system becomes less interpretable, making it more challenging to diagnose faults when issues arise and more difficult to adapt the system to changing circumstances. This creates trade-offs between interpretability and other aspects of performance, with implications for overall system performance and its interactions with its users.

AutoAI proposes an alternative approach, taking advantage of end-to-end learning strategies, while introducing technical innovations that do not sacrifice the interpretability of the overall system.

Designing for decomposability: the AutoAI approach

Instead of optimising each component of the system individually, AutoAI proposes to connect the performance of individual components to the performance of an overall system trained using end-to-end learning. These connections can be made through a network of surrogate models, called emulators, in which each emulator provides a representation of how an individual system component is operating. These representations are aggregated, analysed using Bayesian Systems Optimisation[10] to optimise the system's performance, and their results then used to make adjustments to the real system.

In this approach:

- the designer decomposes the challenge at hand into components that can be automated;
- surrogate models (emulators) emulate the performance of each component, each surrogate model being automatically designed and deployed according to the question asked of the system; and
- outputs from those surrogate models feed into a process of Bayesian System Optimisation that analyses how to optimise the performance of the overarching system.

This allows the designer to exploit an end-to-end learning strategy, but to optimise the system via surrogate models before implementing real-world changes. The system is made more interpretable through the design of the emulators. Emulators can be constructed to answer a specific question, creating a diagnostic tool that can be used to understand how the system is working. Those using the system would be able to interrogate why a particular output had been produced by asking different questions of these emulators, paying back the intellectual debt associated with system complexity.

Because a large number of questions might be asked in a complex system, a potentially large number of different types of emulator might be needed; a larger number than could be crafted 'by hand'. To overcome this challenge, designers need to automate the construction of emulators. By learning which emulators are suitable for different questions – transferring learning between emulators – such automated systems could exploit the knowledge from pre-existing emulators to produce and deploy new emulators as needed; a process of 'deep emulation'.

[10] The AutoAI Programme characterises Bayesian Systems Optimisation as an approach to system optimisation that makes use of end-to-end learning signals and attributes them to system sub-components through the construction of an interconnected network of surrogate models. For examples of related work, see: Dalibard, V., Schaarschmidt, M. and Yoneki. E. (2017) BOAT: Building Auto-Tuners with Structured Bayesian Optimization. In Proceedings of the 26th International Conference on World Wide Web (WWW '17). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 479–488, https://doi.org/10.1145/3038912.3052662 and Alabed, S., & Yoneki, E. (2021). BoGraph: Structured Bayesian Optimization From Logs for Systems with High-dimensional Parameter Space. arXiv:2112.08774

Building and using emulators

To date, most work on emulators has focussed on emulating a single component/model. Exploratory projects led by the AutoAI team are advancing understandings of how to use emulators in structured settings. The first of these explores how constrained Bayesian optimisation problems, such as finding scientifically plausible model parameterisations that conform to past real-world observations, can be formulated as a hierarchical model with informative uncertainties. In other words, we need to represent how uncertainty propagates through the decision making system. How sensitive are the later decisions to assumptions made in earlier decisions? In such a model, optimal decision-making can be shown to be equivalent to information theoretic reasoning about the distribution of possible good decisions – i.e. that data-driven models conform with pre-existing knowledge or theories about how the world works. The second project investigates how auto-encoding neural network models can be interpreted as deep Gaussian processes to automatically determine the minimal required model complexities for informative latent spaces. These models reduce the complexity of problems to a few parameters. The aim is that these parameters are more interpretable to human operators. These models assess the minimal complexity required. A further research effort is using emulations to improve agent performance in high-fidelity simulations. High fidelity is needed for accuracy of simulation, but comes with significant computational costs. If the behaviour of the simulation can be accurately captured by the emulation, then agents' can hone their performance without the computational costs.

Automating the process of constructing and deploying families of emulators across a full system – creating networks of emulators that can be deployed and redeployed on demand – is an ambitious task. It requires tools for the rapid retraining of emulators and a sophisticated understanding of the propagation of uncertainty through the system, based on advances in machine learning engineering, emulation techniques and Bayesian optimisation. It will also require methods that can integrate the outputs from surrogate model networks to re-create a full view of the resulting system. To that end, the AutoAI group maintains Emukit, a software tool for emulation and decision-making under uncertainty.

Emukit is a software framework that facilitates the programming of decision-making systems via the use of surrogate modelling and emulation.[11,12] Its purpose is to provide a set of techniques for managing uncertainty in decision-making, particularly in domains of low data availability.[13]

[11] Paleyes, A., Pullin, M., Mahsereci, M., McCollum, C., Lawrence, N.D., & González, J.I. (2019). Emulation of physical processes with Emukit. arXiv:2110.13293 [cs.LG]

[12] Emukit's origins lie in work by Javier Gonzalez at the University of Sheffield to build software for Bayesian optimisation. Javier Gonzalez and Andrei Paleyes then developed the full Emukit package while working at Amazon. As well as Javier Gonzalez (ML side) and Andrei Paleyes (Software Engineering) the build team included Mark Pullin, Maren Mahsereci, Alex Gessner, Aaron Klein, Henry Moss, David-Elias Künstle as well as management input from Cliff McCollum and Neil Lawrence. See: https://emukit.github.io

[13] For further information, see: https://emukit.github.io/about

To support the workflow of deploying emulators, Emukit offers access to: probabilistic representations of models being used; interfaces to understand what uncertainties are associated with the model; and the ability to ask questions about specific tasks within a system.[14]

Emulators can be used as an effective tool for real-time decision-making. However, when integrating them into a system that will include other models or software, it is important to consider the computational costs of the full system. These costs could arise from simulation, optimisation, or even – for large models – model inference. These costs have to be considered when deploying statistical emulations. It is important to study how algorithms perform in real-time settings, looking at the computational costs of a model as part of properly evaluating the performance of a controller.

Data

Data as a source of technical debt

The ‘software crisis’ of the 1960s and 1970s describes a period in which – despite technical advances in the complexity of computer systems – progress in computer science appeared to stall. This stalling was attributed to the inability of researchers to deliver software solutions that could manage the increasing complexity of the systems into which new software was being implemented. As described by Dijkstra (1972):

> “The major cause of the software crisis is that the machines have become several orders of magnitude more powerful! To put it quite bluntly: as long as there were no machines, programming was no problem at all; when we had a few weak computers, programming became a mild problem, and now we have gigantic computers, programming has become an equally gigantic problem.[15]

In response, software engineering developed new standards and practices to manage this complexity. One modern approach to software systems design is known as a *service-oriented architectures* (SOA). Under this approach, software engineers are responsible for the reliability of the interfaces that others use to access the service they own. The quality of their service is maintained through rigorous standards and testing of software systems.

With data at the heart of today’s machine learning tools, and with machine learning algorithms using this data to drive their decision making, data has become the new software. An important challenge for the field of AI is addressing the resulting data crisis.

[14] For further information, see lectures on Machine Learning and the Physical World by Lawrence, N. D. and Ek, C.H. (2021) at: https://mlatcl.github.io/mlphysical/lectures/04-02-emukit-and-experimental-design.html

[15] Dijkstra, E.W. (1972) The humble programmer. Commun. ACM 15, 10 (Oct. 1972), 859–866. DOI: https://doi.org/10.1145/355604.361591

The modern data landscape is characterised by happenstance data. Classical statistics assumes data is collected with a particular hypothesis in mind. Happenstance data arises today because reduced costs of storage, compute and communication mean that more and more data is recorded without a particular purpose. Unfortunately the quality of this data is often poor. The lack of attention paid to its curation when it is initially collected, and the costs of data cleaning to fix problems, contribute to the overheads associated with creating AI systems. Anecdotally, when applying machine learning methods in practice, practitioners spend 80% of their time on data cleaning. This cleaning work is often not shared; it is repeated by different teams each time they wish to make use of a data stream, adding to the time and expertise required to operationalise machine learning. These 'set-up' costs are compounded by the paucity of tools for monitoring how data is used or for tracking the provenance of data used by models when in deployment. Together, they increase the technical debt associated with running an AI system in the real world.

Both cultural and infrastructural changes are needed in response, through interventions that refocus AI development around data, that encourage organisations to better understand the value and quality of their data, and that build a wider policy environment that facilitates trustworthy data use.

Designing for data: the AutoAI approach

A Data-oriented architectures

In data-driven decision-making systems, the quality of decision-making is determined by the quality of the data. When consuming data from others, developers cannot assume that it has been produced in alignment with their needs or to the quality standards that they require. To excel in data driven decision making, changes are needed to move from a *software first* paradigm to a *data first* paradigm; from service-oriented architectures to data-oriented architectures. Without such a shift, machine learning systems in deployment are likely to accumulate a range of failures associated with declining data quality, incorrect modelling assumptions, and inappropriate redeployment of models.

Making the shift to data-oriented architectures requires reorienting developer teams around data as their primary product. Software services remain important, but are intermediary in producing that data, and the quality of the data must be monitored. Activities such as data cleaning and maintenance need to be prized as highly as software maintenance is today. Currently, software engineers focus on the quality of the data their systems ingest, as poor-quality data can damage the quality of their service. This means that each data stream is being separately checked for quality by different software teams. Data-oriented architectures require a shift of focus, and promote data coupling between software components.

In data-oriented architectures engineering teams are responsible for the quality of their output data flows – implemented, for example, with streaming platforms such as Apache Kafka or other data storage mediums. For raw data this implies recording fidelity and provenance. For data arising from machine learning models this implies monitoring the model, not only in terms of accuracy, but also in terms of wider quality issues relating to fairness and explainability. These data streams would then be consumable by many teams without needing to be separately cleaned by each team needing to make use of that data. To assess the quality of such outputs, new forms of testing will be necessary. These tests would assess quality, fairness and consistency within the data environment, making use of automated tools for the creation and deployment of such tests across the ecosystem.

Projects that successfully deploy machine learning in real life recognise the importance of data. They design software with data as a highest priority. For instance, focus on data is a characteristic of Flow-Based Programming (FBP)[16] – a programming paradigm that represents a system as a data flow graph and promotes data coupling between components. Recent work by the AutoAI team has contrasted FBP with the currently prevalent SOA[17] paradigm for building data-processing applications and for machine learning deployment. This work concluded that FBP is a suitable paradigm for these contexts and offers some advantages over SOA, but requires more mature and rich tooling to operationalise.[18]

New software tools to manage data streams are already emerging. These offer a platform on which to build a data-oriented architecture. Further work is now needed to assess their suitability for machine learning in deployment and to develop the tests that can help users better understand the quality of their data in these streaming services.

B Data readiness

Many of the changes required to implement data-oriented architectures are cultural or organisational:

- → to establish a data-first engineering approach, teams need to decompose systems around the data generating and consuming components, instead of the software components;
- → to enable communication between teams about their data outputs, teams need to be able to accurately describe the quality of different datasets;
- → to prepare data and formulate appropriate questions for analysis, teams need new skill sets, in addition to the traditional skills of software engineers or applied scientists; and
- → to assess the quality of such outputs, new forms of testing are necessary.

[16] Morrison P. J. (1994) Flow-based programming. Proc. 1st International Workshop on Software Engineering for Parallel and Distributed Systems

[17] Perrey, R. and Lycett, M. (2003) Service-oriented architecture. Symposium on Applications and the Internet Workshops

[18] Paleyes, A., Cabrera, C. and Lawrence, N.D. (2021) Towards better data discovery and collection with flow-based programming. Data-centric AI workshop, NeurIPS 2021. arXiv:2108.04105v2 [cs.SE] and Paleyes, A., Cabrera, C. and Lawrence, N.D. (2022) An Empirical Evaluation of Flow Based Programming in the Machine Learning Deployment Context. CAIN 2022, arXiv:2204.12781 [cs.SE, cs.LG]

Foundational to these changes is the creation of a common language around what data quality means in practice. Such a language would be the baseline for planning for new systems, estimating the cost of data cleaning, and building wider understandings of the value of data and data curation.

Data Readiness Levels[19] are an attempt to develop a language around data quality that can bridge the gap between different communities of technical developers and decision makers, such as managers and project planners. Inspired by the concept of Technology Readiness Levels – which quantify the readiness of technologies for deployment – Data Readiness Levels describe the amount of effort required to prepare datasets for analysis. They suggest three grades of data readiness:

- grade C data is *hearsay* data. Data that is purported to exist, but has not been electronically loaded into a computer system;
- grade B data is available electronically, but requires a process of validation before it can be used. This validation might be required to address issues such as like missing values, outlier representation, duplicate records,[20] or to resolve legal or ownership issues; and
- grade A is then data in context, at which point developers can consider the appropriateness of data to answer a particular question.

The team explored the use of data readiness levels in collaboration with other scientists and software engineers in delivering an operational science agenda for policy advice during the Covid-19 pandemic.[21]

C Data policy and ethics

In addition to these technical considerations, those deploying AI systems must grapple with questions about what data can be used for what purpose, and the trustworthiness of their practices for data access and use.

AI systems often use sensitive personal data, and an important challenge is how to extract insights from this data without infringing individual rights. Existing legislation creates a constellation of rights and responsibilities around data use. The EU's General Data Protection Regulation, for example, creates a 'right to an explanation' that gives individuals the right to access 'meaningful information about the logic involved' in automated decisions.[22]

[19] Lawrence, N.D. (2017) Data Readiness Levels, available at: http://inverseprobability.com/publications/data-readiness-levels.html

[20] Grade B also has some of the characteristics of *exploratory data analysis*, as explored in Tukey, E. (1977) Exploratory Data Analysis. Addison-Wesley, 1977. ISBN 0-201-07616-0

[21] These grades are explored further in: The DELVE Initiative (2020), *Data Readiness: Lessons from an Emergency*. DELVE Report No. 7. Published 24 November 2020. Available from https://rs-delve.github.io/reports/2020/11/24/data-readiness-lessons-from-an-emergency.html; https://rs-delve.github.io/assets/pdf/2020-11-24-data-readiness-lessons-from-an-emergency.pdf

[22] For further information, see: https://ec.europa.eu/info/law/law-topic/data-protection/data-protection-eu_en

Emerging regulatory frameworks and toolkits point to the variety of organisational, cultural, and technical interventions that are necessary to align AI systems with legislative requirements.[23] Core to these is the ability to understand how data is used in a system. By providing technical architectures that allow the use of data to be tracked and methods that enable scrutiny of how a system operates, AutoAI offers a route to helping fulfil these requirements.

Despite much recent progress in establishing policy and legislative frameworks for data and AI, an implementation gap persists between the goals set out in such frameworks and the extent to which individuals or groups can meaningfully influence who accesses and uses data about them, and for what purpose. The complexity of changing patterns of data use, the continuing reliance on individual consent as the basis for data use, and the time, energy and resources required to pursue legal intervention in the case of data mis-use all contribute to power asymmetries in the digital environment.

To counter these forces, new institutions are needed to enable data sharing for social benefit while protecting individual rights and freedoms, and while embedding social values in data use.

Data trusts are a novel data institution that are designed to rebalance the power asymmetries between data controllers and data subjects, by integrating data trustees into the negotiations about the terms of data use. A data trust is a mechanism for individuals to pool the data rights created by current legislation into an organisation – a trust – in which trustees make decisions about data use on their behalf. These trusts have received widespread attention from policymakers over the last five years, but significant questions remain about how to implement the ideas they propose. These questions form the basis of an interdisciplinary programme of research and practice, which is the current focus of the Data Trusts Initiative[24] which the AutoAI team has convened in collaboration with Professor Sylvie Delacroix of the University of Birmingham.

Deployment

Once the decomposition is understood, the data is sourced and the models are created, the model code needs to be deployed. When deployed, the resulting system must respond to a range of situations that fall outside the scope of the data on which it was trained. As a result, when any data dependent model moves into production, it requires continuous monitoring to ensure the assumptions of design have not been invalidated.

To better understand the landscape of challenges practitioners face, the AutoAI Programme has investigated issues that present at each step of the deployment pipeline, from data collection and model training

[23] This includes emerging EU legislation on AI (see: https://digital-strategy.ec.europa.eu/en/policies/european-approach-artificial-intelligence) and national-level regulations or guidance (for example, see: https://ico.org.uk/about-the-ico/news-and-events/ai-auditing-framework)

[24] For further information, see www.datatrusts.uk

to quality assurance and system monitoring. A review of machine learning in deployment identified 38 issues and challenges that span every stage of the machine learning pipeline, finding examples of these challenges from across fields and industries.[25] Drawing from this work, AutoAI is developing approaches to increase the effectiveness of machine learning in deployment.

A Performance monitoring in deployment

Verification and validation of system performance is an active area of research and practice. Software changes are often qualified through testing to ensure that existing functionality is not broken by change – known as 'regression testing'. Since data is continually evolving, machine learning requires 'continual regression testing' – oversight by mechanisms that ensure their existing functionality has not been broken as the world evolves around them.[26]

Many of today's continuously deployed systems already rely on testing, but do not yet adequately account for the continuous evolution of the world around them. Further development in best practice around model deployment is needed, and AI-enabled performance monitoring offers an opportunity to fill this gap.

By creating an infrastructure that allows continual monitoring and adjusting of models in deployment, AutoAI can help maintain the performance of AI systems in operation. The networks of emulators that help optimise the system can act as a bridge between the system and real-world conditions. Emulators, along with other highly effective methods for monitoring AI systems,[27] can detect changes in the external environment or in the performance of individual system components, thus acting as a 'shadow system', analysing the consequences of different changes and making adjustments to the relevant system components as needed.

These emulators act as a hypervisor system – overseeing the operation of the machine learning models that comprise the AI system to maintain performance. These hypervisors would analyse the context in which models are deployed, recognise when deployed circumstances have changed, and flag to system users whether models need retraining or restructuring. This system would allow for rapid assessment of the quality of incoming data streams and the overall functioning of the system.

[25] Paleyes, A., Urma, R.G. and Lawrence, N.D. (2022) Challenges in Deploying Machine Learning: a Survey of Case Studies. ACM Comput. Surv. https://doi.org/10.1145/3533378

[26] An approach we refer to as *progression testing* following Diethe, T., Borchert, T., Thereska, E., Balle, B. and Lawrence, N.D. (2019) Continual learning in practice. Presented at the NeurIPS 2018 workshop on Continual Learning. arXiv:1903.05202 [stat.ML]

[27] For a review of such methods, see: Klaise, J., Looveren, A.V., Cox, C., Vacanti, G., & Coca, A. (2020). Monitoring and explainability of models in production. arXiv:2007.06299 [stat.ML]

Automated deployment and maintenance requires networks of emulators that can be deployed and redeployed on demand depending on the changes at hand. Achieving this goal will require innovations in the mathematical composition of emulator models.[28] Different chains of emulators will need to be rapidly composed to make predictions of downstream performance. This requires rapid retraining of emulators and propagation of uncertainty through the emulation pipeline via deep emulation.

B Fairness, interpretability and transparency of AI systems

With growing understanding of the impact that AI failures can have on individuals and communities – and in the context of legislative efforts aimed at mitigating these adverse effects – there has been much recent interest in machine learning research about the fairness, interpretability and transparency of machine learning models. This research seeks to understand the differential impact of machine learning models on different groups, the extent to which different user groups can access meaningful explanations about why a particular prediction or decision has been made, and the interventions that are required to embed social values such as fairness and accountability in environments where AI is deployed. Less attention, however, has been paid to how these concepts of fairness, interpretability and transparency – FITness – relate to the performance of wider AI systems.

When taking a system perspective, the core issue lies not in examining the impact of individual models, which can often be validated using existing statistical tools. The long-term challenge lies in tackling the complex interactions between different components in the decomposed system, where the original intent of each component may not be clear, and when different components may have been repurposed for different tasks over their lifetime. Addressing this challenge requires a shift in thinking – moving from FIT models to FIT systems.

Creating such systems will require the technical foundations discussed above. Decomposability can help users interrogate what factors influenced a decision; data-oriented architectures can help manage data flows in line with regulatory requirements around the use of sensitive personal data; deep emulation and hypervisors can monitor system performance and ensure it adheres to defined standards. It will also require integration of domain, machine learning and systems expertise in the design of AI systems, and careful engagement with affected user communities.

[28] Damianou, A. and Lawrence, N.D. (2013) Deep Gaussian processes. In Carlos Carvalho and Pradeep Ravikumar, editors, Proceedings of the Sixteenth International Workshop on Artificial Intelligence and Statistics, volume 31, pages 207–215, AZ, USA, 4 2013. JMLR W&CP 31. Perdikaris, P., Raissi, M., Damianou, A., Lawrence, N.D. and Em Karnidakis, G. (2016) Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling. Proc. R. Soc. A, 473(20160751), 2017. doi: 10.1098/rspa.2016.0751

4

Outline use cases

The sections that follow consider hypothetical use cases where AutoAI could play a role in improving AI's performance in deployment.

Rider allocation services

Boda Bodas are motorcycle taxis that can be found across East Africa. They transport people and goods, particularly within cities, and provide a source of income and employment. Systems such as Safeboda – a Kampala-based service – provide a ride allocation system for Boda Boda drivers. Safeboda aims to increase the accessibility of Boda Boda services to customers, to boost the income of the Boda Boda riders, and to improve the safety of those using and delivering Boda Boda services, driven by the knowledge that road accidents are set to match HIV/AIDS as the highest cause of death in low/middle income countries by 2030.[29]

In this application, machine learning can play a role in optimising pricing and Boda Boda availability, within a technical environment that – at a system level – tries to ensure efficient and fair matching of jobs to Boda Boda drivers. The overarching system needs to have minimal operational load; it should be deployable and maintainable by a small team without requiring intensive technical attention.

A user logging on to the app is notified about drivers in the local area, with an estimate of the time a ride may take to arrive. Given this information about driver availability, that user may choose to enter a destination. This destination can be used to generate a price estimate. This price may be conditional on which riders wish to travel in that direction, but an estimate needs to be provided before the user agrees to the ride and customer service constraints dictate that this price should not change after the order is confirmed.

In even simple versions of such a platform, several decisions are being made in parallel. These decisions relate to:

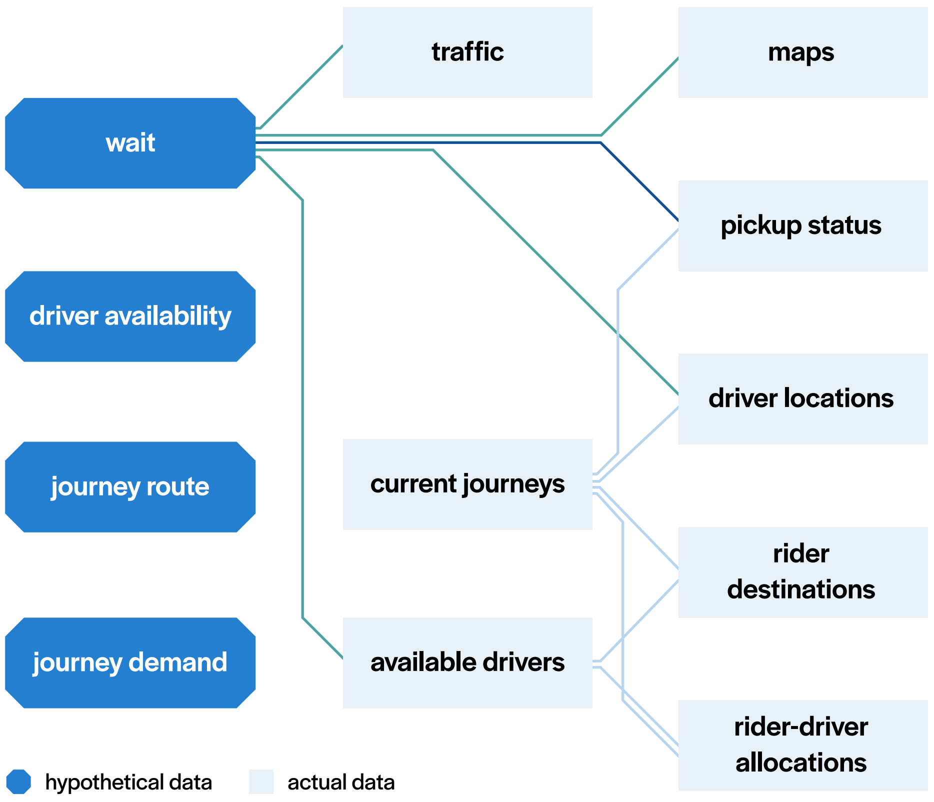
- driver availability, estimating time taken for drivers to arrive at the customer's location using the location of both driver and customer;
- cost estimate, using current location, destination, and availability of drivers; and
- driver allocation, allocating a driver that minimises their transport cost to the destination.

[29] For further information, see: https://safeboda.com/ug/index.php#whysafeboda

These decisions are made using a range of different data types, some of which are known – information about weather or traffic, for example – and some of which are hypothetical (Figure 1). A hypothetical stream is a desired stream of information which cannot be directly accessed. The lack of direct access may be because it relates to events that happen in the future, or there may be some latency between the event and the availability of the data. When a user calls for a ride, for example, they are provided with an estimate of driver availability that is made without knowing the intended destination, or without taking into account the real-world constraints on driver availability that can arise from regional boundaries, drivers reaching the end of their shift, or the intentions of passengers currently using their services.

Figure 1

Software components in a ride allocation system



Data-oriented architectures offer a structure for building these hypothetical streams and deploying them in decision-making in the system. Data-oriented programming allows the system to produce an estimate of the driver allocation (and rough cost estimate) before the user has confirmed the ride – allowing the system to confirm the constraints at hand – by declaring hypothetical data streams that approximate the true driver allocation, but with restricted input information and constraints on the computational latency.

With each system component – whether dealing with hypothetical predictions or real-world data – generating a data stream, the challenge for developers is to understand how these complex data flows move through the system. They will need the ability to investigate how decisions are being made and what influences a decision, for example to check suspicious patterns of activity, or to ensure that the system is not mis-using data from anomalous events (football games generating high demand, for example).

AutoAI's response to this challenge is to build a streaming algebra that describes which data streams are connected, disentangling the flows of data to build a map of the system that can be compiled and implemented in a streaming framework. This data-oriented architecture allows developers to analyse the interconnections between system components, to check how data is being used by the system, and to investigate – if needed – whether data use is compliant with regulatory requirements (for example, around the use of personal data). This map can also be used as the basis for deploying automated monitoring tools to check the allocation system is operating within desired parameters – for example, using classical software verification or statistical techniques to detect outlier events, or to ensure that decisions about individual services are not made on the basis of prohibited characteristics.

Designing such a system requires a combination of domain expertise relating to the operation of Boda Boda services, machine learning expertise, and software engineering and system design expertise. Once the system is deployed, there will be further challenges around maintenance, interactions with users, and redeployment as the environment changes.

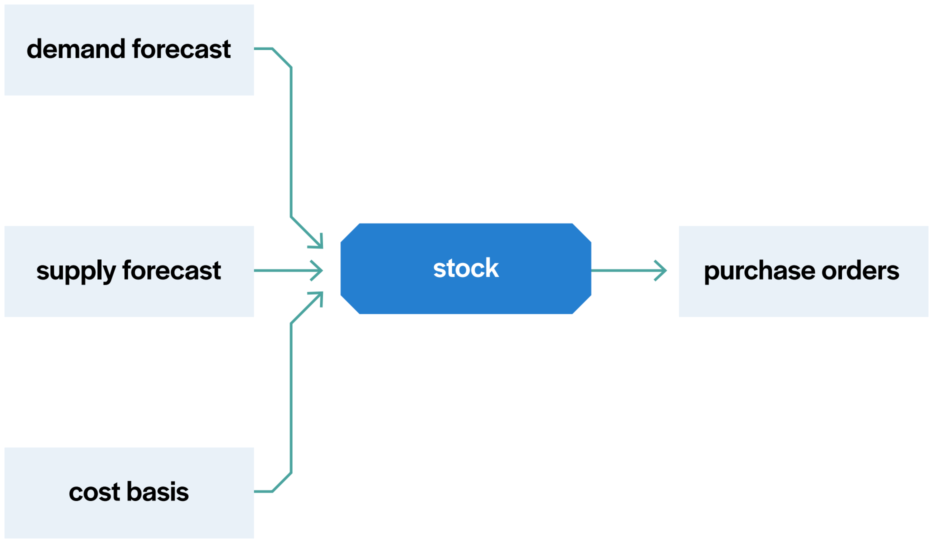
Supply chain management

Automated buying systems – often deployed in supply chain management – are complex decision-making systems. In such systems, the objective is to match demand for products to supply of products, often while minimising customer waiting times and costs associated with storing unpurchased goods. Such systems are built on core components that include:

- predictions of demand for the relevant product;
- predictions of supply of the product; and
- decisions about how much product to keep in storage.

Figure 2

The components of a putative automated buying system



Business decisions about whether to make new orders are made on the basis of the supply and demand (Figure 2).

The classical approach to building these systems was to create a monolithic system. Constructed in a similar way to software applications such as Excel or Word, monolithic systems rely on a single code base, which might become highly complex. Such systems might make use of shared, dynamically linked libraries for user interfaces or networking, but such software still often has many millions of lines of code.

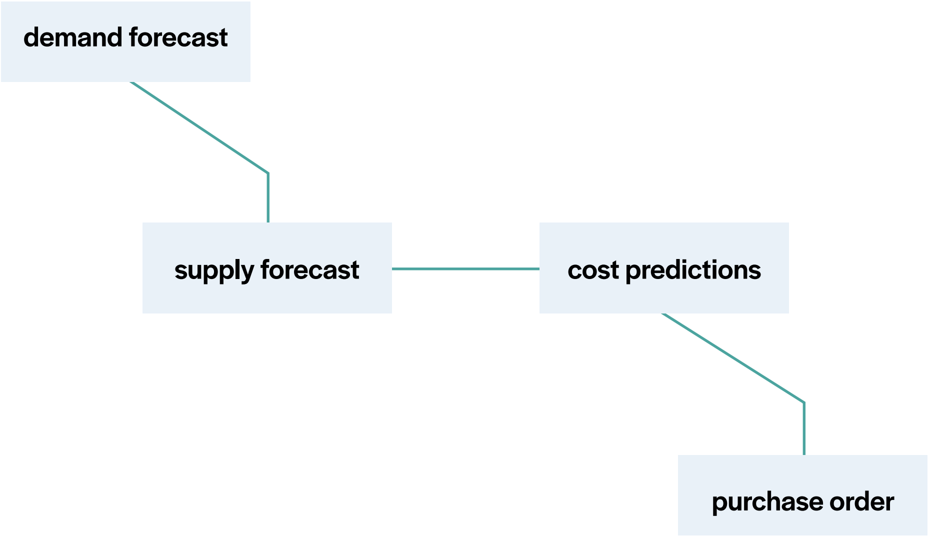
Monolithic systems are both difficult to develop and to scale when computation demands increase. A service oriented architecture offers an alternative path to integrating predictions about supply and demand. Rather than a single code base, the code for such systems is developed in different interlinked components, with individual services handling different requests (Figure 3).

In practice, each of these services is often 'owned' and maintained by an individual team. The team is judged by the quality of their service provision. They work to detailed specifications on what their service should output, what its availability should be and other objectives like speed of response. This allows for conditional independence between teams and for faster development. This approach is today common practice for software development.

The challenge associated with such systems is their relative lack of interpretability – an intellectual debt – owing to the complex web of interactions between services. AutoAI's data-oriented architectures tackle this by modelling the whole system as a data flow graph, which enables tracking the movement of data between components (see case study on rider allocation services). Data-oriented architectures also enable new approaches to end-to-end learning of supply chains, allowing developers to generate more accurate predictions of supply and demand.

Figure 3

A potential path of models in a machine learning system



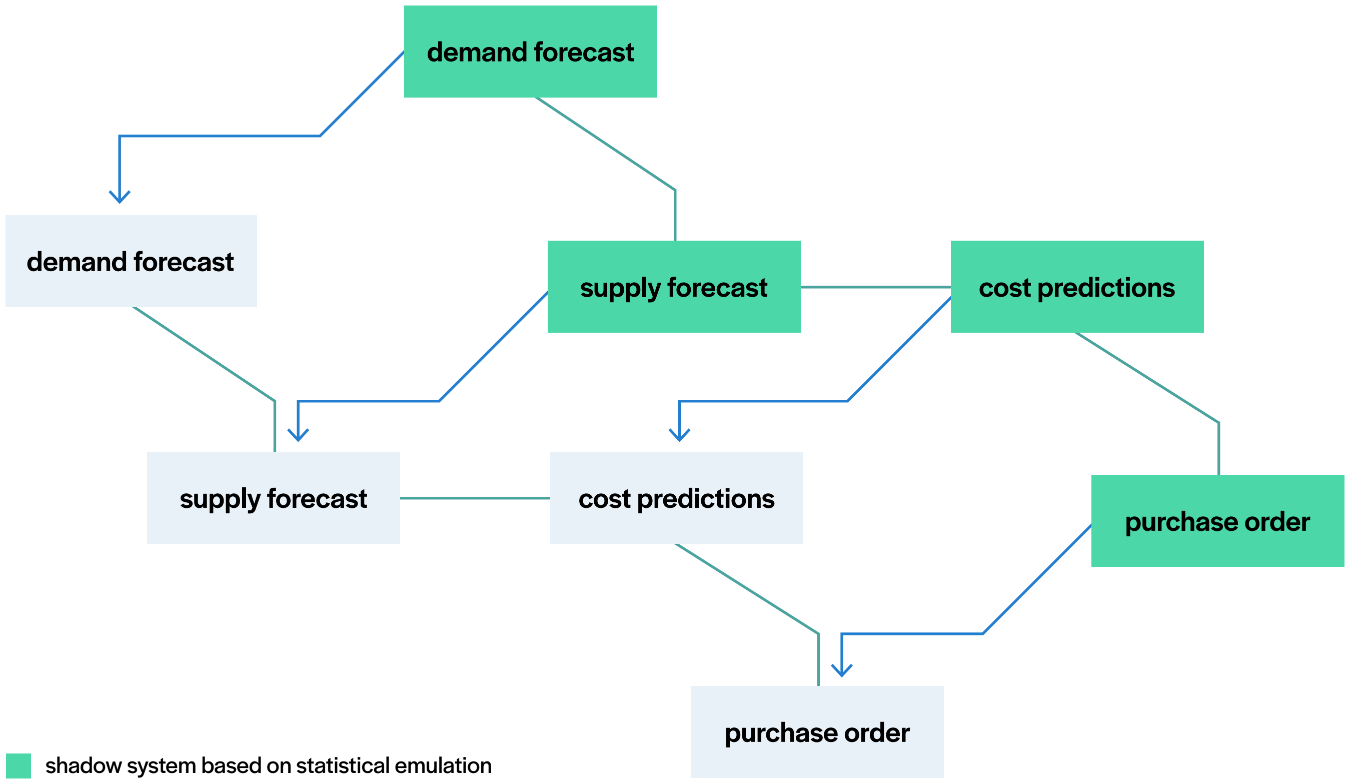
In many real-world supply chain systems, decisions are made through simulating the environment. Simulations can produce accurate representations of real-world systems, informing decision-making about the volume of different products to purchase and store, but are often computationally-intensive and time-consuming to run.

An alternative approach is to deploy statistical emulators alongside these sophisticated simulations. A statistical emulator is a data-driven model that learns about the underlying simulation, reconstructing the simulation with a statistical model. Importantly, emulators can learn with uncertainty, so it 'knows what it doesn't know'. As well as reconstructing an individual simulator, the emulator can calibrate the simulation to the real world, by monitoring differences between the simulator and real data.

In this way, networks of emulators designed to sit alongside the machine learning models in a supply chain system can provide a 'shadow system' – or hypervisor – that monitors system performance. These emulators can be used by business leaders to interrogate why a particular purchasing decision has been made and by analysts wishing to understand the implications of different purchasing patterns. They can also be deployed to facilitate end-to-end learning of the system, generating insights into the optimal operating conditions for the supply chain.

Figure 4

A potential path of models
in a machine learning system



Careful design of emulators to answer a given question leads to efficient diagnostics and understanding of the system, but in a complex interacting system an exponentially increasing number of questions can be asked. This calls for a system of automated construction of emulators which selects the right structure and redeploys the emulator as necessary. Automatically deploying these families of emulators for full system understanding requires advances in engineering infrastructure, emulation and Bayesian optimisation.

This ability to scrutinise the fairness of AI systems becomes increasingly important in domains where decisions might have significant personal or social implications. The same model considered above for buying, for example, can also be considered in the case of a banking application. In a typical banking application, a bank receives loan requests from customers. For an individual customer, before making a loan, the bank may wish to make a forecast around their costs (expenditures on food, housing, entertainment, etc) and their income (salary, rental income, etc). These forecasts would inform the conditions of the loan; how much the bank is willing to lend, and under what interest rates and repayment conditions. These terms will be based on previous experience of loaning, but also constrained by regulatory conditions often imposed by a financial regulator.

In many regulatory environments, the bank will be restricted in terms of what information they are allowed to use in dictating loan terms. For example, in the EU prohibited characteristics such as race, gender, sexuality, religion and health status cannot be used (even indirectly) for making the loan. Along with stipulating these characteristics, the GDPR also gives particular stipulations for rights individuals have to receive an explanation around consequential decisions, such as obtaining a loan.

Both banking and supply chain exhibit the challenges of a composition of decisions where upstream decisions may effect downstream outcomes. The context of these decisions matters because different regulatory environments specify different standards. However, the need for explainable maintainable AI systems exists across the domains.

Climate Ensembling Project

General Circulation Models (GCMs) of Earth's climate provide robust simulations of large-scale average climatic variables, such as end-of-century global average temperature, under various future greenhouse gas emissions scenarios. Each GCM is an imperfect representation which has different strengths and weaknesses when simulating historically observed climate. Averaging the simulations of a set of GCMs in a multi-model ensemble produces predictions that represent a 'best estimate', given the possible range of approximate representations of the climate system. These ensemble predictions provide critical information for those formulating climate mitigation policies.

Adaptation to the most severe impacts of climate change urgently requires reliable information about near-term, local-scale changes to environmental conditions that can be used to inform decision-making in government and business. This in turn requires specific information about local conditions, rather than global averages. In response, a wide range of spatial downscaling, bias correction and other statistical post-processing methods have been developed to derive such information from GCM simulations. Prediction in the context of adaptive decision-making requires a tailored, task-specific approach to the choice of GCM post-processing steps, and selection of method to combine the outputs of multiple GCMs in a multi-model ensemble.[30]

Optimal selection of GCM post-processing and ensemble methods for a specific climate risk prediction application requires multiple data-intensive experiments. Scientists usually configure such large-scale experiments manually. This repetitive manual process is error prone, reducing experimental efficiency and raising the risk of poor reproducibility. It is challenging to manually guarantee careful treatment of a large set of climate models that produce large amounts of data.

[30] For a description of work by Mala Virdee on this topic, see: https://ai4er-cdt.esc.cam.ac.uk/StaffDirectory/students-all/students. For further information about this work, see: https://mlatcl.github.io/projects/climate-ensembling.html

The repetitive and parameterisable nature of this kind of experiment should favour their automation. Data-oriented architectures can be used in such automation as a medium to orchestrate the execution and evaluation of computational graphs that represent experiments in different scientific domains. The data-oriented architectures approach provides an open environment where scientists can easily access the data at different experimentation stages, which is useful for both the finding of cause-effect relations between experiment variables, and the automatic reproducibility and verification of large-scale experiments.

Work by the AutoAI team is demonstrating how data-oriented architecture principles can be put into practice in this context, providing a set of software libraries that can orchestrate evaluation of computational graphs in a data-oriented way, using software containerisation and cloud computing stacks (such as AWS). The project involves working with outputs from a large set of climate models that are known to produce large amounts of data, which means careful treatment of data streams and re-use of completed computations are critical for fast experimentation and reproducibility.[31]

[31] More details on the data and models used for the Climate Ensembling project can be found at: www.wcrp-climate.org/wgcm-cmip/wgcm-cmip5

5

An AutoAI research agenda

Using these ‘3 D’s of system design’ as lenses through which to interrogate the functions required of AI systems in deployment, our research agenda for the creation of AutoAI emerges.

This agenda requires:

- advances in machine learning, and ways of reasoning about models;
- new systems architectures and programming techniques to solve technical challenges while operating at scale (‘solve to scale’);
- the integration of machine learning in wider systems and new approaches to systems design;
- the creation of data-oriented architectures; and
- identification of organisational and policy levers to influence AI adoption.

By advancing research and practice across these areas, the ambition of the AutoAI programme is to scale our ability to deploy safe and reliable AI solutions. In so doing, this work will:

- enable software systems for data-oriented architectures;
- create intelligent systems for monitoring and emulating the underlying complexity of machine learning systems;
- automate deployment and redeployment of machine learning systems; and
- identify technical and policy interventions to support ethical use of personal data.

The sections that follow sketch out the Programme’s areas of research interest.

Data-oriented architectures

New mechanisms to programmatically discover, monitor, and re-deploy machine learning models are necessary to tackle the issues of technical debt that are associated with increased complexity in AI systems. Data-oriented architectures are the foundation for such mechanisms. For example, if each machine learning model declares which data streams it is trained on and is monitored for deviation from expected performance standards, then it will be possible to create automated systems that identify when model retraining is needed. The resulting ‘hypervisors’ would identify when dataset shifts might influence performance,[32] identify conflicts between model outputs, and raise flags when the system appears to violate pre-determined guardrails, such as metrics for fairness.[33] The ability to organise interactions between system components through a data-oriented architecture should further allow system users to make more sophisticated use of predictions arising from different models. These predictions could, for example, be used to monitor model calibration, consistency, accuracy and fairness.

Data-oriented architectures have the potential to mitigate the challenges that arise when deploying machine learning algorithms in real-world systems. Implementing these architectures will require streaming infrastructures that can track and programmatically discover the inputs and outputs from different components of a system, helping users identify when system conditions have changed. The AutoAI Programme is developing automated methods for the compilation of such a system. Its forthcoming work will:

- identify and lay-out the basic components of data-oriented architectures in streaming format;
- introduce hypothetical streams for automated model declaration; and
- integrate factor graphs with monitoring tasks.

There are some attempts to define and discuss data-oriented architectures in the literature.[34] However, there is still not a clear definition of the principles that should drive the design of data-oriented architecture-based systems. In addition, it is not yet clear to what extent current machine learning-enabled applications have applied data-oriented architecture concepts. The AutoAI Programme's current focus is on understanding basic principles of data-oriented architecture systems, formulating common vocabulary, and building a community of academics and practitioners interested in developing data-oriented architectures. A forthcoming survey paper will present data-oriented architecture principles, definitions, definitions and an analysis on how these principles have been adopted by current AI-based systems architectures.

[32] Candela, J.Q., Sugiyama, M., Schwaighofer, A. and Lawrence, N.D. (2009) editors. Dataset Shift in Machine Learning. MIT Press, Cambridge, MA, 2009. ISBN 0-262- 17005-1

[33] Recognising that there may be differences between machine learning notions of fairness and public conceptions, which may not be quantifiable. Bridging these ideas requires further dialogue about the implications of substantive and procedural notions of equality of opportunity, bringing insights from across disciplines and from engagement with affected communities. For further discussion on this point see: Lawrence, N.D. (2022) Reclaiming control, available at: http://inverseprobability.com/talks/notes/ai-reclaiming-control.html

[34] Joshi R. (2007) Data-oriented architecture: A loosely-coupled real-time SOA. Vorhemus C. and Schikuta E. (2017) A data-oriented architecture for loosely coupled real-time information systems. In Proceedings of the 19th International Conference on Information Integration and Web-based Applications & Services (iiWAS '17). Lawrence N. D. (2021) Modern data-oriented programming, available at: http://inverseprobability.com/talks/notes/modern-data-oriented-programming.html

Data readiness and provenance

Data quality is a fundamental driver of system quality. Improving the ‘data readiness’ of organisations and teams requires a framework for establishing a common understanding of the work required before a data stream can be used. Data maturity assessments offer a framework for quantifying and labelling the data readiness of any stream of data. Data oriented architectures can then provide a basis on which these data maturity frameworks can be built.

The AutoAI Programme will seek to automate the deployment of tools for data cleanliness and assessment. Its current areas of focus include:

- identifying the actions needed to help made organisations ‘data ready’;
- developing data maturity assessments to promote common understandings of data quality issues;
- designing tools to automate record matching and data deduplication; and
- automating tools for data validity assessment.

Work by the AutoAI team has already explored the applicability of data readiness assessments in the context of Covid-19 response, identifying them as a tool to support rapid deployment of data and development of data science collaborations across organisations.[35] Forthcoming work will further investigate the implementation of these principles in the context of data-oriented architectures.

Data governance, stewardship and ethics

Data can be governed through a collection of policies, ethical frameworks, technical standards, legal or regulatory instruments, professional norms and operational processes. Effective data stewardship requires practitioners to select the most effective governance tools to achieve their desired outcome, in line with the values and aspirations expressed by affected communities and the wider policy environment.

The AutoAI Programme is investigating the technical and institutional data governance interventions that can support safe and reliable use of AI. In its efforts to develop and support equitable data sharing through technical innovation, it is:

- investigating what failure modes emerge from the deployment of machine learning, and what points of intervention can be leveraged to correct these failures;

[35] These grades are explored further in: The DELVE Initiative (2020), Data Readiness: Lessons from an Emergency. DELVE Report No. 7. Published 24 November 2020. Available from https://rs-delve.github.io/reports/2020/11/24/data-readiness-lessons-from-an-emergency.html

- progressing discussions around personal data sharing, with a particular focus on the development of data trusts;
- creating AI methods that are responsive to the evolving regulatory and policy environment; and
- investigating the role that AutoAI can play in helping ensure the FITness of an AI system and support compliance with regulatory requirements.

Data governance work by the AutoAI team has already gained international recognition. The launch of the Data Trust Initiative, funded by the Patrick J. McGovern Foundation, has resulted in seven research collaborations, three pilot data trusts projects, and a major work strand in the Global Partnership for AI.[36] These projects have supported the production of a new framework for operationalising data trusts,[37] increased understandings of current practice,[38] and developed a community of data trusts practitioners working together to advance the implementation of this novel approach to data stewardship.

One of the key goals of AutoAI is to establish an approach for building Fair, Interpretable and Transparent (FIT) machine learning systems. Building on an analysis of the Information Commissioner's Office AI toolkit, the AutoAI team will seek to bridge the gap between the practice of the software engineer (via our work on the data-oriented architecture) and the requirements of regulatory frameworks, identifying which aspects of such regulatory toolkits might be programmatically realisable. Research relating to FIT systems will move forward under the Climate Ensembling project, recognising that in the climate change context unfair outcomes may be associated with different geographic locations having data with different fidelities of sensing due to predominant presence of weather monitoring facilities in the global north.

End-to-end system optimisation

Achieving the end-to-end system optimisation that is required to automate deployment and redeployment of system components requires new ways of managing connections between machine learing models.

Data-oriented architectures would allow automated analysis of these connections, while hypervisors – emulators that monitor model performance – allow interrogation of their performance. Combining emulators in an end-to-end learning system requires new techniques in stochastic process composition and deep probabilistic modelling, to create a form of 'deep emulation'. Achieving this goal requires further work to understand the most suitable algorithms to use in different emulators

[36] For further information, see: www.datatrusts.uk and https://oecd.ai/en/wonk/data-sharing-data-trusts

[37] Montgomery, J. (2022) Creating a pathway to successful real-world data trusts, available at: https://datatrusts.uk/blogs/creating-a-pathway-to-successful-real-world-data-trusts

[38] For further information, see: Montgomery, J., Lawrence, N.D., Oh, S. (2022) Creating real-world data trusts: progress so far and the path ahead, available at: https://oecd.ai/en/wonk/creating-real-world-data-trusts-progress

(for example, for different sizes of data set). It will also require learning strategies to determine which emulators are suitable for which types of question, through structural learning of emulators, exploiting the factor graph structure produced by the data-oriented architecture.

The AutoAI Programme will deploy deep emulation to create machine learning hypervisors for data-oriented architecture. It will:

- explore the development of new methods for Bayesian machine learning, for example investigating what connections can be made between machine learning models and hierarchical systems;
- investigate how requirements form machine learning models and algorithms change when they are embedded in a wider system, and what metrics are needed to characterise whether a model is correct or useful;
- develop methods to represent uncertainties in hierarchical and multi-component systems;
- evaluate explicit and implicit variational approximation techniques for deep structural learning;
- develop techniques for counter-factual emulation, exploring what features of a system should be emulated, how, and the interactions between emulation and end-to-end learning; and
- connect emulation and Bayesian Systems Optimisation for monitoring of whole-system performance.

System design and continual system meta-learning

AutoAI proposes to deploy hypervisors to monitor performance in deployment, assessing system compliance with standards for accuracy, bias, fairness and consistency. At its most basic level, a hypervisor could be an anomaly detector for detecting, for example, data set shift.[39] When performance degrades, the hypervisor initiates a system response to redeploy the degrading model. More complex hypervisors would consider the interconnected system of emulators.

To support this system of hypervisors, and building on the Emukit software system,[40] AutoAI will create software tools that can operate in a continual learning environment. This will allow Bayesian methods created for end-to-end system optimisation to be used in support of automated deployment of machine learning models. Downstream effects of redeployment will be measured by deep emulation, using transfer learning to bootstrap previous emulations to rapidly deploy new emulators.

[39] Candela, J.Q., Sugiyama, M., Schwaighofer, A. and Lawrence, N.D. (2009) editors. Dataset Shift in Machine Learning. MIT Press, Cambridge, MA, 2009. ISBN 0-262- 17005-1

[40] Which provides a platform for Bayesian Optimisation, Multi-Fidelity Emulation, sensitivity analysis, Bayesian quadrature and experimental design

Future work will continue to develop methods for statistical emulation, such as using emulators for decision making and uncertainty quantification of real-world systems. The AutoAI Programme will automate redeployment of machine learning components through deep emulation. In support of this goal, the Programme:

- has compared current methods for analysing workflows and machine learning pipelines, and identified strengths of different streaming architectures;
- has defined principles of building data processing software systems that prioritise data and enable efficient deployment of machine learning; and
- is developing statistical monitoring systems for deployed machine learning models that identify when dataset shift has occurred, and retrain accordingly.

Future work will make progress in:

- creating variational online learning techniques for avoiding catastrophic forgetting;
- identifying what information about a machine learning module can and should be sent to other modules in a system;
- assessing optimal strategies to model, formalise and integrate pre-defined knowledge into learning strategies, and consider to what extent accumulated knowledge of systems can be used to update their predefined knowledge in dynamic environments;
- designing ‘shadow systems’ that can be used to explore counterfactual explanations about how a system works and test different deployment strategies;
- investigating how AI-enabled monitoring can help users understand performance in deployment, the reasons for any performance changes, and optimise system functionality (for example, optimising resource usage through targeted training on less data); and
- integrating concepts from control theory into machine learning to develop novel approaches to maintaining optimal system performance.

Information dynamics, infrastructure, and programming at scale

A challenge for AI systems design is that the predictions from different machine learning components are made on different time frames, but often about the same quantities. For example, prediction of demand may be made on a twelve month (annual), 3 month (quarterly), monthly, weekly and daily depending on whether the prediction is needed for income forecasts, labour planning, excess inventory removal or purchasing. In any complex decision-making system, the interlinking of these decisions leads to a form of information dynamics.

Building on the infrastructural work around deep emulation that helps to characterise the information dynamics of a system, AutoAI will investigate ways of dampening these characteristics. The AutoAI Programme will propose interventional solutions for identifying and fixing informational dynamic instabilities, using deep emulation to stimulate and resolve feedback loops in the context of data-oriented architectures.

A connected infrastructure challenge is managing compute demands for large scale AI systems. Computation is at the core of every algorithm and is allocated in an ad-hoc manner by engineers and researchers with expertise in the relevant problems and algorithms. However, automatically allocating compute in a data-driven way would increase efficiency and robustness. For example, control algorithms that are used in industrial settings often have a manually set computational budget meant to trade-off the speed of the action produced and the quality of the action chosen.[41] Manually setting this trade-off leads to suboptimal decisions and suffers from non-stationary behaviour.

The AutoAI Programme is studying and formalising resource allocation in bounded time for machine learning. The objective is to build on the structured formalism of data-oriented architectures to enable automatic compute allocation for part of the machine learning process which could be a core component of any AutoAI system. A first illustrative piece of work shows how the lack of consideration of computational costs in robotics can lead to issues during deployment. In recent years, the paradigm of deep reinforcement learning has made significant progress in addressing many decision-making problems from robotics to games. However, the intensive computational costs associated with the algorithms have gained less attention. This creates issues in the deployment of those algorithms in systems that require time sensitive inference.[42] Studying how modern algorithms perform in real-time settings underlines the importance of looking at the computational costs of the model to properly evaluate the performance of a controller.[43] This work from the AutoAI group was first presented at the NeurIPS Preregistration workshop. By first publishing study designs, researchers have opportunities to evaluate the performance of their systems more effectively. This implies a new approach to publishing machine learning studies.[44]

[41] Neunert, M., Farshidian, F. and Buchli, J. (2014) Adaptive real-time nonlinear model predictive motion control. In *IROS 2014 Workshop on Machine Learning in Planning and Control of Robot Motion*

[42] Huyen C. (2020) Machine learning is going real-time, available at: https://huyenchip.com/2020/12/27/real-time-machine-learning.html

[43] Thodoroff P., Li W., and Lawrence N.D. (2021) Benchmarking Real-Time Reinforcement Learning. Preregistration Workshop, NeurIPS 2021, available at: https://preregister.science/papers_21neurips/26_paper.pdf

[44] One example of such an approach comes from projects such as: https://preregister.science

While emulators can be computationally efficient, as is shown by works on Bayesian optimisation[45] and probabilistic numerics,[46] their predictions come with additional uncertainty. This leads to trade-offs between an uncertain but fast prediction against a slower but more computationally exact decision. Allocation of the computational budget is a core decision for AutoAI, and an issue being explored further through the Climate Ensembling project (introduced earlier).

Future work will define a common framework to assign and estimate the value of computations in machine learning. The first goal is to illustrate how many sub-fields of machine learning use similar ideas to reason about computations and develop a common language to discuss those decisions depending on the information available (making such a framework practical depends on underlying software frameworks such as data-oriented architectures). Practically, this will be done by using established decision-making frameworks and applying them in our context. The second goal is to develop new algorithms that enable dynamic computation allocation based on new information. In the context of AutoAI, those algorithms could enable more efficient use of computations for applications like hyper-parameter search or climate simulations.

Building an AutoAI community

Tackling the research directions outlined in this agenda requires a community of specialists from across technical domains, domain experts in different application areas, and practitioners with experience deploying machine learning systems on the ground. To help foster this community, the AutoAI team is creating bridges between different technical and practitioner domains, through:

→ **Teaching:** an open-access lecture course Machine Learning and the Physical World introduces core technical concepts that can be used to create machine learning systems that are suitable for real-world challenges. By studying how to create machine learning models with a principled treatment of uncertainty, the course helps participants understand how to leverage prior knowledge of a system to create machine learning tools whose decisions can be interrogated by their users.[47]

→ **Convening:** an important aspect of the project is building a wider research community around AutoAI challenges. The AutoAI AI Forum brings together businesses, researchers and practitioners interested in the deployment of AI in real-world contexts and

[45] Shahriari, B., Swersky, K., Wang, Z., Adams, R.P. and De Freitas, N. (2015) Taking the human out of the loop: A review of Bayesian optimization. Proceedings of the IEEE, 104(1), pp.148–175

[46] Oates, C.J. and Sullivan, T.J. (2019) A modern retrospective on probabilistic numerics. Statistics and Computing, 29(6), pp.1335–1351

[47] For further information, see: https://mlatcl.github.io/mlphysical

the technical approaches that can ensure such deployment is safe and effective. Starting with members of the AutoAI team and key contacts in the start-up community, the ambition is that this Forum will provide a hub for knowledge exchange, development of best practices, and ideas that drive further innovation in machine learning technologies and their use. Workshops at major conference venues such as ICML are also broadening the team's engagement with the research community, and generating wider interest in the AutoAI research agenda.[48]

→ **Partnership-building:** collaborations that bridge academic research and deployment are central to the AutoAI approach, and the Programme currently supports fifteen collaborations across seven partners in five application areas. Building on a long-standing collaboration between Professor Lawrence and Data Science Africa, the AutoAI team are also supporting a new Data Science Africa exchange in which researchers from across the continent develop collaborations and projects using AutoAI approaches to tackle local needs.[49]

[48] See, for example: https://icml.cc/Conferences/2021/ScheduleMultitrack?event=8368

[49] The first Data Science Africa Fellow, Morine Amutorine, has been investigating what lessons for policy and practice can be drawn from attempts to deploy data-enabled systems in support of Covid-19 crisis response. For further information about her work, see: https://mamutorine.github.io/about

6

Developing the research agenda

Designing AI systems that can be deployed safely and effectively requires a community of AI specialists, domain experts, practitioners, and affected stakeholders. AutoAI provides an umbrella to help build this community.

By bringing together different technical disciplines and by embedding research in deployed environments, AutoAI offers a route to:

- creating systematic understandings of deployment issues, building on current understandings the errors that arise in deployment and developing a way of reasoning about these errors from the perspective of system design;
- delivering a step-change in the technical capabilities of AI systems, through new ways of reasoning about machine learning models, and techniques and tools that support the operation of AI at scale;
- linking different technical communities, integrating insights from systems, control theory and policy to define and implement the characteristics of trustworthy AI system; and
- creating AI tools that work more effectively in practice, through accessible systems that make decision-making processes easier.

In support of these goals, the AutoAI Programme will work alongside partner organisations, integrating research with real-world use cases. Through collaborations with Data Science Africa and the AutoAI AI Forum, the Programme will identify opportunities to trial AutoAI methods and create AI systems that operate safely and reliably in deployment. Across its research areas and collaborations, AutoAI will embed a culture of open data science, seeking to widely distribute research ideas and to make the resulting AI tools openly available.

Today's challenges in the deployment of AI systems are fundamentally about solving at scale, through sophisticated decision-making systems that can operate in changeable, real-world conditions. This research agenda sets out the role AutoAI can play in meeting these challenges and is our first step towards building a wider ecosystem of knowledge and practice.

This document was produced in 2022 by the AutoAI team: Christian Cabrera Jojoa; Carl Henrik Ek; Markus Kaiser; Neil Lawrence; Eric Meissner; Jess Montgomery; Andrei Paleyes; Pierre Thodoroff; and Mala Virdee.

The AutoAI Programme is funded by EPSRC and The Alan Turing Institute, as part of its Senior Turing AI Fellowship scheme. It benefits from its engagement with Data Science Africa (DSA), and in particular the work of its first DSA Fellow, Morine Amutorine.

The Data Trusts Initiative collaboration between the University of Cambridge and University of Birmingham is further supported by Paula Bibby and Sylvie Delacroix, and funded by the Patrick J McGovern Foundation.

UNIVERSITY OF CAMBRIDGE

In partnership with

Data Trusts Initiative

The Alan Turing Institute

Office for Artificial Intelligence