The Roles and Modes of Human Interactions with Automated Machine Learning Systems

A Critical Review and Perspectives

THANH TUNG KHUAT, Complex Adaptive Systems Lab, University of Technology Sydney, Australia
DAVID JACOB KEDZIORA, Complex Adaptive Systems Lab, University of Technology Sydney, Australia
BOGDAN GABRYS, Complex Adaptive Systems Lab, University of Technology Sydney, Australia

As automated machine learning (AutoML) systems continue to progress in both sophistication and performance, it becomes important to understand the 'how' and 'why' of human-computer interaction (HCI) within these frameworks, both current and expected. Such a discussion is necessary for optimal system design, leveraging advanced data-processing capabilities to support decision-making involving humans, but it is also key to identifying the opportunities and risks presented by ever-increasing levels of machine autonomy. Within this context, we focus on the following questions: (i) How does HCI currently look like for state-of-the-art AutoML algorithms, especially during the stages of development, deployment, and maintenance? (ii) Do the expectations of HCI within AutoML frameworks vary for different types of users and stakeholders? (iii) How can HCI be managed so that AutoML solutions acquire human trust and broad acceptance? (iv) As AutoML systems become more autonomous and capable of learning from complex open-ended environments, will the fundamental nature of HCI evolve? To consider these questions, we project existing literature in HCI into the space of AutoML; this connection has, to date, largely been unexplored. In so doing, we review topics including user-interface design, human-bias mitigation, and trust in artificial intelligence (AI). Additionally, to rigorously gauge the future of HCI, we contemplate how AutoML may manifest in effectively open-ended environments. This discussion necessarily reviews projected developmental pathways for AutoML, such as the incorporation of reasoning, although the focus remains on how and why HCI may occur in such a framework rather than on any implementational details. Ultimately, this review serves to identify key research directions aimed at better facilitating the roles and modes of human interactions with both current and future AutoML systems.

Additional Key Words and Phrases: Human-Computer interaction, Automated machine learning (AutoML), Autonomous machine learning (AutonoML), Roles and modes of human interactions, Machine learning workflow, Fairness, Explainability, User interfaces, User experiences, Stakeholders, Reasoning, Closed-world AutonoML systems, Open-world AutonoML systems, Data-driven machine learning, Knowledge-driven machine learning, Industry 4.0, Industry 5.0

1 INTRODUCTION

Broad interest in machine learning (ML) has ebbed and flowed ever since the 1950s, but recent years have arguably witnessed a new phase in the history of the field: an unprecedented level of technological uptake and engagement by the mainstream. From deepfakes for memes to recommendation systems for commerce, ML has become a regular fixture in broader society. Unsurprisingly though, this ongoing transition from purely academic confines is not smooth; the general public does not have the extensive expertise in data science required to fully exploit the capabilities of ML.

The ideal solution for democratisation is to make the application of ML optionally independent of human involvement. This is the primary goal of automated/autonomous machine learning (AutoML/AutonoML), an endeavour that, despite a rich multi-faceted history [148], has only truly

Authors' addresses: Thanh Tung Khuat, Complex Adaptive Systems Lab, University of Technology Sydney, Sydney, New South Wales, 2007, Australia, thanhtung.khuat@uts.edu.au; David Jacob Kedziora, Complex Adaptive Systems Lab, University of Technology Sydney, Sydney, New South Wales, 2007, Australia, david.kedziora@uts.edu.au; Bogdan Gabrys, Complex Adaptive Systems Lab, University of Technology Sydney, Sydney, New South Wales, 2007, Australia, bogdan.gabrys@uts.edu.au.

taken off within the last decade. Popularised by significant optimisation advances applied narrowly to model selection [251, 284, 291], the scope of AutoML has since expanded to automating all aspects of an ML application. Indeed, as long as there is a will and a way, it seems inevitable that ML systems will move ever closer towards autonomy.

As of the early 2020s, much has been written specifically around mechanisms and integrated systems for automating operations in both general ML [148] and the fashionable subclass known as deep learning (DL) [70]; the topic of mechanising the latter is abbreviated as AutoDL. These discussions have mostly taken the notion of 'automation' to heart, wrestling with the challenges of how computers can make high-level decisions on their own. However, one important topic has been left underexplored: how do humans fit into the picture? This is crucial to consider, as, no matter how far its capacity for autonomous function evolves, the purpose of an AutoML system is to support human decision-making. Thus, perhaps counterintuitively, interactions cannot be an afterthought [6].

Even with an academic focus on model accuracy and algorithmic efficiency, systems cannot be considered optimal if they do not welcome and make use of optional human input. Moreover, beyond academia, the concept of 'performant' ML becomes much more complex and user-centred [258]; the most promising algorithms and architectures will likely be ones that flexibly tailor outputs to satisfy a very broad set of requirements. Then there is debate on just how much autonomy ML systems should be given. While the nature of the human-system relationship may eventually become one of collaboration [311], it is unlikely that humans will ever relinquish supervisory oversight [81]. Many researchers have echoed similar opinions, stating that human experience is indispensable and AI cannot be expected to autonomously operate in a socially responsible way [319]. For this multitude of reasons, a holistic appreciation of AutoML requires an associated study of human-computer interaction (HCI).

This is a rich topic; the nature of human interactions with AutoML, both in terms of roles and modes, has evolved and will continue to evolve alongside developments in the field. Consider, as an analogy to those developments, the history of artificial intelligence (AI) with respect to the game of chess. In the late 1960s, Mac Hack became the first chess program to play in human tournaments and even score a victory in doing so [109]. Automated but heavily reliant on domain knowledge – it incorporated approximately 50 expertise-based heuristics – and hardly a threat to human dominance in chess, Mac Hack can be likened to proto-AutoML model-recommendation systems that were developed prior to the 2010s [259, 304]: novel and impressive for the time, but severely limited. Eventually though, by 1997, the swell of computational resources and advances in algorithmic techniques enabled a chess-playing computer known as Deep Blue to defeat a reigning world champion [45]. As with the new wave of hyperparameter-optimising AutoML systems in the 2010s [251, 284, 291], themselves becoming more and more capable of scaling competition leaderboards [82], Deep Blue heralded an era where computers would be far more competent than humans at performing a specific task.

Notably, even during the famed contest of 1997, Deep Blue was far from autonomous, leveraging a human-prescribed database of openings and endgame meta-knowledge, while also being manually adapted by grandmasters between games. Only in 2017, with the initial release of AlphaZero [329], has human input almost completely been removed, with an AI system autonomously learning to dominate humans in chess via self-play. In fact, the newest generation of chess-based AI has started to shift the roles of humans from mentors to students, with, for instance, an AI propensity for 'h-pawn thrusts' giving high-level players pause for thought [198]. The field of AutoML has not yet reached the same level of autonomy[1], but it is nonetheless worth asking: is this the state of interactions to plan for in the future? Will AutoML eventually produce more insight on how to solve an ML task than it currently receives?

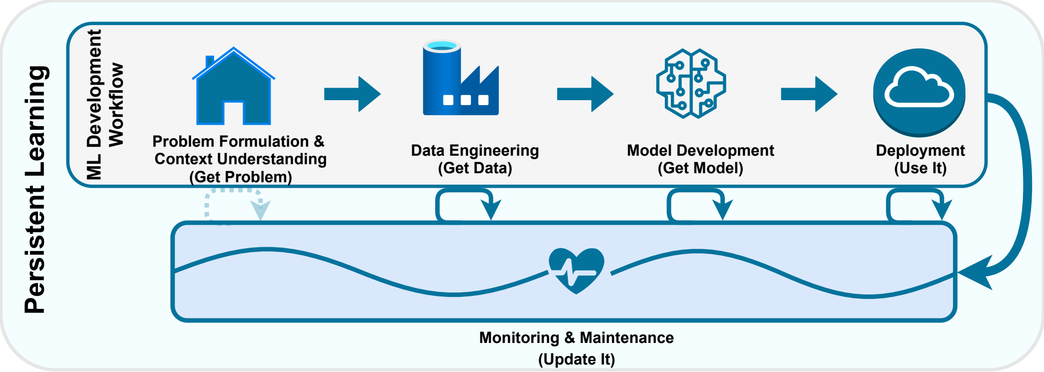

Persistent Learning
ML Development Workflow
Problem Formulation & Context Understanding (Get Problem)
Data Engineering (Get Data)
Model Development (Get Model)
Deployment (Use It)
Monitoring & Maintenance (Update It)

Fig. 1. General schematic for a machine learning (ML) workflow, i.e. the operations involved in producing and maintaining an ML model for an ML application.

A comprehensive overview of HCI in AutoML, both current and prospective, needs to be carefully structured. For instance, human involvement in ML applications can be partitioned into two categories: productive and consumptive. While the latter refers to how end-users engage with and benefit from an ML model, the former relates to how such a model comes about. These 'productive' practices can be codified in many ways into an 'ML workflow' [53, 280], but one particular representation [70, 148, 258] is schematised in Fig. 1.

Within the depiction of an ML workflow, it is clear that there are several phases of operations involved in developing, deploying and maintaining an ML model. Of these, the model development stage receives the most focus in academic AutoML research, especially in the case of DL and neural architecture search (NAS) [70], but there have been numerous automation efforts applied to the rest of an archetypal ML workflow. In fact, the capacity to continuously monitor performance and adapt to dynamic changes in data environments has previously been highlighted as a key prerequisite for the transition between AutoML and AutonoML [148]. Relatedly, there exist many theoretical proposals for supporting online learning [162], and initial experimental research towards making AutoML systems 'persistent' has recently been undertaken within academia [14, 51]. Meanwhile, in industry, the emerging trend of 'MLOps' reflects the importance of automated deployment to real-world demands [258]. In essence, any developers interested in engineering a comprehensive AutoML/AutonoML system must understand the idiosyncrasies of each and every workflow phase, e.g. the associated formats of human inputs/guidance, the minimal requirements for baseline operations, the possible opportunities for additional human-assisted learning, and so on.

Alternatively, rather than partitioning HCI in AutoML by when an interaction occurs, it is sometimes more informative to consider who is undertaking the interaction. This perspective is particularly natural in commerce and industry [258], where, beyond the end-users that consume the outputs of an ML model, there are typically numerous stakeholders linked to the production of a model. These may include technicians in the form of data scientists or software developers,

[1] The AlphaZero approach is shared by AutoML systems such as AlphaD3M [72], but, for the purpose of the chess analogy, the two are not equivalent; automation in chess is significantly more advanced than automation across an entire ML application.

business staff in the form of project managers or domain experts, regulatory groups in the form of third-party auditors or government agencies, etc.

Importantly, the obligations and interests of different stakeholders are usually not mappable to individual stages within the ML workflow depicted in Fig. 1. Moreover, the modes of their interactions may also vary substantially. Some roles will demand fine control over AutoML processes, while others will simply require an entry point for inputs. Some roles will desire a window into the mechanisms involved, while others will only want to be alerted if things go wrong. Whatever the case may be, these requirements must be considered at the fundamental level of algorithm and architecture. It is not optimal to focus solely on predictor accuracy and efficiency during system design, only to service any remaining real-world expectations with a hasty patch-up.

Crucially, it is worth emphasising that the concept of a 'user' is intrinsic to the stakeholder view of AutoML. This may seem like an unnecessary distraction to those that are solely interested in improving the statistical theories underlying ML algorithms, but that attitude ignores the greater ecosystem in which ML operates: human decision-making. For instance, humans may be willing to tolerate disliking 40% of AI-recommended music, whereas a 20% false-positive rate for AI-recommended convictions is arguably abysmal. Simply put, human context matters more than any agnostic accuracy metric. It follows then that successfully translating ML model performance into real-world outcomes is contingent on a set of engagement-related requirements [12, 73, 76, 256, 264], which we will bundle here under the title of 'user experience' (UX). This includes topics that have recently diffused into academic discussions around ML and AutoML, such as accessibility, transparency, fairness, reliability, and so on [319].

Accordingly, the concept of a user interface (UI) – the implementation need not be monolithic – becomes particularly important to AutoML within the stakeholder perspective, as this is where UX can most directly be managed. Indeed, designing intelligent UIs is critical for supporting human-guided AutoML [105, 168], where a technical user might, as an ideal, tweak problem settings, explore data characteristics, limit model search spaces, etc. These interactions may also be constrained or facilitated by the manner in which they occur, e.g. via touch-screen, voice commands, gesture recognition, or even brain signals [320]. In short, the field of AutoML would be well served by greater discussions around the concept of interfacing so that, beyond simply enabling the control of ML operations, users can both inject domain knowledge and extract comprehensible information with ease.

Turning to factors that influence UX, explainability is high up on the list. This is especially a challenge for AutoML, as the core principle of automation is to decouple humans from certain operations. It may seem wasteful then, if not counterproductive, to spend research effort in making those processes transparent, consequently encouraging humans to reinvolve themselves. Sure enough, many current AutoML tools are staunchly black-box systems [319], veiling how ML models are built and how predictive/prescriptive outputs are generated. But herein lies the nuance; the aim of AutoML is to remove the *necessity* for human engagement, not the option. Thus, technical obscuration actually impedes ML performance if users cannot understand how to properly insert domain knowledge that would otherwise be beneficial to an ML task [180]. This is especially a drawback at the current point in time, because human-in-the-loop learning is still often more advantageous than machine-centric ML [285].

Regardless, even if AutoML systems were to be completely autonomous, their innards untouched by humans, explainability is also necessary to promote trust [187]. Surveys indicate data scientists tend to be sceptical of ML models provided by AutoML tools if there are no mechanisms for transparency and understandability [73]. Similarly, end-users only follow ML recommendations if the system can show the reasoning behind them [302]. This reticence by people to use results they cannot understand or explain can be frustrating for simple business applications, but it is completely warranted in high-stakes contexts [244], including medical diagnosis, financial investment and criminal justice. To do otherwise could be disastrous. For example, adverse outcomes have been linked to the COMPAS recidivism prediction model [66], the BreezoMeter real-time air-quality prediction model used by Google during the California wildfires in 2018 [192], and black-box medical diagnosis models in general [74].

Another factor that affects UX, even and especially if stakeholders are not directly aware that they are 'using' the results of ML, is fairness. This socially conscious requirement has recently been taken up as an important issue by academic research [50, 193, 326], indicating just how far ML has embedded itself into the mainstream, and recognises that predictive/prescriptive accuracy and error may disproportionately affect different people in different ways. Now, granted, there have been efforts towards automating mechanisms for discovering and preventing discrimination in ML models [116], but the challenge is that there are many possible technical definitions for fairness, often orthogonal and sometimes contradictory [254, 307]. Once again, human context matters. So, it is an open question as to how human oversight can best be integrated within an AutoML system, enforcing ethical requirements upon a mechanised process.

Of course, while every ML algorithm applies its own assumptions, many 'unfair' biases are often sourced from biological neurons, i.e. human brains. These can be injected into learning systems via data and knowledge, manifesting in both information content and sampling. As a result, human cognitive biases that are internalised can lead to a deterioration in model reliability, and there are many high-profile examples of this occurring [131, 327]. The severity of these impacts can also vary depending on context. Healthcare is one example of a high-stakes environment, where cognitive biases in clinical practice can have a strong influence on medical outcomes [228, 253]. Indeed, similar flaws in predictive systems have been shown to hinder social minorities from receiving extra care services [214]. Accordingly, there is an imperative to more thoroughly consider the nature and implementation of bias mitigation strategies within AutoML.

Fundamentally, the point of all this discussion is that, given a suitable conceptual framework, such as the dual workflow/stakeholder perspective of ML operations, it is possible to engage with many HCI-related issues that, unaddressed, will impede the heretofore surging pace of progress in AutoML. Moreover, this kind of systematic approach does not just clarify the current state of HCI in AutoML; it provides a lens through which the future of this trend in ML can be forecasted. This does not mean speculating on detailed implementations of HCI-related mechanisms, but rather understanding the projected evolution of human-system interactions, particularly as algorithms and architectures become better at their job. Thus, the aforementioned chess analogy remains useful in illustrating this progression as AutoML systems shift further along the spectrum of autonomy [270].

However, it is still valuable to conjecture just a little bit further. AlphaGo [267] and AlphaZero [268] are extremely competent at their respective games, but they remain constrained within particular environments. An equivalent AutoML system would essentially be autonomous for every phase of the ML workflow in Fig. 1 except for one holdout: problem formulation and context understanding. Such a constraint is not unexpected, as this phase is likely to be the last bastion of necessary human involvement in ML. Unfortunately, it does stand in the way of numerous AI applications. For instance, there exists plenty of research and development in the field of autonomous vehicles [122, 167], yet the challenge of operating in unpredictable and effectively boundless driving environments remains, to date, daunting [119]. Nonetheless, without delving into the domain of artificial general intelligence, these constraints will eventually relax. The novel MuZero system [257] is already emblematic of an emerging reinforcement-learning approach that can be applied agnostically to a variety of games with diverse rules, autonomously building competent models from first principles. In theory, cognitive models may eventually supercharge this process even further, enabling ML systems to efficiently transfer knowledge from one problem to another by actually *understanding* context, rather than by outright ignoring it. So, as AutoML truly becomes AutonoML, and then begins to relax into open-world learning: will human-system interactions change yet again?

As is evident by now, there are many important issues to consider in the overlap between AutoML and HCI. This review addresses these topics, marking the final part in a series of monographs dedicated to a systematic and conceptual overview of AutoML [70, 148, 258]. Specifically, with the series having previously focussed on how computers may perform ML/DL in the absence of humans [70, 148], this work aims to recontextualise AutoML/AutonoML back within the ecosystem of human decision-making. In fact, because ML does not operate in a vacuum within the real world, some organically arising consequences of this interlinkage are already captured by the previous technological survey in the series [258]. This review, however, dives much deeper into the fundamentals of HCI in AutoML, driven by the following set of questions:

- How does HCI currently look like for state-of-the-art AutoML algorithms, especially during the stages of development, deployment, and maintenance?
- Do the expectations of HCI within AutoML frameworks vary for different types of users and stakeholders?
- How can HCI be managed so that AutoML solutions acquire human trust and broad acceptance?
- As AutoML systems become more autonomous and capable of learning from complex open-ended environments, will the fundamental nature of HCI evolve?

To best grapple with these questions, the rest of this monograph is structured as follows. Section 2 examines HCI and state-of-the-art AutoML as of the early 2020s. It does so with respect to the workflow/stakeholder perspectives of AutoML, after these are first systematised. Modern approaches to UIs and current concerns around UX, e.g. in terms of explainability and fairness, are also surveyed. Section 3 then extrapolates progress in AutoML to where associated systems are effectively autonomous in all high-level ML operations, excluding problem formulation and context understanding. The evolution of HCI with respect to genuine high-performance AutonoML, albeit restricted to constrained environments, is considered. Section 4 follows by pushing this limit, relaxing restrictions and considering ML within open-ended environments. To anchor such a scenario, modern theories for incorporating 'reasoning' in ML systems are surveyed and debated. Subsequently, changes to the nature of HCI with respect to these upgraded forms of AutonoML systems are theorised upon. A synthesising discussion is then presented in Section 5, identifying existing issues and potential research directions that may hinder or facilitate the successful interplay of HCI and AutoML, both now and in the future. Finally, Section 6 concludes this review, summarising key findings and perspectives around the roles and modes of human interactions with AutoML/AutonoML systems.

2 INTERACTING WITH AUTOML SYSTEMS: CURRENT PRACTICES

This section is dedicated to reviewing HCI in modern practicable AutoML. However, it is first worth distinguishing this review from the one that precedes it [258]. Consider the concept of 'AutoML ergonomics': the study of interactions between humans and other elements of an AutoML system, where such an understanding can be used to optimise human well-being and overall system performance. In the context of AutoML, human well-being includes comfort, safety and productivity while running an ML application. Under such a definition then, both reviews deal with AutoML ergonomics. That said, the focus of the prior monograph is the mainstream technological emergence of AutoML, examining how stakeholder needs have shaped both the implementation and application of AutoML systems/packages. Accordingly, its survey was entirely grounded in concrete products; ergonomic optimisations in HCI were identified as an organic *outcome* of the technological translation from academia to industry. In contrast, this monograph flips cause and effect. Specifically, leaning on broader scientific literature, this review partially examines how ergonomic principles for HCI act as a *driver* for designing/informing AutoML interactivity, both present and future. Of course, the two reviews overlap when discussing the current state of AutoML, but the complementary perspectives of HCI prove illuminating.

With this complementarity in mind, we begin by reintroducing the concepts of an ML stakeholder and ML workflow in Section 2.1 and Section 2.2, respectively. This enables discussions around *who* is interacting with modern AutoML and *what* phase of an ML application their interactions concern. We then start to examine the notion of interfacing with AutoML systems in Section 2.3, listing out essential functions that a UI should support, as well as the modalities in which interactions may be communicated. This section could almost be considered a discussion of 'physical' ergonomics, as it mostly concerns the comfortable and productive communication of human intent via physical means. Section 2.4 then highlights significant challenges facing HCI in modern AutoML, reviewing existing attempts to improve trust in ML solutions via efforts to support explainability and fairness. This section could be considered a discussion of 'cognitive' ergonomics, where improving the outcomes of interactions goes hand in hand with optimising the mental processes of a stakeholder, e.g. situational awareness and decision making. Finally, Section 2.5 synthesises these ergonomic considerations, examining recent proposals for a list of critical requirements that an optimal AutoML UI should satisfy. Ultimately, through these various discussions, we capture a comprehensive snapshot of how the roles and modes of human interactions with AutoML systems are currently regarded in the early 2020s.

2.1 Types of Stakeholders

A popular motivation and selling point for AutoML has always been the democratisation angle, in that associated systems can allow 'anyone' to run an ML application, including lone hobbyists and laypeople without data-science expertise. However, most automated learning systems are embedded deeply within human social environments, with the expectation that various groups of humans will interact and collaborate with these systems [76]. Thus, while most AutoML products still lean towards single-user interfaces, it is necessary to consider a broad range of stakeholders, including those whose interactions are relatively indirect. Indeed, the previous review spoke in terms of business organisations [258], where each entity within an organisation has its own role and requirements concerning a learning system. Moreover, the previous monograph categorised these stakeholders with an eye to the outcomes they seek, with the notion of 'performant ML' denoting an efficient achievement of these aims. Here, we similarly provide a classification of stakeholder types, but the focus is now on the stakeholder-system information flows that their roles require.

In condensed form, there are effectively four classes of stakeholders that can engage with an ML application, i.e. the focus of AutoML operations. These categories are the technical group, the business group, the regulatory group, and the end-user group. Naturally, their purview may involve various phases of an ML application, ranging from problem formulation and context understanding to monitoring and maintenance; see Section 2.2. Nonetheless, under such a classification scheme, their roles and consequent interactions with an AutoML system are relatively well defined, presented in Table 1. We note that this does not mean real-world job titles and responsibilities need to stay confined within a particular category. For instance, an ML expert may be a hybrid stakeholder with two parallel agendas: one, work in the technical group to guide AutoML when searching for optimal ML solutions, and two, work in the regulatory group by providing auditing feedback to a government agency about the reliability and fairness of an AutoML system. The convenient categorisation in

Table 1 simply systematises the interactions that an AutoML system may experience, along with the information flows required to satisfy their intent.

Before proceeding, though, we raise a nuanced point. Most discussion in this monograph does not consider the top-down construction of an AutoML system, the standard approach in academia/industry, to be an interaction *with* the system. This assertion remains true even though certain technicians, under advisement from other stakeholder groups, are responsible for enacting this construction. On the other hand, the *configuration* of the system very much falls within the scope of HCI in AutoML, even if an ML application is not presently being run. Consequently, one could claim that interactions with an AutoML system to configure a new default parameterisation are part of designing that system. Specifically, depending on whether the intent of this HCI is permanent reconfiguration or problem-specific customisation, the roles of specific stakeholders can blur between AutoML design and the processing of an ML application. To some extent, this is simply a matter of terminology; every AutoML system retains some form of a skeleton that is designed/constructed *prior* to any HCI. Even the most novel of hypothesised frameworks, such as organically emergent AutonoML/AutoDL architectures [70, 148], are anchored somewhere with hard-coded principles. Ultimately, given that the majority of stakeholders wish to *use* AutoML, not just set it up, we acknowledge the nuance but speak of HCI mainly in terms of running ML applications on an existing system.

Turning now to the categorisation of stakeholders, the first group listed in this section is **the technical group**, including data scientists, data analysts, ML engineers, software developers, and system engineers. Traditionally, this group is responsible for developing, deploying, monitoring and maintaining ML solutions, so, at least at present, they are likely to be closely involved anywhere an AutoML system attempts to automate any of these technical operations. This involvement is especially vital given that ML solutions can be very complicated, often structured as 'ML pipelines' of data transformations that we call 'ML components'. Indeed, no existing AutoML package is close to perfect at finding an optimal ML pipeline for every ML problem, so the technical group continues to have a hand in supporting, constraining or outright making many decisions during this process.

The functional roles of a technician can be numerous, with their inputs determining/influencing the use of data samples, feature constraints, model types, hyperparameter bounds, performance metrics, learning/adaptation strategies, computational resources, control/analysis mechanisms for fairness and explanation, and so on. However, the core focus of these efforts is ensuring an ML solution performs technically well, e.g. in terms of predictive accuracy and scalability. Thus, the technical group expects to have access to the ML solution object, if it is not directly returned by an AutoML system, as well as any outputs relevant to technical performance. Additionally, technicians may also be responsible for verifying that any of the mechanisms/features within an AutoML package are sound and fit-for-purpose, e.g. bias mitigation or high-performance computing, even if they are not necessarily the stakeholders that will engage with their outcomes. Finally, we stress that, while the last decade has seen significant advances in AutoML, few modern systems are genuinely fire-and-forget upon being given their inputs. AutoML can accelerate the search for good ML models, but technicians still lie at the heart of solution-seeking, which is why so many AutoML vendors continue to offer close back-end support, even for products that support user-driven ML. As such, the best modern AutoML systems are expected to provide feedback that can assist a technician with understanding solution space, e.g. via model comparisons or other representations of sampled hyperparameters, so that they can better refine these optimisations. Arguably, the value of these particular outputs may diminish somewhat as AutoML improves its automated capabilities; see Section 3.

Table 1. Overview of different stakeholder groups and their roles and interactions with AutoML systems

Stakeholder	Main roles of human interactions	Human's Input	AutoML's output
Technical group	• Collect and explore data • Use domain knowledge to guide AutoML algorithms find and select the best learning models • Compare different learning models • Verify accuracy and performance of AutoML algorithms • Assess the explainability, understandability, reliability, accessibility, and transparency of solutions • Bias and fairness checking for the obtained learning models • Evaluate the security, accessibility and scalability of resulting learning models in production environments • Deploy, monitor, and maintain learning models	• Training data • Selected features/samples and feature requirements • Rules for data pre-processing • Search spaces of models and hyperparameters are narrowed down • Resource constraints • Requirements for model assessment • Requirements for model/hyperparameter comparison • Criteria for types of explanation, fairness, and transparency of solutions • Technical and non-technical constraints • Learning and adaptation strategies • Settings for operation environments	• Data visualisation and statistics • Resulting learning pipelines • Performance and accuracy of the obtained learning models • Graphical controls/ analyses for AutoML processes • Details of the searched learning models and hyperparameters • Model/hyperparameter comparison views • Explanations of outcomes generated by learning models in different forms • Outputs of metrics related to fairness • Reports related to functional and non-functional constraints • Figures related to the operation of learning models in the real time for monitoring and maintenance • The availability of learning models in the production environment and continuous learning and adaptation abilities in real time
Business group	• Provide business requirements for the development and design of AutoML systems • Provide the acceptance criteria of AutoML systems in terms of functionalities and usefulness • Assess and verify the effectiveness of AutoML systems related to business aspects • Assess the maintainability, scalability, accessibility, and usability of AutoML systems	• Business requirements or constraints • Business success criteria • Non-functional constraints, e.g., maintainability, accessibility, scalability, usability, resources, and profits	• Reports for experimental results for business success criteria and business requirements • Results corresponding to each non-functional requirement
Regulatory group	• Validate and audit the functional requirements of the systems compared to specification and design documents • Validate and audit the non-functional requirements of AutoML systems such as security, safety, reliability, transparency, fairness, ethical issues, social norms, law compliance, and usefulness	• Test cases/Input data to evaluate and audit the functional and non-functional requirements for resulting learning systems	• Reports or experimental results for input test cases • Views to show reasons or processes to achieve the outcomes for test cases
End user	• Use and interact with the deployed learning models • Analyse and make decisions based on the recommendations provided by AutoML models • Control and supervise the operating procedures of the AutoML system • Planning and goal management	• Provide the real inputs for the AutoML system via user interfaces • Provide feedback to the AutoML system	• Outcomes for each input data • Show reasons or explanations for outputs • Continuous learning and adaptation to user behaviours and feedback

The second group listed in this section is **the business group**, including business analysts, domain experts, and project managers. While the technical group ensures that the mechanics of an ML application run smoothly, business stakeholders are the ones that embed the AutoML machinery within real-world contexts and problems. Thus, they mainly contribute to either end of an ML workflow, translating business requirements and sourcing raw data for application inputs, then verifying the suitability of solution outputs for use and deployment. After all, a quick and accurate ML model is not helpful if it solves the wrong problem or cannot be leveraged for consequent impact. At a higher level, some business stakeholders are also responsible for deciding whether an AutoML system is worth using in the first place, but, to date, such decisions are either made prior to interacting with the system or in response to running the product. In the latter case, assessing the value of an AutoML package is often equivalent to assessing the experiential/solution quality of one or more processed ML applications.

In terms of inputs, the business group will typically intend to provide real-world criteria and constraints into an AutoML system. Ideally, these stakeholders should also acquire business-relevant feedback on the performance of both the AutoML system and the ML solutions it produces/deploys. Such metrics will typically involve operational costs, e.g. electricity bills. Additionally, many businesses will be particularly interested in the final outcomes of an ML application, e.g. profit from user subscriptions to an ML-driven product, disease cases averted by an epidemiological model, etc. Such information is more challenging to provide as it requires AutoML services to embed and monitor an ML solution flexibly in a customisable deployment environment. This caveat leads to an important point; we stress intent and ideals when defining business inputs and outputs. At the current time, albeit with a growing number of exceptions [258], most AutoML packages do not provide an easy way to blend a real-world business context with the abstracted mechanics of ML. It is thus still a common expectation that technical stakeholders serve as a bridge for many interactions between the business group and an AutoML system, e.g. filtering data inputs by business constraints or interpreting scalability/maintainability requirements as financial costs. However, should the field of AutoML eventually discover/implement mechanisms for autonomous context understanding, the notion of a technician as an intermediary may become defunct; see Section 4.

The third group listed in this section is **the regulatory group**, including governance staff, third-party auditors, and government agencies. The growing importance of the group reflects the fact that, as ML has become a practicable technology widely adopted by industry, many issues have become unveiled when applying abstract mechanistic ML processes within the nuanced complexity of real-world environments. Human societies, which ideally have access to the characteristic of empathy, already struggle to quantify the qualities of human existence when making ethical decisions. Thus, integrating a black-box AutoML system into this decision-making process can be high-stakes and induce more detriment than benefit, even if, by some technical criterion, it is fast and accurate. Simply put, modern AutoML, as with the ML processes it wraps around, faces issues of trust; see Section 2.4. Consequently, the role of regulatory stakeholders is to ensure compliance around specific social requirements: user safety, model reliability, outcome fairness, cybersecurity, legal adherence, societal expectations, ethical standards, etc.

Now, ML and the automation of its high-level processes are very much frontier technologies, at least with mainstream industry as the target. Naturally, as with the history of the internet, legislation tends to lag behind technological capacity, so there are currently no standardised regulatory practices or associated entities for the general industrial use of ML, let alone AutoML. Thus, at present, the focus of the archetypal auditor is solely on the final outcomes of an ML application, much like with the business group. Typically lacking data-science expertise, they are likely to assess an AutoML system indirectly and post hoc. Because of this, technical stakeholders do, in practice, take on hybrid responsibilities shared with this group, directly probing ML solutions for technical outputs related to, for example, model bias. That said, many AutoML vendors have recently invested quite a bit in this space, so there are presently an increasing number of accessible interfaces for regulatory outputs [258]. In fact, with the way the technology is evolving, the notion of an AutoML auditor may itself eventually arise, testing the suitability of a system for particular classes/domains of ML problems. In such a case, the regulator would expect not only to interact with informational outputs, e.g. as part of traditional reactive regulation, but also to provide inputs in the form of test cases, i.e. proactive regulation. As a concrete example, consider an epidemiological ML model that optimises a health budget to minimise infection within a population. A simple scenario test introduced by an auditor may reveal that the ML model removes medical support for the infected, thus 'solving' an epidemic by accelerating deaths, which is problematic. The value of scenario modelling to the regulatory group is clear, and, as surveyed previously [258], some AutoML services already provide this functionality.

The last group listed in this section is **the end-user group**, covering any stakeholder that interacts with an AutoML-produced ML solution. Any associated inputs and outputs regarding an ML model can be abstracted as queries and responses, respectively. Crucially, end-users do not typically engage with an AutoML framework directly, at present, unless the system wraps around an ML solution or provides an integrated environment for model use. Even so, it is difficult to concisely summarise interactions for end-users as ML products have so many different use cases. For instance, it is entirely possible that, with a static ML solution produced by and detached from an AutoML system, its end-user never engages with the solution factory at all. This disconnect is especially true of users that do not even directly interface with the ML model, e.g. communities that unknowingly benefit from an ML-based epidemiological decision-maker. Alternatively, as an example of an often overlooked interaction, many commercial ML-based products require installation on various computational environments, which tends to involve an end-user. This process could potentially be managed/optimised by an AutoML deployment mechanism, although we are not currently aware of such a feature being supported for widespread use. It is also possible in certain situations for end-users to receive outputs *about* an ML model, e.g. simplified explanations detailing why a prediction was made. Such outputs become increasingly desirable if the role of the end-user starts to blur with the business group, e.g. a client driving an ML application.

The nature of end-user interactions can become even more complex once the static assumption for ML models is discarded. If an AutoML system is responsible for maintaining and monitoring an ML solution, i.e. AutonoML, and if the inputs of an end-user are driving ML model updates and adaptation, then it follows that the user is, perhaps unknowingly, interacting with the AutonoML machine. Such commentary is perhaps better relegated to the discussion of idealised fully automated systems in Section 3, given that there are no popularised ML products powered entirely by an AutoML back-end, let alone an autonomously adapting one. Nonetheless, it is worth highlighting that end-users do already impact ML models via active learning, e.g. by correcting a spam-email filter. Whether or not these adaptive mechanisms should be considered under the umbrella term of AutoML when they do not involve certain core functionality, such as model search, is debatable. However, it remains clear that end-users, as the intended beneficiaries of an ML model, greatly influence an AutoML system, whether this influence is relayed by the other stakeholder groups or provided automatically via direct/indirect feedback.

2.2 Roles and Modes Within the Machine Learning Workflow

While the previous section demonstrates a simple categorisation of stakeholders that interact with AutoML by their roles and motivations, it also shows how difficult it is to concisely capture every interaction and associated input-output flow that each role requires. This challenge is partially due to the fact that a complete end-to-end ML application involves many different processes. Thus, to better grapple with the diversity of expected interactions, this section expounds a perpendicular perspective of HCI anchored in the conceptual framework of an ‘ML workflow’, briefly introduced in Section 1, which has also underpinned our three other AutoML focused reviews [70, 148, 258].

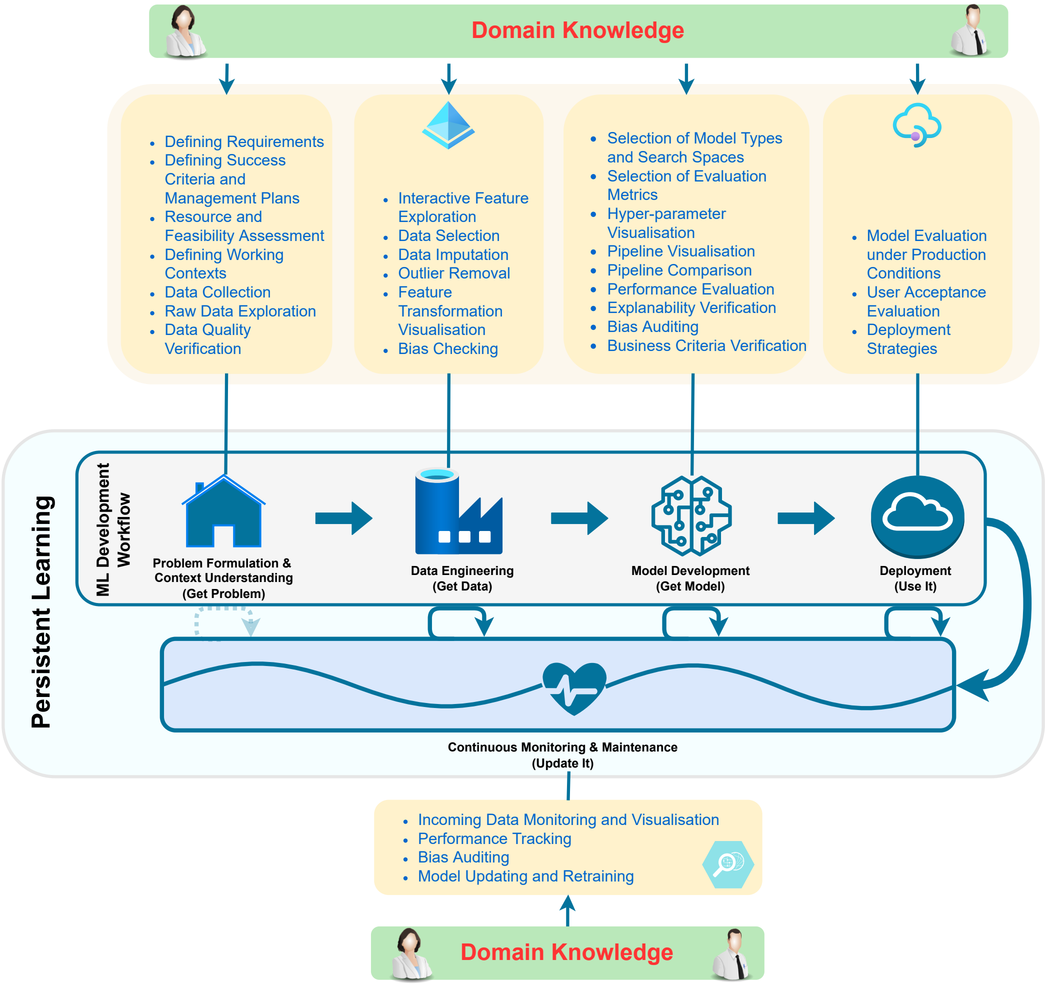

Domain Knowledge
Defining Requirements
Defining Success Criteria and Management Plans
Resource and Feasibility Assessment
Defining Working Contexts
Data Collection
Raw Data Exploration
Data Quality Verification
Interactive Feature Exploration
Data Selection
Data Imputation
Outlier Removal
Feature Transformation Visualisation
Bias Checking
Selection of Model Types and Search Spaces
Selection of Evaluation Metrics
Hyper-parameter Visualisation
Pipeline Visualisation
Pipeline Comparison
Performance Evaluation
Explanability Verification
Bias Auditing
Business Criteria Verification
Model Evaluation under Production Conditions
User Acceptance Evaluation
Deployment Strategies
Persistent Learning
ML Development Workflow
Problem Formulation & Context Understanding (Get Problem)
Data Engineering (Get Data)
Model Development (Get Model)
Deployment (Use It)
Continuous Monitoring & Maintenance (Update It)
Incoming Data Monitoring and Visualisation
Performance Tracking
Bias Auditing
Model Updating and Retraining
Domain Knowledge

Fig. 2. Summary describing the roles and modes of human interactions at various phases of an ML workflow.

As an initial overview, an ML workflow comprises of five phases, which we discuss as follows: problem formulation and context understanding in Section 2.2.1, data engineering in Section 2.2.2, model development in Section 2.2.3, deployment in Section 2.2.4, and continuous monitoring and maintenance in Section 2.2.5. Their typical chronological arrangement is displayed in Fig. 2. Historically, loosely speaking, the automation of high-level ML operations began with hyperparameter optimisation (HPO) in the model development phase before gradually extending out along the ML workflow. Indeed, in the years since an ICML-2014 workshop[2] formally introduced the term

[2] https://sites.google.com/site/automlwsicml14/

'AutoML', elements of data engineering have been absorbed into the endeavour, while, most recently, the notion of 'MLOps' has similarly brought attention to automated deployment [258]. Problem formulation and context understanding are, however, virtually untouched by AutoML, while monitoring and maintenance, prerequisites for autonomous lifelong learning [148], are also nascent areas of research and development. Frameworks that attempt to handle true AutonoML are scarce, although one proposed general-purpose architecture [139], experimented with in later studies [14, 15, 140–144, 251], is an early pioneering example of a successful system attempting to automate the majority of an ML workflow.

Crucially, most academic research into AutoML is driven by the desire to process ML operations independently of human interaction. It has not been a typical priority in this setting to facilitate HCI. Only very recently has there been a shift back towards opening up an architecture, driven by the technological emergence of AutoML in the mainstream. There is now a growing expectation that humans should have the ability to configure/control AutoML processes at all phases of an ML workflow or at least wherever an AutoML package is designed to operate. The range of these possible interactions are broad; a summary of where humans make decisions during an ML application, both optionally and necessarily, is included within Fig. 2. These interactions are associated with particular portions of the workflow, and we now proceed to discuss each phase individually.

2.2.1 Problem Formulation and Context Understanding

The first phase of the ML workflow covers problem formulation and context understanding. Its purpose is to translate stakeholder aims, typically the goals of a business, into a concrete ML task with clearly defined objectives. This translation of a real-world problem is a crucial step, given that the core of an ML application is an ML model/algorithm, which attempts to learn, i.e. approximate, some desirable mapping between query and response space [148]; the conversion must be appropriate. As such, this phase not only includes the collation of raw data sources, assessing their quality for driving a learning process, but also involves verifying that an intended project is feasible to approach with ML techniques [280]. Technical and business stakeholders are expected to collaborate closely at this stage, fostering a high-level understanding of the problem context and its associated data environment, often undertaking exploratory analysis and, if proactive regulatory stakeholders are also involved, identifying and eliminating bias factors. Of course, not all related processes during this preparatory phase are data-specific. Even beyond establishing the requirements and scope of an ML project, e.g. defining success criteria from both domain and organisational perspectives that ML solutions have to meet, many logistical tasks remain. These include the formation of development and risk management plans, the examination of prior art to assess novelty and feasibility, the general allocation of human and computational resources, the establishment of collaboration/communication practices among team members, etc.

By and large, existing AutoML frameworks have not focussed on this phase, instead assuming that system operators will have a well understood and narrowly defined ML task all ready to process. This current state of technology is not unexpected, as the second sub-phase of context understanding is immensely challenging to mechanise, discussed further in Section 4, while automating the first sub-phase of problem formulation requires installing AutoML with 'individual agency'; this veers into topics that are presently science fiction. Nonetheless, in the absence of mechanised formulation/understanding, several AutoML services exist that support humans in developing their own context understanding, e.g. by automating algorithms for exploratory data analysis. Likewise, from the application planning perspective, certain vendors do embed mechanisms to assist collaborative engagement with an ML project, e.g. team communication channels and version control. These have been noted as part of the preceding survey [258].

It is well known that visualisation plays a crucial role in this phase of an ML application, particularly when generating and communicating the results of exploratory data analysis [225]. Automated visual analytics can clarify the structure and format of big/complex data, identifying missing values and other issues. More importantly, these HCI mechanisms can efficiently bring the statistical properties and shallow knowledge content within a data source to the eyes of a stakeholder [211]. In accordance with cognitive ergonomics, this lightens their mental workload, particularly with understanding a problem context, and allows greater productivity in setting up an ML project. Naturally, there are many options to assist these visualisation processes, including software libraries like matplotlib [132], toolkits like InfoVis [86], and other commercial systems such as Power BI or Tableau. How well these options integrate with existing AutoML systems differs from one to the next. As an additional side note, while the outputs of visual analytics are, by definition, visual, the inputs for HCI can vary; see Section 2.3. Vizdom is an example of a progressively interactive system that can be interfaced with via pen and touch [60].

As aforementioned, collaboration between stakeholders is vital for most sub-tasks within this phase, which remains true of exploratory data analysis. Business domain experts and data-science technicians must reach a consensus on the data/requirements to propagate further along an ML workflow, especially as they often do not share the same perspectives due to their different roles and responsibilities [185]. Interactive visual analytic tools on electronic whiteboards, such as Vizdom [60] or Northstar [159], reflect one approach for facilitating a cooperative strengthening of context understanding. When different stakeholder groups are, in concert, able to transform, visualise and analyse various characteristics of raw data, this can leave them better informed about the types of questions to seek ML-based answers to, as well as which features and subsets of raw data to supply for subsequent phases of an ML workflow. The pooling of domain knowledge can also identify which data sources are low-quality, irrelevant or, worse, counterproductive when relying on ML to seek informative patterns. Indeed, the early detection of unstable entries, abnormal data distributions and other undesirable outliers can be handled via manual/automatic exclusion [280], reducing the risk of poor ML performance. Importantly, all data-specific operations at this stage involve only a rough form of curation; serious data engineering is left for the next stage of an ML application.

Finally, beyond establishing organisational practices to tackle an ML-solvable problem and curating/exploring relevant data, the preparatory period is also the time to make other ML-specific decisions; this is equivalent to configuring an AutoML system ahead of running an ML application [323]. Specifically, constraints can be set within the architecture in anticipation of what the business goals require. For instance, if explainability is important, model search may be configured to eschew DL networks in favour of decision-tree variants. As another example, a meta-learning module can be activated if prior experience is judged helpful for a new ML problem. Technically, such interactions are associated with later phases of the ML workflow, especially configuration-based refinements, but it is still worth noting that, chronologically, an initial approach for solving a problem arises in tandem with its formulation.

2.2.2 Data Engineering

The second phase of the ML workflow covers data engineering. Its purpose is to produce and provide high-quality data that allows ML algorithms to effectively learn desirable patterns via the formation of ML models. This phase can be revisited many times during an ML application if subsequent phases detect any issue in the data or otherwise seek refinements. Importantly, this is also the first stage of an ML workflow that can be heavily mechanised [90, 139, 249, 251], e.g. in terms of data imputation and feature selection. However, at the current time, AutoML tools are not capable of capturing domain knowledge, which still leaves human interaction indispensable for various important aspects of data cleaning, data routing and feature engineering [319]. Admittedly, while domain experts were somewhat more important in Section 2.2.1 for the curation of raw data and its initial exploration, technicians come to the fore at this point, with many data engineering processes more focussed on constructing inflows suited to the abstract and theoretical requirements of ML algorithms. Nonetheless, collaboration is ideally maintained, and HCI can involve the control and visualisation of many processes: data imputation, sample/feature selection, feature transformation, outlier removal, etc. With regulatory oversight, this is also an opportune time to check and mitigate bias that may creep in as part of the transformative process.

We re-emphasise, as discussed within Section 2.2.1 in terms of context understanding, that visualisation is invaluable for allowing stakeholders to mine data for insights. Interactive tools, e.g. Vizdom [60] and Northstar [159], continue to facilitate easy engagement that allows for the comprehension of data and the formulation of additional hypotheses [49]. Granted, the roles of the interaction are slightly different, as stakeholders are no longer exploring data from a business standpoint but a technical one. For this reason, it is possible to imagine much less input being necessary in this phase than during problem formulation, and many automated techniques have already been developed to assist in, for example, feature selection [52]. In fact, the goal of the problem formulation phase is to manifest a workable ML task, so, with the fully automated systems idealised in Section 3, human control should theoretically no longer be required in the development of an ML model. Simply put, an ML application should be *runnable*, in a minimalist manner, without human input. However, for an ML application to perform *well*, data engineering is required, and, in practice, modern AutoML still requires much oversight in terms of validating transformations, avoiding spurious correlations, and so on. Visualisation is also beneficial when combined with explanation approaches [252] and interpretability metrics [256]. Lessons learned from practical case studies indicate that interpretability and explainability are critical requirements for measuring data quality [90].

Turning to tasks involved within the data engineering phase, we note that feature engineering is a hefty one, which, as argued elsewhere [323], can be subdivided into two steps: generating features from raw data and improving their discriminative nature, i.e. ensuring that there is a high level of mutual information between feature and target variables. The first of these two steps is very difficult to mechanise, given that the relevance of features to an ML problem depends highly on the context understanding developed in the previous phase of the ML workflow. For instance, a computer has no intuitive reason to convert a raw date into a day of the week, yet humans know well that many behaviours correlate with the weekly cycle; this relatively complex transformation can immediately provide useful information to an ML application. On the other hand, other simple transformations involving algebra or conditional logic, to boost discriminative capability, are easy for a machine to explore, e.g. thresholding a numeric 'day of the week' variable to serendipitously convert into a binary 'is weekend' variable. Thus, AutoML research has mainly targeted this latter step of feature enhancement [323]. Even then, modern mechanisms for automated feature engineering, while substantially explored [148], are far from perfect, and, in practice, even this step involves heavy interaction from technicians, with useless/unstable features often discarded by hand.

Of course, technicians are involved in many more sub-tasks related to data engineering. For instance, as with features, i.e. the columns of a traditional dataset, instances of data, i.e. the rows, must also be selected carefully. Supplying a model with the best sampling of training data requires careful assessment to eliminate uninformative noise. Histograms, box plots, scatter plots and magnitude-shape plots [61] are all popular graphical tools to detect outliers. Similarly, attention from data scientists may also be required to handle missing values. If the relevant data instances are not excluded, presumably with additional domain-expert input, stakeholders must decide on an appropriate method for data imputation. Depending on data types and learning purposes, missing entries can be imputed by relying on mean, median or prediction values. These values can be generated by, for instance, learning models [27], matrix factorization [2] or a technique involving multiple imputations [204]. Often, there is a risk of introducing inappropriate artifacts into training data via these techniques, and, in many cases, these will not be identifiable until later phases of the ML workflow, i.e. after a model has been run. Technical stakeholders thus need to execute and compare the results generated by different imputation techniques to select the best approaches for specific problem contexts. Accordingly, it is not uncommon for AutoML packages to facilitate HCI for data review and, where appropriate, the modification of imputed values with domain knowledge.

Finally, it is worth stating that human input does not inherently improve a learning system, especially when humans are responsible for curating data and designing AutoML subsystems. Problem formulation and data transformation frequently infuse ML model inputs with existing human biases. These may appear in the form of sensitive attributes with associated causal impacts and may also manifest via the under-representation or over-representation of certain groups [212], e.g. in features relating to gender, age, race, country of birth, etc. Naturally, ML models may reproduce or amplify such biases and related discrimination [147]. A more detailed discussion of the issues involved is delayed until Section 2.4.4. For now, it is essential to highlight that the end of the data engineering phase is a prime opportunity to identify and mitigate biases within samples, labels and generated features. Somewhat ironically, however, it is still the role of human stakeholders to catch and correct the flaws of other human stakeholders, generally via exploratory data analysis. It is not even clear whether this will ever truly be mechanised, as recognising biases may require automated context understanding; see Section 4. To be fair, there are a growing number of tools that assist in this pursuit, e.g. Amazon SageMaker Clarify [118] and AI Fairness 360 [21]. Such tools tend to flag issues via various statistical metrics, e.g. statistical parity difference measures, class imbalance rate and the conditional demographic disparity in class labels. However, while many existing metrics exist for structured data, bias detection methods are far rarer for multimodal data that includes unstructured formats such as text, images and video.

2.2.3 Model Development

The third phase of the ML workflow covers model development. Its purpose is to yield a good model or a combination of models, i.e. an ML solution, which meets the defined requirements for an ML problem. Of all the phases in an ML workflow, this one has received the most significant attention in automation efforts. Numerous AutoML packages tackle the challenge of both HPO and model search, with approaches ranging from simple grid/random search to more sophisticated principled strategies, e.g. Bayesian optimisation. However, configuration search spaces are typically vast, especially in the case of AutoDL [70], and the computing resource requirements in both time and memory are immense. Moreover, optimal outcomes rely not just on the availability of learning algorithms but also on previously specified business goals and the data that is supplied. It is well known that there is no gold-standard ML solution applicable to all types of problem, with specialised algorithms outperforming general models for many tasks [280].

To some extent, research efforts have explored meta-learning [172] to improve model search, automatically leveraging information acquired in previous runs. One such example involves narrowing the search space for a current ML task based on the predictors that performed well for a similar problem [173, 208, 209]. However, there is still no comprehensive solution able to transfer the learned knowledge effectively between different tasks [319]. With an exception of the previously mentioned pioneering automated machine learning framework [139] where meta-learning has been used as one of the fundamental integrated concepts in multiple ML model development, monitoring and maintenance stages, the AutoML community has only included meta-learning approaches in a handful of other systems, mainly academic, e.g. warm-starting either the initial hyperparameters of optimisation algorithms [89] or, as part of Auto-sklearn, the learning algorithms themselves [88]. For now, humans are still, in many cases, better at learning from previous experiences and generalising knowledge for new tasks. The importance of facilitating HCI at this phase of an ML workflow is thus evident, with empirical evidence demonstrating that AutoML outcomes can be improved by the inclusion of domain knowledge in the search process [285, 313]. In practice, this manifests as technical stakeholders selecting constraints for the search space, including/excluding hyperparameter configurations and model types, possibly pipelined with preprocessors. Technicians are also expected to select the performance metric with which to optimise this search, e.g. classification accuracy or mean squared error.

We refer to previous reviews for a detailed discussion about the technical mechanisms involved in ML model production [70, 148]; one can infer from these monographs that an AutoML system with good HCI support should optionally allow a human to interface with all the control/feedback flows it lists. However, few existing AutoML frameworks provide that easy access, acting as black-box systems instead. This lack of clarity in operation and the resulting lack of trust in outcome are significant factors preventing widespread uptake and use [168, 313]. In fact, while the ML transparency debate has often focussed on understanding model outcomes, it has been suggested that it is an equally important requirement for both data and model-constructing processes [67]. Indeed, the integration of visualisations for input data distributions and feature engineering processes has been noted to improve human trust and understanding with respect to AutoML frameworks [73]. The same motivation can be extrapolated to model development. As a result, various mechanisms for transparency have recently been integrated into AutoML frameworks so that technical stakeholders can more easily interact with the underlying algorithms. This facet of HCI can be categorised into two main research directions: hyperparameter visualisation and pipeline visualisation [215].

In terms of hyperparameter visualisation, Google Vizier [106] and VisualHyperTuner [221] are two examples that display how the tuning of hyperparameters, within constraints, affects resulting model performance. An alternative, ATMSeer [313], which is built on an existing AutoML framework, extends its visualisation to multiple scales. Users can configure a search space, analyse sampled solutions in real-time, and explore the effects of tuning variables at the level of hyperparameters, hyper-partitions, and models. However, this is not necessarily enough. As mentioned earlier, ML solutions are sometimes sought as pipelines of data-transforming components, where the predictor connects to preprocessors and postprocessors; these can include data cleaners, feature engineers and ensemble aggregators. Such a setup has an analogue in the layered representation of a DL network [70]. Unsurprisingly, finding ways to depict these ML pipelines and show them to stakeholders in a speedy manner is key to efficient comprehension and ongoing human guidance for ML solution search [261], whether for debugging or search-space refinements. One example of a pipeline visualiser is AutoAIViz [315], which utilises ‘conditional parallel coordinates’ to represent sequential pipelines and their hyperparameters in a two-level hierarchical structure. Specifically, AutoAIViz depicts a chain of ML components at one level and then the hyperparameters for each component at the next. However, its utility is limited by the capacity to show only three components in sequence, i.e. two transformers and one classifier. It cannot portray long pipelines with optionally stacked predictors, let alone complex structures like directed acyclic graphs [249–251].

To date, very few visualisation tools exist that can genuinely grapple with sufficiently complex ML pipelines. One notable anomaly is PipelineProfiler [215], exemplified in Fig. 3, which enables the interactive analysis of complex ML pipelines for technical stakeholders. Such an interface appears to be the way forward for transparent HCI and model development processes in AutoML, supporting perspectives ranging from a high-level overview of pipeline architectures to the low-level depiction of a hyperparameter search space for a selected pipeline. It is also capable of comparing and

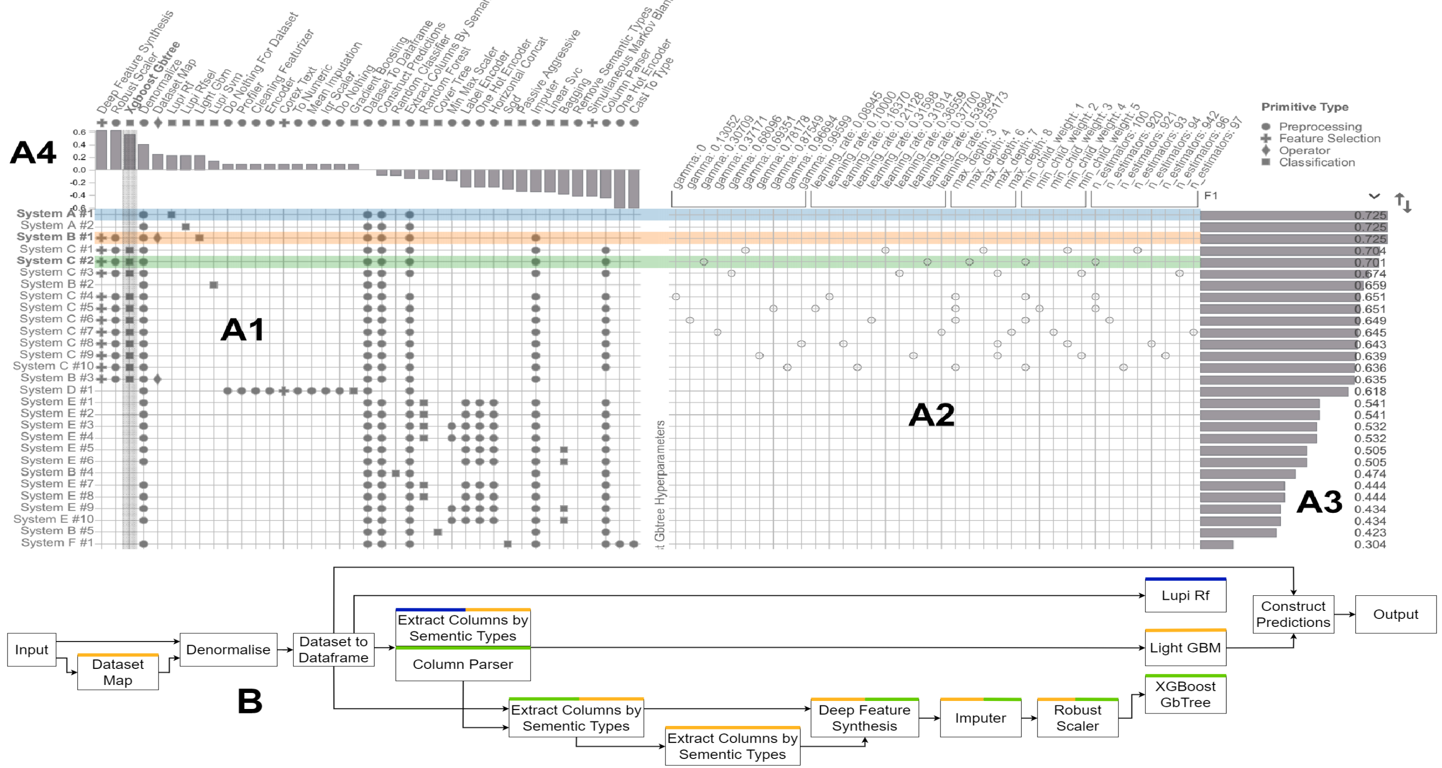


Fig. 3. An example of an interactive user interface in *PipelineProfiler* [215]. A1: A pipeline matrix showing primitives (columns) belonging to the pipelines (rows). A2: A hyperparameter matrix contains the hyperparameter values corresponding to the selected primitive *Xgboost Gbtree* for different pipelines (rows). A3: Pipeline scores, where users can choose different metrics to rank the pipelines. A4: Primitive contribution panel presents the correlation values between primitives and the pipeline score (in this case, the feature selection method *Deep Feature Synthesis* has the highest correlation with the F1 measure). B: Pipeline comparison panel visually contrasts selected pipelines in terms of structures and the used primitive components.

assessing multiple ML pipelines across different automated learning systems, although, admittedly, the profiler is only integrated with packages in the DARPA Data-Driven Discovery of Models ecosystem [177]. Of course, comparisons of ML solutions are fundamental to any process of model selection, whether fully automated or human-assisted, so any tool that captures the contrasts helps facilitate HCI. Another example that does this is ClaVis [124], which trades the ability to deal with complex pipelines for a very comprehensive suite of model-agnostic visualisation capabilities. Supporting arbitrary classifiers, ClaVis aids users in ranking them by different metrics, provides a scatterplot correlating hyperparameters with model performance, clusters predictors with similar prediction results, generates a history line chart to analyse training behaviours, and constructs confusion matrices for detecting class-specific misclassifications.

It is essential to perform an initial evaluation process with human intervention during model development to assess the quality and performance of models generated automatically by AutoML tools. In this step, technical stakeholders and business experts assess the models based on performance metrics and comparison to other pipelines. However, they also verify the explainability of learning systems to end users, solution robustness to biases, and the satisfaction of outcomes concerning predefined business criteria. It has been suggested that learning systems should be assessed on at least seven complementary properties, including performance, robustness, explainability, scalability, model complexity, resource demand, and fairness [280].

In addition to performance, explainability is becoming crucial in human-AutoML interactions when interpreting outcomes, thus developing human trust around automated learning systems, enabling the discovery of new insights and supporting informed decisions [76]. Therefore, one of the roles of humans in the evaluation step is to assess the explanation mechanisms attached to an automated learning system, e.g. those based on rules, examples involving factuals/counterfactuals, and visualisations. Certain research [302] has examined rule-based and example-based explanations, assessing their impact on human understanding, their ability to convince, and their effect on task performance when a human makes decisions assisted by learning systems. Empirical results showed that rule-based explanations have a small positive impact on understanding system outcomes, while explanations based on both rules and examples can make users follow the guidelines more often even when they are incorrect. However, neither of these types of explanations can help users improve task performance or make correct predictions in new situations. The authors explained this by a lack of clarification for the rationale of underlying behaviours of the systems. In contrast to these other types, visualisation is a natural way to overcome this drawback. One example of a visual analytics framework [274] for interactive and explainable learning systems is designed to target three user groups: model users, model developers, and business users. The first two groups, i.e. model users and model developers, are familiar with using or developing ML models, and so this group needs to understand, diagnose and refine constructed models in a given application context. In contrast, business users are non-experts in ML and they only care about how to apply learning models for specific domains. Therefore, the explanation mechanisms for each group of users with different professional backgrounds require several specific configurations; information flows for each group were discussed in Section 2.1.

It was previously pointed out that there is a considerable gap between current studies on AutoML systems and requirements regarding ethical factors [90]. Therefore, other aspects that need to be evaluated in this phase are the bias and fairness of learned models. Because learning models rely mainly on data that is collected and created by humans, whatever biases exist in human activities they can also enter the learning models, potentially being propagated and amplified [212]. Therefore, bias factors need to be evaluated and adjusted after building the learning models. If white-box learning models are used, developers and experts can modify their internal processes. For example, classification rules can be corrected, probabilities in Bayesian models can be adjusted, or the class label at leaves of decision trees can be changed. If black-box learning models are constructed, different explanation mechanisms are required to verify and understand the reasoning behind the results. Their predictions can then be altered if bias factors are detected. For instance, the proportion of decisions between protected groups and unprotected groups can be adjusted by changing the predictive results of instances close to the decision boundary [146], and experts can manually check these potential causes of discrimination. Several existing tools, such as Amazon SageMaker Clarify [118] and AI Fairness 360 [21], also provide metrics to audit the fairness of the trained learning systems. ML developers and experts can use them to evaluate and correct the learning systems in an attempt to produce fair results.

Finally, in the model development phase, developers and domain experts also need to reevaluate learning systems by the criteria posed in the first stage of the project, both in terms of business and ML. Technical stakeholders typically expect to check what the models learned by carefully observing the features with a high impact on the outcomes and verifying whether they are reasonable and consistent with business domains. If any criterion is not satisfied, the model development process should be repeated from the data engineering stage to verify and assess the quality of training data.

2.2.4 Deployment

The fourth phase of the ML workflow covers development. Its purpose is to situate a trained ML model within a computational environment, supplied by sufficient resources, and allow end-users to benefit from its learned knowledge. This process occurs after the construction and evaluation of an ML model, with technical and business stakeholders deciding whether the obtained model can be deployed for practical use. A part of this procedure involves technical stakeholders evaluating learned models under production conditions. First, the operational abilities of the ML solution need to be verified against the available computational resource constraints, working environment, and services. A potential risk that can arise is that production data is substantially different from training data. As a result, the actual performance of the learned model has to be incrementally assessed under the operational environment and conditions. The ML solution may be adjusted according to the deployed hardware and the production environment for each incremental run. In some essential cases, domain adaptation approaches [325] can be used to improve the generalisability of the learning models. While the construction of learning systems is regularly carried out in offline mode, inference and associated decision-making in a realistic environment must often operate online under specific energy budgets and security standards. In practice, these tasks need to be evaluated by deployment engineers and ML experts.

When the ML model obtained by AutoML tools passes all evaluation criteria on the deployment environment, it may face user rejection, with its usability overestimated during the development process. Therefore, the ideal practice is to test a prototype with end-users in the deployment phase. It is crucial to assess whether these stakeholders understand the query responses provided by the trained model. User guidelines around functions, business rules and domain concepts related to ML models can potentially be generated automatically [102] to provide end-users with explanations for the functionalities, operational mechanisms and restrictions that drive an ML solution.

With different version control systems for trained models and data, an incremental deployment strategy can minimise undetected errors in the development process. The system can then be rolled back to the previous version when an error is detected. In this way, the cost of fixing errors can be minimised. In practice, various deployment environments inevitably exist in the real world, and each possibly has its own characteristics in terms of latency, energy use, memory, and so on. AutoML techniques can be used as a mediator for interactions with different layers of a computing system. For instance, some AutoDL works [70] attempt to automate the search process of hardware configurations for deployment, e.g. via field-programmable gate arrays [205]. However, each commercial/industrial entity possibly poses different requirements for the flexibility/rigidity of hardware constraints and ways of provisioning resources. These constraints need expert knowledge related to hardware, software infrastructure, and computational resources. They mark the roles of human involvement in the deployment process of ML solutions.

2.2.5 *Monitoring and Maintenance*

The fifth phase of the ML workflow covers monitoring and maintenance. Its purpose is to regularly, if not continuously, examine ML models after deployment for signs of performance degradation, which are accompanied by decisions on whether and how to adapt the deteriorating ML solution. As an example, consider a cybersecurity system that automatically detects and warns of cyber-attacks; the performance of this learning system should be constantly monitored and evaluated by operators. The system must regularly be fed training data that contains knowledge about new types of viruses and offensive/defensive methods, thus ensuring that the automated system still meets the design standards. Human roles are critical in such a process, covering verification, supervision, evaluation and maintenance during the operation of the automated systems. In practice, the main reason for the impairment of learning models over time is the change in data distributions [98]. Other reasons include the degradation of hardware and the updating of a system, leading to a shift in the operational environment [280].

The most effective way to detect the deterioration of a deployed learning model is by continuously monitoring its performance over time. This process requires continuous evaluation of ML model performance using the same metrics employed during its development process. All input signals and outcomes of the learning model need to be inspected by end-users and technical stakeholders via log files or visualisations for the monitoring phase. Specifically, while incoming data can be automatically validated [255], it should be corrected in real-time with inspection by users. Statistical information of incoming data should be compared to that of the training data to detect any change in the distribution. Abnormalities in the learning system can be automatically identified, followed by a notification of the relevant stakeholders [20]. However, technical stakeholders should ideally have the power to act on or disregard the abnormalities based on their domain knowledge.

Although many different methods can be used to automatically detect changes in data distribution, with types of dynamics detailed elsewhere [98], the final decision on when to update the learning system is currently left to domain experts and technical stakeholders based on their knowledge, the operational information provided by the system, the costs to update a model, and the costs of doing nothing, i.e. leaving model performance to deteriorate. Ideally, feedback provided by an AutoML system should note significant changes within input data, the number of anomalies, and the degree of any performance deterioration; this knowledge can give technical users the informed authority to trigger a model update.

When updating an ML solution, new data has to be collected to retrain any relevant models so that they can capture an identified change in data distribution. If a completely new model needs to be developed from scratch, the whole ML development process needs to be repeated, which is time-consuming. Therefore, most adaptive AutoML research aims to fine-tune models on the new data [14, 15, 248]. In some cases, new learning pipelines will be constructed. Regardless of the updating approach, deploying the new ML solution without any oversight provided by technical stakeholders and domain experts is risky. To avoid such risks, the newly updated model should be compared with previous versions to determine whether predefined criteria are still satisfied. This process should be open to end-user feedback and leverage any domain knowledge provided by relevant stakeholders.

Various automated adaptation strategies discussed in recent studies [14, 51] can be used to automatically adapt and evolve AutoML systems according to a change in data distribution. However, there are no sophisticated adaptation mechanisms currently employed in AutoML frameworks, although the principles of monitoring/maintenance are starting to bleed in slowly. DataRobot is one example that introduced a champion/challenger setup in mid-2020, aiming to manage AutoML processes and test predictive models in a production environment [292]. The champion/challenger approach deploys an original model, called the champion, and several new or retrained models, called challengers, to shadow the champion model. The challenger models compete against each other for an opportunity to be the new champion. In practice, the outcomes of this ongoing competition in the production environment are usually reviewed by technical stakeholders, who decide which model should be selected for ongoing usage. The swap-out could easily be automated, but, presumably for governance-based reasons, they ensure that "a designated approver will route her final decision to which model will actually become the new standard (i.e., the new champion)". In general, humans should monitor and evaluate the results and performance of any updated model because many relevant business factors and domain-specific requirements cannot be gauged from the training data alone. As a result, performance and fairness metrics for all model versions, new and old, are usually viewed through different visualisation mechanisms, enabling humans to monitor and evaluate the learning system over time. Of course, continuous learning and continuous adaption with the collaboration of humans may be potential future goals for the development and operation of AutoML systems; see Section 3.

2.3 The User Interface: Many Modalities

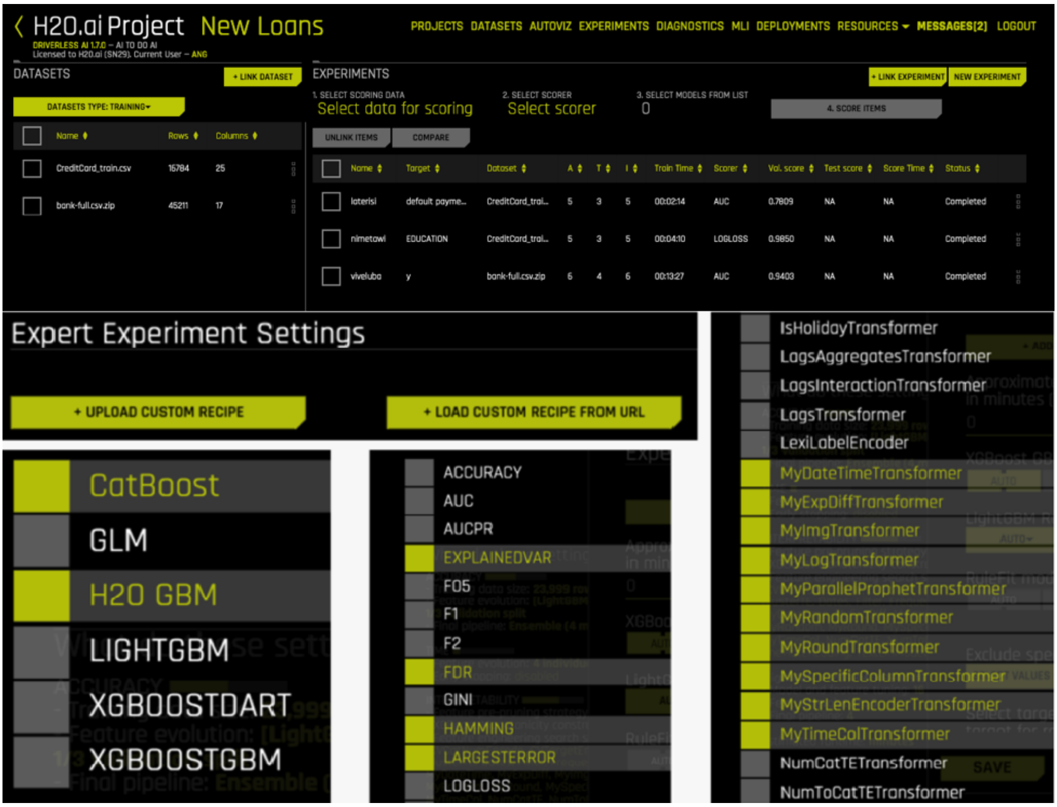
Even if AutoML systems acquire a high level of autonomy and adaptive intelligence, which is the focus of Section 3, it is difficult to imagine a world where human action is no longer required for any task. Instead, it is more realistic to see the machinery of AutoML as an assistant rather than a replacement. After all, at the current time, there is no evidence that computational algorithms can consistently outperform humans when dealing with uncertainty or when coming up with quick solutions to problems that require both cognitive capabilities and implicit knowledge. Humans and intelligent systems have complementary strengths and weaknesses, so collaboration seems key to optimising ML-based productivity [78]. Naturally, such collaborative interactions are usually performed via UIs. Therefore, having discussed the *who* and *what* of HCI in modern AutoML, we now explore *how* these interactions occur in practice. This section involves examining what basic operations the most complex ML interactions can decompose into, then discussing how these operations are presently/potentially communicated.

To begin with, it is worth noting that many academic implementations of AutoML are programming libraries; stakeholders are often required to code scripts that call these prepackaged automated mechanisms. However, in industry, most AutoML systems manage the inputs/outputs of HCI via a graphical user interface (GUI). An example is displayed in Fig. 4. Of course, AutoML frameworks often differ in GUI specifics, yet they all tend to decompose into similar sets of interactive functions. We examine these here, leaning heavily on previous categorisations related to interactive visual analytics [275, 324]. In short, at the simplest level, stakeholder intent is facilitated by (a) selection, (b) exploration, (c) abstraction/elaboration, (d) reconfiguration, (e) encoding, (f) filtering, and (g) connection. These techniques were proposed in the context of data exploration and data mining tools, but they remain appropriate for describing HCI in AutoML tools. We will additionally append the operation of value entry to this systematic listing.

The details of these fundamental operations are as follows:

(a) *Selection*: This interaction technique allows users to select visible items of interest via typical GUI elements such as drop-down lists and check-boxes (as shown in Fig. 4). The intent

[3] https://www.h2o.ai/blog/new-innovations-in-driverless-ai/


H2O.ai Project New Loans
DRIVERLESS AI 1.7.0 – AI TO DO AI
Licensed to H2O.ai (SN29). Current User – ANG
PROJECTS DATASETS AUTOVIZ EXPERIMENTS DIAGNOSTICS MLI DEPLOYMENTS RESOURCES MESSAGES[2] LOGOUT
DATASETS
+ LINK DATASET
DATASETS TYPE: TRAINING
EXPERIMENTS
+ LINK EXPERIMENT
NEW EXPERIMENT
UNLINK ITEMS
COMPARE
4. SCORE ITEMS
Expert Experiment Settings
+ UPLOAD CUSTOM RECIPE
+ LOAD CUSTOM RECIPE FROM URL

1. SELECT SCORING DATA: Select data for scoring
2. SELECT SCORER: Select scorer
3. SELECT MODELS FROM LIST: 0

□	Name	Rows	Columns
□	CreditCard_train.csv	15784	25
□	bank-full.csv.zip	45211	17

□	Name	Target	Dataset	A	T	I	Train Time	Scorer	Val. score	Test score	Score Time	Status
□	laterisi	default payme...	CreditCard_trai...	5	3	5	00:02:14	AUC	0.7809	NA	NA	Completed
□	nimetowi	EDUCATION	CreditCard_trai...	5	3	5	00:04:10	LOGLOSS	0.9850	NA	NA	Completed
□	viveluba	y	bank-full.csv.zip	6	4	6	00:13:27	AUC	0.9403	NA	NA	Completed

■ CatBoost
□ GLM
■ H2O GBM
□ LIGHTGBM
□ XGBOOSTDART
□ XGBOOSTGBM

□ ACCURACY
□ AUC
□ AUCPR
■ EXPLAINEDVAR
□ F05
□ F1
□ F2
■ FDR
□ GINI
■ HAMMING
■ LARGESTERROR
□ LOGLOSS

□ IsHolidayTransformer
□ LagsAggregatesTransformer
□ LagsInteractionTransformer
□ LagsTransformer
□ LexiLabelEncoder
■ MyDateTimeTransformer
■ MyExpDiffTransformer
■ MyImgTransformer
■ MyLogTransformer
■ MyParallelProphetTransformer
■ MyRandomTransformer
■ MyRoundTransformer
■ MySpecificColumnTransformer
■ MyStrLenEncoderTransformer
■ MyTimeColTransformer
□ NumCatTETransformer
□ NumToCatTETransformer

Fig. 4. An example of GUIs in an AutoML tool (H2O Driverless AI[3]).

of *selection* can also be communicated via more sophisticated actions, e.g. dragging and dropping blocks into boxes, as employed by KNIME [24] and RapidMiner [125]. In this way, stakeholders can parameterise an AutoML system by ordering the use of specific data transformation methods, ML algorithms, hyperparameter ranges, and evaluation metrics.

(b) *Exploration*: This interaction technique allows users to observe and inspect varying subsets of data samples via a GUI. The Direct-Walk approach, e.g. using hyperlinks, is one standard implementation that smoothly navigates a view between informational contexts [324]. For instance, when plots of input data samples are shown, clicking on a specific sample can reveal information relating to that selected sample.

(c) *Abstraction/Elaboration*: This interaction technique allows users to adjust the focus and granularity of data representation on demand. Elements of a GUI that support such an operation typically enable navigation between several levels of detail, from an overview to narrow specifics, depending on the needs of a user. Such GUI components can take many forms, including tool tip texts that detail specific data items wherever a mouse cursor is moved over, treemaps that facilitate many drill-down operations, and zooming mechanisms to change the scale of representation. As an example, ATMSeer [312] presents an overview panel showing the high-level statistics of an AutoML process alongside a separate profiler view; this profiler allows users to traverse down a search history of ML models and hyperparameter configurations at varying granularities of detail.

(d) *Reconfiguration*: This interaction technique provides users with multiple ways of examining data/results through changes in their representation. It is equivalent to the adage, “look at things from a different angle”. Often, the *reconfiguration* technique is employed to modify how data items are arranged and aligned, e.g. sorting columns or rows in tabular views. Other examples include using jitter on dense plots to avoid overlap or modifying attributes on scatter plot axes. Ideally, a new and clearer perspective allows the extraction of new information from observed data/results.

(e) *Encoding*: This interaction technique provides users with the ability to change the fundamental visual representation of data items, e.g. colours, sizes, shapes, fonts, and orientations. Closely related to *reconfiguration*, the technique of *encoding* goes even deeper into the format of data/results, e.g. changing from a pie chart to a stacked column chart. Another typical example of this operation is using visualisation toolboxes to specify the colour for certain variables.

(f) *Filtering*: This interaction technique allows users to change the set of displayed items via conditions and constraints. Specifically, this involves users providing or selecting a condition as input, after which data items are included in or excluded from a view based on whether they satisfy the condition. Check-boxes and sliders that allow the selection of conditions/ranges are GUI elements capable of supporting the *filtering* operation and, consequently, dynamic query control [275].

(g) *Connection*: This interaction technique provides users with the ability to highlight associations or relationships among data items presented within the same view or across different views. Such connections can also be automatically found/displayed, e.g. selecting a data point in one plot and seeing where it sits within another. The KNIME system exemplifies this by applying *connection* to data points selected by a brushing method, thus cross-highlighting them within other active plots, e.g. so stakeholders can see the same sample in both a line and scatter plot [24].

(h) *Value entry*: This interaction technique is used in most existing AutoML toolboxes to enable the specification of variables, e.g. hyperparameter ranges or performance-metric thresholds chosen without sliders. It may seem like an obvious interaction that all GUIs should provide, especially for the insertion of domain knowledge, but it is not strictly necessary. Because all other interaction techniques are sufficient for results analysis, value entry operations can be excluded for systems with default behaviour. Of course, if this default behaviour is effective, then the AutoML system can be treated as fully automated; see Section 3.

Showing how these various forms of basic operations contribute to interfacing with an AutoML system, Table 2 provides a breakdown of complex interactions across the entirety of an ML workflow. However, to develop this section, we now consider modalities in which such intent may be communicated with an AutoML system. Granted, it may initially seem that there is not much to discuss here. Presently, most AutoML systems are controlled via standard keyboard/mouse user inputs. Likewise, feedback is typically communicated visually via computer monitors, with warning/completion signals possibly transmitted by audio. However, ML models produced by the higher-level AutoML systems are often implanted within production/deployment environments where inputs/outputs can take the form of complex representations. For instance, an ML model query-response cycle could take the following form: gestures to machine code to audio. This fact becomes important when noting that clean distinctions between an ML model and its encompassing AutoML system are not always possible. Interfacing with an ML model can be functionally identical to interfacing with an AutoML system, especially those integrating tightly with auxiliary services. So, forms of interaction with an ML model must be included in any discussion of HCI and AutoML.

Table 2. An example of complex AutoML interactions across the ML workflow decomposed into basic operations.

Stage	Complex interaction	Input	Output	Basic operations
Problem formulation & Context understanding	Data exploration visualisation	Raw samples and features	Graphs and plots	Selection, Exploration, Reconfiguration, Encoding, Abstraction/Elaboration, Filtering, Connection
Data engineering	Interactive feature exploration	Samples and features	Selected features, graphs and plots	Selection, Exploration, Reconfiguration, Encoding, Abstraction/Elaboration, Filtering, Connection
	Data selection	Samples	Selected samples, graphs and plots	
	Data imputation	Samples, data imputation methods	Imputed samples	Selection (types of imputation), Value entry
	Outlier removal	Samples with outliers, outlier handling methods	Outliers are removed; graphs and plots	Selection (types of outlier handling, types of visualisation), Exploration, Filtering, Connection
	Feature transformation visualisation	Samples and features, feature transformation techniques	Transformed features, graphs and plots	Selection (types of feature transformation, types of visualisation), Exploration, Filtering, Connection, Abstraction/Elaboration, Encoding, Reconfiguration
	Bias checking	Samples and features, bias checking measures	Outcomes regarding the bias measures, graphs and plots	Selection (bias checking measures, types of visualisation), Exploration, Filtering, Encoding
Model development	Selection of model types and search spaces	Model types, parameters	Selected models and value ranges of the selected parameters	Selection (models and parameters), value entry (parameters)
	Selection of evaluation metrics	A list of evaluation metrics	Selected evaluation metrics	Selection
	hyperparameter visualisation	Parameters and visualisation types	Graphs and plots	Selection, Exploration, Abstraction/Elaboration, Filtering, Connection, Encoding, Reconfiguration
	Pipeline visualisation	Pipelines and visualisation types	Selected pipelines, graphs and plots	Selection, Abstraction/Elaboration, Exploration
	Pipeline comparison	Pipelines	Graphs and plot for the selected pipelines	Selection, Abstraction/Elaboration, Exploration, Connection
	Performance evaluation	Performance metrics	Selected performance metrics and results, graphs and plots	Selection, Exploration, Filtering, Abstraction/Elaboration
	Explanability verification	Explainable metrics	Selected explainable metrics and results, graphs and plots	Selection, Exploration, Filtering, Abstraction/Elaboration, Encoding
	Bias auditing	Testing samples and features, bias checking measures	Outcomes regarding the bias measures, graphs and plots	Selection (testing datasets, bias checking measures, types of visualisation), Exploration, Filtering, Encoding
Deployment	Model evaluation under production conditions	The constructed learning models and production conditions	Results, reports, and plots	Selection (of conditions and measures), Exploration (of results), Abstraction/Elaboration
	Deployment strategies	Deployment methods and environments	The deployed learning systems	Selection, Value entry, Exploration, Abstraction/Elaboration
Monitoring and Maintenance	Incoming data monitoring and visualisation	Incoming data and requirements	Operating outcomes, graphs and plots	Selection (measures and data), Exploration, Filtering, Value entry, Abstraction/Elaboration, Encoding, Reconfiguration
	Performance tracking	The learning systems, input data, performance metrics	Results, graphs and plots	Selection (performance metrics, visualisation types), Exploration, Filtering, Abstraction/Elaboration, Encoding

Bias auditing	Testing samples and features, bias checking measures	Outcomes regarding the bias measures, graphs and plots	Selection (testing datasets, bias checking measures, types of visualisation), Exploration, Filtering, Encoding
Model updating and retraining	Updating and retaining methods	The updated learning models	Selection, Value entry, Exploration, Filtering

Of course, although modern AutoML systems do not use exotic modes of communication for control/feedback, this is not due to any technical barrier, as is the case with the fully automated systems idealised in Section 3. The technology is there; it is merely an issue of what stakeholders currently desire/require. So, we take a broad scope when discussing the present/potential communication modalities with AutoML systems, summarised in Fig. 5.

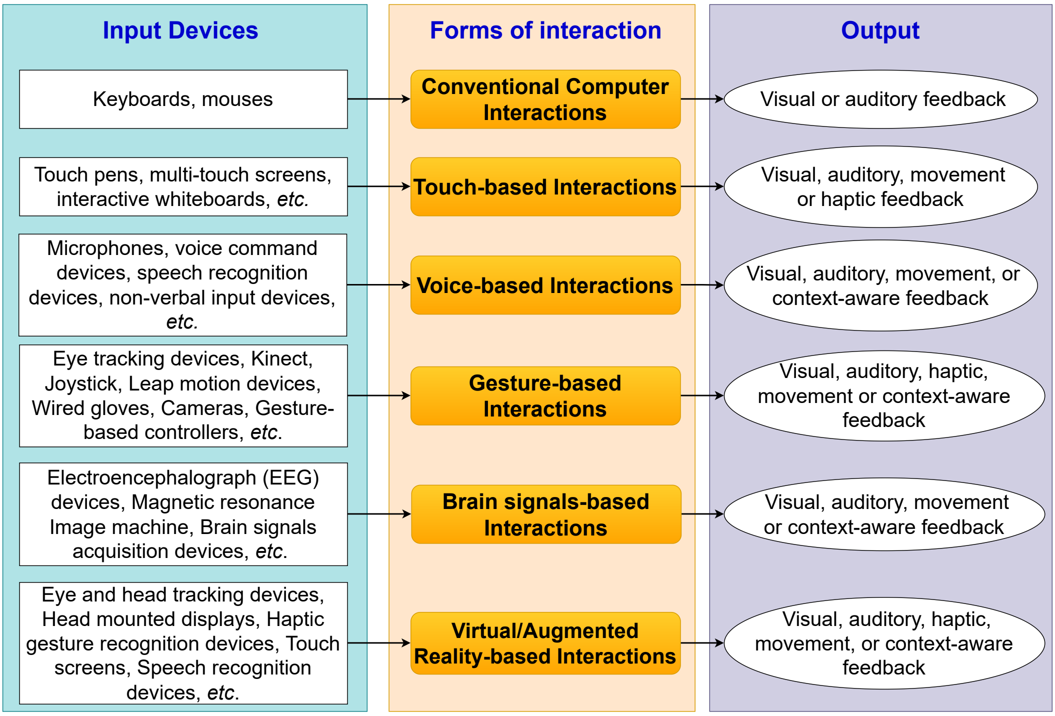

Input Devices
Forms of interaction
Output
Keyboards, mouses
Conventional Computer Interactions
Visual or auditory feedback
Touch pens, multi-touch screens, interactive whiteboards, etc.
Touch-based Interactions
Visual, auditory, movement or haptic feedback
Microphones, voice command devices, speech recognition devices, non-verbal input devices, etc.
Voice-based Interactions
Visual, auditory, movement, or context-aware feedback
Eye tracking devices, Kinect, Joystick, Leap motion devices, Wired gloves, Cameras, Gesture-based controllers, etc.
Gesture-based Interactions
Visual, auditory, haptic, movement or context-aware feedback
Electroencephalograph (EEG) devices, Magnetic resonance Image machine, Brain signals acquisition devices, etc.
Brain signals-based Interactions
Visual, auditory, movement or context-aware feedback
Eye and head tracking devices, Head mounted displays, Haptic gesture recognition devices, Touch screens, Speech recognition devices, etc.
Virtual/Augmented Reality-based Interactions
Visual, auditory, haptic, movement, or context-aware feedback

Fig. 5. Communication modalities for HCI with ML solutions and AutoML solutions, exemplifying input devices and forms of output feedback.

Beyond the conventional keyboard/mouse and monitor/audio feedback loop already mentioned, touch-based interactions are perhaps the next most pertinent to AutoML discussions, especially given the popularity of smartphones, tablets and other touch-based accessories. Already, touch-based interactions have been widely deployed for communication with learning systems in aviation [101, 195, 243], industrial production [107], and autonomous vehicles [22, 309]. Likewise, the AutoML community has already begun exploring this domain of communication, with interactive UIs designed for touch screens as part of, for example, the Vizdom [60] and Northstar [159] packages. Unsurprisingly, this modality, if implemented well, has several benefits. For instance, it is easier for touch screen devices to field multiple sources of inputs than traditional computer accessories, thus supporting greater collaboration between technical users and domain experts for both data exploration and progressive model development.

Another modality of interest is voice, given that humans typically speak faster than they type. Voice-based communications can be less efficient at certain forms of control as they do not inherently provide spatial information, e.g. like a cursor. However, if translated appropriately, they can be particularly efficient for value entry and, in well-defined interfaces, selection. Indeed, voice has previously been used to facilitate HCI between learning systems and users, including those with intellectual disabilities. Example domains include navigation [240], robotics [114], and self-driving cars [130]. Voice has also been considered as a supplementary mode of HCI alongside other modalities [169]. Of course, voice interactions require learning systems to have an automatic speech recognition module that can transform vocal content into computer-readable outputs. Ideally, the systems should also have a way to synthesise responses and a mechanism to make those responses intelligent, i.e. partake in dialogue. Recently, AutoML methods have been applied to the automated construction of speech recognition modules. For example, Deepgram AutoML [276] is the first framework designed to make training/deploying speech recognition modules for practical applications more efficient. However, for now, this is as far as the field of AutoML and the topic of voice have crossed over; we are not aware of any substantial AutoML systems with voice-operated control.

Vision-based interfacing is another emergent form of communication between humans and learning systems, seen as both natural and cost-effective [184]. The use of gestures and eye gazes belongs to this category, having been proved capable of controlling robots and other machines. It relies on computer vision approaches to detect and understand ocular motion or the configuration of human hands. This communication modality has two requirements: hardware devices to capture visual structure/motion and learning algorithms to interpret commands inherent in these visions. Within this category, gesture-based interaction is already widely used for many practical applications of learning systems. For example, real-time control of robotic arms with hand gestures has already been shown possible [233], and, elsewhere, an interactive semi-autonomous control interface has allowed human-robot collaboration to facilitate various coarse motions in robotic arms [231]. Similarly, HCI based on ocular activity is also used in different practical sectors, e.g. for monitoring and control in the field of teleoperation when the hands of a user become busy [184]. Relevant domains include mobile robots [48] and the control of smart wheelchairs [314]. While AutoML does not appear to have leveraged vision-based interfacing yet, such a modality seems to be a natural fit with certain operations, e.g. tracking gaze during data/results analysis to abstract/elaborate/explore.

In recent years, interfaces based on brain signals have become potential methods of control/interaction with vehicles, robots, and ML models [184]. In this type of interaction, it is essential to acquire electroencephalogram (EEG) signals through specialised devices and analyse them. In most cases, an ML module will be used to translate these electrical signals in a human brain into intent or cognitive stages, after which these mental commands are used to control and communicate with applications [175]. Robotic teleoperation is one domain where neural control seems promising, but many challenges are still to be overcome. Decoding neural activities and mapping these signals to motion commands join a list of complications, including latency, noisy/erratic low-dimensional user commands, and asymmetric control signal inputs [202]. In practice, it seems the most promising approach is to distinguish between levels of operation, using human-in-the-loop supervision to communicate high-level intent and facilitate task planning, then use robotic systems to automate lower-level processes with reduced reliance on user control [59]. Given that neural control is still in its infancy, it is difficult to predict how it would factor into an AutoML system, e.g. whether it would allow the communication of finely resolved intent or whether it would be better reserved for directing a maximally autonomous system; see Section 4.

Augmented Reality (AR) is the final modality we list, argued to be appealing for real-time HCI with automated learning systems due to its enhanced UX [19]. An intuitive interface based on AR can strengthen the interactions between users and a working environment, in which virtual components are inserted into human perception of the real world. Such a UI enables users to leverage their senses and engage more closely with information residing in the working environment. In effect, the productivity/quality of human decision-making can be improved. Thus, AR has been used for many applications in the industry of smart manufacturing [19], with interfaces usually employed for the quick maintenance of manufacturing systems. Integrated with continuous monitoring mechanisms, an AR interface can provide maintenance technicians with on-site context awareness and safe operational procedures [287], as well as real-time instructions from remote experts [83]. Interfaces based on AR are also appealing for their instructional capability, explored previously via on-site technical training programs that were designed to teach trainees real-environment skills quickly [194]. The AR modality has also enabled technical users to train machine operations [46] via real-time role-playing interactions that convey intent to robots without the need for offline training [47]. Many other practical applications related to collaboration between humans and automated learning systems can be found elsewhere [19]. In addition to AR, Virtual Reality (VR) can likewise be used to build interactive training applications for end-users in many domains, e.g. games, medical applications, rehabilitation, and industrial/commercial training. Methods that automatically construct, adapt and assess training content in VR environments, based on learning models and expert knowledge, are reviewed elsewhere [306]. Unsurprisingly, given the abstract nature of many ML operations, AR/VR has not yet been substantially explored in AutoML. However, it is not hard to imagine 'tangible' AR/VR objects eventually representing system objects/mechanisms, e.g. virtual blocks for the components in an ML pipeline.

Finally, to cap off this section, we briefly look beyond the current state of AutoML and communication modalities that, in some form, are within easy reach. First, we note that most of the above discussion assumes humans dictate the information communicated to an AutoML system. Likewise, the assumption is that they can operate with all the feedback they receive. However, as AutoML systems are expected to operate more autonomously in managing their ML tasks, they will likely become much more complex. Consequently, beneficial HCI does not just rely on supporting many convenient communication modalities. AutoML UIs should also be context-aware, providing users with *only* the information and interactivity that are essential for their current situation [107]. Context-awareness requires learning systems to be equipped with various sensors, aggregating raw data into higher-level context information that can be used to understand and adapt UI inputs and outputs, respectively, in real-time.

Additionally, with AutoML systems projected to evolve from relatively simple tools into more multi-faceted entities, there is no reason to imagine that all the previously listed modalities cannot be supported. Consider an autonomous car as an analogous example. Drivers can interact with the car using touch screens. Ocular tracking systems can be used to analyse distracted behaviours and promote safety. Gesture-based control is usually deployed to interact with vehicular navigation systems, as well as video, audio, and air conditioning systems [130]. Voice-based interactions can, via inputs, be used to control entertainment devices or detect emotional responses to the driving environment, and, via outputs, voice can convey instructional information back to the driver. Moreover, it is known that understanding the cognitive/physiological state of a driver is crucial to building effective driver assistance systems and autonomous vehicles. Hence, interactions based on brain signals have also been proposed to detect, analyse and monitor driver state [328]. Additionally, AR-based interactions can be used to increase driver awareness by highlighting elements present in a natural environment, e.g. road signs, crosswalks or pedestrians [30]. Perhaps, similarly, an AutonoML system may eventually be much more closely integrated with daily human activities across a full spectrum of communication modalities, possibly being called upon, at will, to apply ML techniques to practical problems in the environment of a user. Such speculation is left to Section 4.

2.4 Improving the Outcomes of Interactions

Having discussed the *who*, *what* and *how* of HCI in AutoML, we now delve into the *why*. For many years, particularly in academia, the development of AutoML theory/technology has been driven by the goal of making machines take over ML operations from humans. So, why has there been such a recent shift towards facilitating better HCI between users and AutoML systems? There are several reasons, but they all acknowledge that, however advanced and autonomous AutoML machinery becomes, ML is ultimately performed for humans. If a human is not satisfied by the outcomes of their interactions with an AutoML system, the system will simply fall out of use. So, this section looks more closely at the big issues bringing attention to HCI in AutoML as of the early 2020s.

Simply put, beyond matters of practical usability, the greatest obstacle for AutoML uptake is trust; see Section 2.4.1. It is clear that the explainability of AutoML systems is vital to confident stakeholder engagement, discussed in Section 2.4.2, and we examine current approaches towards boosting this quality in Section 2.4.3. Of course, even with a better understanding of AutoML operations, humans still bring cognitive biases to their interactions with ML-processing computers, and we detail these in Section 2.4.4. Indeed, fairness in the outcomes of ML applications is likewise crucial to fostering trust, elaborated in Section 2.4.5, and we examine current approaches towards mitigating biases in Section 2.4.6.

2.4.1 Towards Trustworthy AutoML

Apart from matters of utility and performance [258], AutoML systems and the ML models they produce are only used for practical applications – without controversy – if they are explainable and fair. Fundamentally, mechanised algorithms need to be trustworthy, and trust has previously been considered an adhesive quality for interactions between humans and ML systems [263]. It is argued that the purpose of building trust between humans and intelligent systems is to allow stakeholders to foresee the behaviours of learning systems in the presence of risk [134]. Here, we review what affects human perception/trust of automated systems from psychological and societal perspectives.

Recent survey results indicate that stakeholders continue to hold a negative view of decisions generated automatically by ML algorithms because of their potential risks [8]. Experiments show that this negativity is minimised when users understand the characteristics and operations of algorithms, suggesting that transparency is a crucial factor in making AI systems more ubiquitous [263]. Another recent study similarly indicates that the accommodation of explanatory means within learning systems boosts trust [264], allowing users to, via transparency and accountability, comprehend underlying decision-making processes. In fact, contrary to assumptions that practitioners may not want to involve themselves in the details of an ML algorithm, research suggests that users usually do wish to understand how algorithms work, at least to assess whether the outcomes are fair and reliable [317]. Ultimately, if one reviews the literature, it quickly becomes apparent that trust in ML is closely linked to stakeholder satisfaction in several interrelated concepts: fairness, accountability, transparency, and explainability (FATE) [264]. The higher the satisfaction, the greater the trust.

Admittedly, the FATE acronym is not necessarily exhaustive. For instance, people frequently assess explanations based on prior knowledge and beliefs, not just their comprehension of learning algorithms. Thus, the concept of causability has been proposed to measure the quality and effectiveness of explanations to users; the associated metric assesses whether stakeholders attain a strong causal understanding of results within a specified context [127]. Other experiments [264] make a stronger argument, stating that, apart from simply being a metric for explanation quality, causability is an antecedent of explainability and forms an essential element in trust. The data science community will likely continue to debate the best way to systematise these concepts. However, it appears clear that facilitating FATE and causability in AutoML systems would strengthen the quality of collaborations between humans and user-centred learning systems.

In considering the design of trustworthy systems, some have discussed leaning on the notion of interpersonal trust in sociology [134], as divided into two types: intrinsic and extrinsic. It is suggested that intrinsic trust appears only when users can successfully understand the actual reasoning process of learning systems, one that is consistent with either the logic of human reasoning or the priors held by a user. In contrast, extrinsic trust can be achieved through the persuasiveness of model outcomes and other behaviours; without mechanical transparency, a trustworthiness evaluation scheme needs to rely on this. Naturally, one criterion for building extrinsic trust is ensuring that the distribution of samples used in evaluating the quality of a trained model matches the distribution of unseen data when the models are deployed in practice [134]. In simple terms, data used to test an ML model should represent the real world.

A prerequisite for extrinsic trust, where the logic of an AutoML system is hidden, is plausibility. In general, humans need to believe that model outcomes do not outright contradict their understanding of an ML problem and an associated data environment. At the very least, seeming contradictions should be quickly resolved with the consideration of accessible insight. Otherwise, implausibility can dramatically weaken understandability [93]. That said, plausibility is a highly subjective aspect based on personal assessments of model utility and explanations. A seeming contradiction is not a sign of falsity; it only indicates that the model is currently inconsistent with a stakeholder pool of knowledge/understanding. In truth, as a crowdsourcing study shows [93], the plausibility of explanations is greatly affected by human cognitive biases and fallacies; see Section 2.4.4. Thus, to distinguish whether the error of plausibility arises from human or machine, debiasing mechanisms are necessary to integrate within a trustworthy AutoML system; see Section 2.4.6.

To conclude, trustworthy AutoML systems will ideally contain mechanisms to facilitate fairness, accountability, transparency, explainability, and causability. Intrinsic trust can be fostered by ensuring users understand internal operations or associate them with human reasoning, while extrinsic trust can be attained by seeking consistent forms of outcome evaluation and promoting plausibility. Many of these aspects are closely bundled with explainability, so we proceed to dive into this topic in upcoming subsections. However, by and large, modern AutoML systems can easily fall victim to human cognitive biases, e.g. via the input of training data or other interactions, which can adversely affect performance. Therefore, after the explainability discussion, the subsequent subsections focus on human bias factors, fairness, and design principles for HCI interfaces with unbiased protocols.

2.4.2 *The Importance of Explainability*

A recurring criticism that modern ML algorithms have faced, including state-of-the-art DL, is that they are hard to trust because one can never be sure when they will incur nonsensical or hazardous errors [187]. The decision-making process in these algorithms is typically obscured, meaning that algorithm designers themselves cannot explain their outputs. Similarly, most modern AutoML tools are also black-box systems that do not provide much human-understandable detail on how they select ML pipelines and associated hyperparameters [319]. This difficulty in interpreting and assessing the inner processes of an AutoML system makes it hard to trust the resulting ML solutions that they produce [313]. Indeed, humans do not usually adopt systems that are not explainable, accountable, tractable and trustworthy [332], especially if the reasons leading to outcomes are unverifiable. It follows then that developing explainable/interpretable learning algorithms is key to promoting confidence in their usage [134]. Some recent evidence supports this by recognising that engineering transparency characteristics into AutoML systems would significantly increase stakeholder understanding and, correspondingly, trust [73]. Of course, the topic of explainability is very much a part of HCI, as explanation is an interaction between a learning system and a human [179].

The detailed benefits of integrating explainability mechanisms into AutoML systems are many. For instance, interpretability is a prerequisite to improving the fairness of learning models, it aids the anticipation of system behaviour under challenging circumstances, and it facilitates ease of HCI [12]. Indeed, while we focus on trust in this section, the importance of explainability to practical operations should not be discounted. With a better understanding of how black-box systems generate results, it becomes easier for stakeholders to diagnose errors and refine the operation of the systems. When errors are understood to occur, particularly early, users can take appropriate actions such as reporting failures, updating systems, adding more training data, and rerunning an ML application with reconfigured settings.

Naturally, to construct learning models that are effective and explainable, many studies lean on insights drawn from the social sciences to understand explanation as it relates to human psychology [199]. Seemingly, a good explanation does not only show why an ML model makes a prediction/prescription, X, but also explains why the model selects X over an alternative, Y. Therefore, the use of counterfactuals in learning systems can play a critical role in achieving this objective because they enable people to identify and infer cause-effect or reason-action relationships between events, thus justifying past results and predicting future outcomes [43]. Rule sets can be used to build such counterfactual explanation summaries [235], providing ML-leveraging decision-makers with a high-level global comprehension of underlying models.

Ultimately, given how critical explainability is to AutoML trust and uptake, the explanation process must be robust. Explainable systems must forge human-interpretable reasoning for the decisions they make, e.g. in terms of ML pipeline selection or input-specific predictions/prescriptions, via the extraction/collation of knowledge from raw data that is compact, precise, and complete. This knowledge must be transferred to stakeholders via easily understandable means, e.g. in the form of rule sets and visualisations. A deeper dive into modern-day approaches for boosting explainability is presented in Section 2.4.3. However, while transparency and clarity are prerequisites for trust, they do not inherently induce it. Certainly, user-friendly explanations can support the plausibility of model outcomes, expose any unfairness inherent in an ML solution, and enhance performance/productivity via strengthened user engagement. Nonetheless, explainability on its own is simply a metric for information quality. Belief in that information relies on human cognitive biases and how they are managed via HCI processes; these discussions begin in Section 2.4.4.

2.4.3 The Methodology of Explainability

Having motivated the importance of explainability to user engagement with AutoML, we now review typical approaches that are either in use or proposed to support this characteristic. In the literature, two main criteria are usually used to classify explanation methods for ML systems: scope and stage [308]. In terms of scope, explanation methods can be divided into global and local types. Global explanation methods aim to understand the entire inferential process of a learning system when producing predictive results, examining conditional relationships between model output and predictive features within a complete dataset. For local explanation, the purpose is to explain why the learning system generates a specific outcome for a single sample. In contrast, the notion of stage examines when the techniques are applied, i.e. ante-hoc or post-hoc [308]. Ante-hoc techniques aim to consider and enforce explainability from the beginning of model development, while post-hoc methods leave the training process of ML models alone; understanding the behaviours of a model are left to external explainers incorporated at testing time. A deeper review of ML explanation methods is out of scope for this monograph, and interested readers are referred to recent surveys on the topic [12, 176, 200, 308] Here, we focus mainly on the formats that humans require of explanations produced by automated learning systems to explain how and why specific predictions/prescriptions are made. These output formats include rules, textual explanations based on examples/rationales, and visual explanations.

It has been argued that explainability should be studied as it arises within a purely human context, i.e. a shared meaning-making process involving both explainers and explainees [297]. In such interactions, people usually use why-questions to obtain reasons for decisions. Often, these are asked in comparative fashion: "Why was decision X made rather than decision Y?" This contrast enables explanations to be expressed as differences between an existing phenomenon (fact) and a hypothetical (foil) [178]. Moreover, a contrastive explanation tends to make communicating information in regular life both concrete and concise. A similar approach can be applied to AutoML, distilling an abundance of information into compact discrimination between two or more choices that an ML model faces. This contrastive technique can be used in rule-based explanations, example-based textual explanations, and visual explanations. For instance, consider a bank customer that is 35 years old, has black hair, earns $4500 per month and has two children. It is one thing to state that the bank rejects home loan applications "IF *customer age* < 30 AND *monthly income* < $5000". It is another thing, arguably more informative, to explain that the particular customer would *not* be rejected if their monthly income was at least $5000 greater. The use of contrastive explanations cuts through obfuscating detail and helps stakeholders identify the differentiating boundary between fact and foil, i.e. which specific conditions would make the outcomes change.

Now, as is becoming a trend within this monograph, we note that visualisation is a natural and attractive way to communicate the outcomes of an AutoML system and generated ML models to a human. This assertion applies to explanatory information as well. Many graphical tools exist, attempting to unveil the internal operating mechanisms of an AutoML framework. These include Google Vizier [106], VisualHyperTuner [221] and ATMSeer [313], which deploy graphs and scatter plots to illustrate searchable configuration spaces for HPO. Similarly, as depicted earlier in Fig. 3, PipelineProfiler portrays and compares ML pipeline structures with their parameters [215]. Depending on the problem context, features essential to model predictions can also be highlighted via various graphical means. Heat maps with different colours can be used, in the field of computer vision, to emphasise special regions of an image [238]. Likewise, in the field of natural language processing (NLP), they can bring attention to specific words in a paragraph [279].

Another intuitive form in which to present explanations to humans is the use of textual descriptions, which can be conveyed in written or spoken format [260]. They can, of course, be bundled with supporting evidence and contrastive situations to increase plausibility. For instance, consider an explanation for a predictive ML model that rates upcoming films: "Movie A directed by XYZ is likely to score a good rating because it is an action film. Three out of four action films directed by XYZ were previously rated good, i.e. C, D, E but not F. Movie B directed by XYZ is likely to score a bad rating because it is a drama film. One out of three drama films directed by XYZ were previously rated good, i.e. G but not H or I." Evidently, the use of examples aids in describing past phenomena and generalising knowledge to new scenarios [207, 224]. More fundamentally, though, textual explanations in natural language are appealing for HCI due to a couple of simple reasons [75]. Firstly, they are intuitive to laypeople, who cannot necessarily grapple with the inner structure of learning models and data characteristics in technical terms. The use of contextually appropriate textual explanations thus significantly boosts communication effectiveness with non-experts. Secondly, humans can often digest the information contained within natural language quickly, allowing for collaborative interactions between learning system and user occurring in near real-time. Both these reasons essentially boil down to the fact that the use of natural language is inherently comfortable and expressive for humans. Of course, textual explanations can always be supplemented with visualisation techniques, should deeper and more technical analyses be required [260]. For now, most modern AutoML systems do not have sophisticated explanatory mechanisms that use natural language, but research is undoubtedly ongoing in the broader data science community. There have been efforts, for instance, to train a computational model that can automatically transform the internal states and action data of an autonomous system into human-centred textual explanations [77].

Natural language, however, often uses many filler words that do not allow for information to be contained compactly; rule-based explanations are an alternative. Using rules to explain data-driven learning models allows for concise structure and logic, yet humans still find them relatively easy to understand [176]. They can provide both global and local explanations for the predictive outcomes of an ML model [92], i.e. in the form of a rule set for a query-response space and an individual rule for a specific example, respectively. Rules are typically constructed in the form of 'IF . . . THEN . . .' statements together with AND/OR operators to represent value combinations of input features and predictive outcomes. For instance, in line with a previous example, a home-loan application rule can be expressed as follows: "IF *Age* > 60 AND *monthly income* < $6000 THEN Output = Rejected". The parts before and after 'THEN' are called antecedent and consequent, respectively. The antecedent usually comprises one or more premises combined with logical operators. However, rules do not need to be crisp, i.e. adhere to classical logic, where the combination of premises leads to an entirely true or false consequent. Explanations can be expressed as fuzzy rules, where relations between antecedent and consequent are true to a non-binary degree [112]. Indeed, the integration of fuzzy rules into learning algorithms can aid with reasoning/explaining the internal logic of black-box ML models such as neural networks [217] and support vector machines [213]. Fuzzy-based sets of premises and conclusions can also be used to build arguments, which are essential components in a defeasible reasoning process. Defeasible reasoning aims to build a common-sense qualitative understanding in highly uncertain contexts, where the availability of new information can lead to the fallibility of preconditions and the withdrawal of previously generated conclusions. Given that ML models are usually imperfect approximators of a desirable function, the integration of defeasible reasoning into data-driven learning systems is appealing, potentially improving capabilities for knowledge representation/extraction, grappling with incomplete/uncertain information, and communicating the internal processes of systems/models [182].

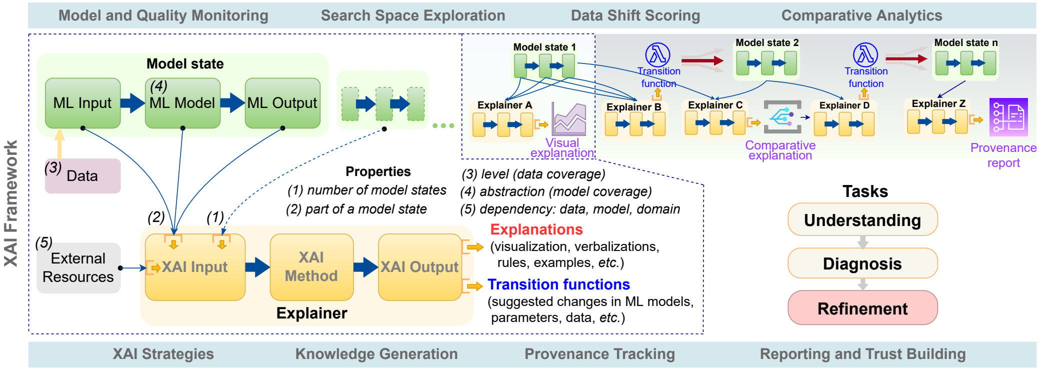


Fig. 6. A schematic of an iterative explainable ML framework aiming to understand, diagnose, and refine ML models [274]. Each 'explainer' object, possessing five properties, takes in one or more model states and data characteristics as inputs, then deploys explanatory approaches to produce an explanation or transition function as outputs. The framework also includes global monitoring/steering mechanisms, listed in the long cyan blocks, to manage the explainers.

In practical applications, different types of explanation can be integrated into automated learning systems using a single unified framework. One such example, illustrated in Fig. 6, is ExplAIner, which employs both visual and textual explanations for outcomes generated by learning models [274]. These assist in continuously interpreting, diagnosing and refining ML models during their operation, serving as ideal supporting mechanisms for technical stakeholders. The resulting explanations also aid other stakeholder groups in understanding why an ML model made particular predictions/prescriptions. In detail, the ExplAIner framework incorporates so-called 'explainer' objects that transform model configurations and input data into one of two outputs: one, explanations in different forms of visualisation/verbalisation, or two, transition functions that suggest new configurations for model refinement. Eight global monitoring/management mechanisms also support the framework: search space exploration, model quality inspection, the comparative analysis of solutions, reporting processes for trust-building, a module for knowledge generation, etc. Naturally, given the growing importance of HCI and data-driven learning systems, ExplAIner is but a single example, with many modern AutoML frameworks beginning to integrate various explanation approaches for ML solutions. For instance, Google Cloud AutoML Tables [58] provides users with feature importance explanations to visualise which features contribute most to the training process of models and individual predictions. Similarly, the Mimic Explainer package integrated with the Azure AutoML suite [13] likewise supports feature-importance explanations.

2.4.4 Human Cognitive Biases and Their Impacts

Explanation mechanisms can boost human comprehension/reasoning related to AutoML operations. Unfortunately, it is argued that, while these mental processes are typically rational and heuristic, they are subject to distortion via cognitive biases [312]. This is not a new concept; the term 'cognitive bias' was promulgated in the 1970s to describe flawed patterns of thinking in making judgements and decisions [295]. Specifically, human cognitive biases can be defined as unconscious errors in cognition resulting to issues with memory, attention and other mental facets that can impair the rationality/accuracy of judgements and decisions [245]. They are considered the results of a brain attempting to simplify a complex world to help make decisions quicker. Confirmation bias, anchoring bias, availability heuristics, the misinformation effect and self-serving bias are all common examples of cognitive bias. A more extensive list can be found elsewhere [29].

As humans play a critical role in operating AutoML systems, influencing processes ranging from training-data preparation to the constraining of search spaces, cognitive biases can be propagated through to the resulting ML models. Technically, the building of AutoML systems can also embed biases within mechanisms, but these can be harder to 'correct' if they are non-configurable. The critical point here is that human biases can potentially impact ML application outcomes, so it is essential to identify and manage them. Bias mitigation can help boost the reliability of an AutoML system and, accordingly, increase human trust in the machinery. Of course, we note that the results of an ML model can often seem more plausible if they *conform* to existing biases, so this is not a trivial topic. Somewhere along the line, a connection needs to be hypothesised between a possible bias and a negative outcome, either potential or realised, e.g. unfairness; see Section 2.4.5. In any case, this subsection simply discusses common biases that exist in ML algorithms, as well as human cognitive biases that can adversely affect a human understanding of AutoML outcomes.

Importantly, the ecosystem of HCI is complex, and human biases can distort ML outcomes at various phases of information flow. Some biases may be introduced at the very beginning by system/algorithm designers, while others may take effect at the very end, i.e. when users interpret model outputs. A recent keynote talk [104] attempts to categorise biases associated with learning systems into three groups: data bias, automation bias, and algorithmic discrimination. Data bias refers to how training data is selected. If the input data provided to the learning system is biased, then the output is likely to be biased. For instance, if an automated recruitment system is trained on candidate resumes for which males dominate the elite level of a career path for historical reasons, an ML model may incorrectly learn that male employees perform intrinsically better than female employees. This gender bias is just one context-specific example of problems arising with class-imbalanced data. Of course, in the workforce example, no bias has necessarily been introduced in the data-preparation phase; it is simply that the real-world environment does not represent the equitable function one wishes an ML model to learn. However, data bias is possible even if a real-world distribution is otherwise unbiased, simply because humans can curate raw data with uneven sampling. Whatever the case, the bias in this data can propagate down an ML workflow and integrate into its outcomes [120]. Other forms of data bias include data heterogeneity, where data is pooled from various sources with different formats, e.g. text, images, audios, and videos. Many ML algorithms often work with one data source that is assumed to be independent and identically distributed (*i.i.d.*). Data heterogeneity usually violates this assumption [174], weakening the validity of an algorithm. Yet another form of data bias links to the issue that high-dimensional data often results in overfitting, with ML algorithms subject to the curse of dimensionality [69]. This issue is usually tackled by reducing dimensionality via feature engineering/selection, but this procedure introduces biases of its own.

Next up is automation bias, which can be considered a fallacious 'appeal to authority', the authority here being an AutoML system. Specifically, humans tend to accept suggestions produced by automated decision-making systems, even in the presence of contradictory information [104]. Unsurprisingly, an over-reliance on automation leaves users exposed to the many potential biases inherent in such systems [8, 116, 326]. The concept of 'death by GPS', i.e. accidents caused by uncritical acceptance of driving commands, is one dramatic example of automation bias. Admittedly, it can seem antithetical to promote trust by removing a bias already correlated with trust, but misplaced faith is dangerous, often leading to extreme scepticism in a technology when that faith is broken. It is also not contradictory to claim that AutoML faces trust-based obstacles in uptake while also stating that automation bias is widespread; most contemplation around engaging in a service occurs, for better or worse, before that engagement begins. Regardless, the risks of automation bias are clear, especially in high-stakes applications, and ideal AutoML systems must either remind users to remain critical or otherwise facilitate both explainability and accountability.

Finally, algorithm bias refers to systematic errors within high-level AutoML mechanisms and low-level ML algorithms. Such errors are often an unavoidable part of generalising patterns and approximating functions. The issue is that any systematic errors in an ML model, including an inability to capture nuance within a problem context, can lead to adverse outcomes, e.g. discrimination via the preferential allocation of privileges to those with wealth [56]. As a side note, definitional boundaries are fuzzy in the broader literature, so algorithm bias often overlaps with data bias, sometimes referring to the replication of unethical societal biases hidden within training data. There are many examples of such an occurrence in real-world applications. For instance, previous research found evidence for racial bias in the US health care system, with favouritism shown to white patients over sicker black patients [214]. It was discovered that an associated algorithm incorrectly correlated high health care costs with the sickest patients, prioritising them for intensive-care programs. On the surface, this conclusion does not seem unreasonable, i.e. using health costs as an indicator of health needs. However, because of systemic societal racism, black patients are less likely to receive the health care they need, thus spending less money than white patients. Effectively, the model did not consider the nuance of health care availability. In such a way, ML algorithms can learn, amplify and even legitimise institutional biases [104]. Granted, given that algorithmic biases are systematic and repeatable, one of the best countermeasures is to know what assumptions underlie a procedure before employing it. However, this remains a significant challenge for modern AutoML, which simultaneously opens up ML technology to non-experts while, for the most part, obscuring its internal processes.

Thus far, except for automation bias, we have mainly focussed on errors distorting model outcomes. However, the final step of an ML application is often conveying predictions/prescriptions to a stakeholder, and cognitive biases can certainly affect understanding these. Therefore, ideal AutoML systems are expected to optimise the communication of model results, not just the model itself. This assertion is especially true as, nowadays, learning algorithms are increasingly seen as important socio-technical artefacts [153] embedded in many societal/organisational structures, associated with a diversity of relationships, incentives, and roles in society [8]. Accordingly, a recent study argues that explainable AI must be designed to express helpful explanations while avoiding misunderstandings due to typical biases faced by human reasoning, e.g. representativeness bias, availability bias, anchoring bias, and confirmation bias [312]. Another study similarly discusses the potential impacts of cognitive biases – it highlights twenty popular types – on the way humans interpret learning models, particularly inductively learned rule-based systems [155]. Ultimately, careful design of how rules and other forms of explanation are represented seems to be the best path for avoiding human cognitive biases during AutoML-specific HCI.

2.4.5 *Fairness*

Biases have many consequences, but both researchers and the public have recently highlighted one particular concern related to ML applications and, by association, AutoML systems: fairness. Admittedly, the concept of fairness is challenging to define, with many varied attempts in the literature, and there is still no consensus on the matter [254]. This state of affairs is because fairness is not an objective statistical notion but a subjective socioethical concept [193], one that can change over time and between problem contexts. Indeed, different cultures can have discordant perspectives on what fairness means and how important it is. That said, for decision-making, fairness is principally defined as the absence of bias/favouritism towards an individual or group based on an inherent characteristic [57]. These ‘protected’ characteristics usually include age, religion, gender, skin colour, ethnicity, etc. Other definitions of fairness may focus on either individuals or groups, seeking similar treatment for different entities. This discussion already leans into a deep topic, with distinctions sometimes drawn between equality and equity; the former allocates every entity the same resources/opportunities, while the latter acknowledges varying circumstances and aims for equal outcomes.

From a technical view, many metrics have been proposed for fairness in ML. Consider a binary classifier with predictions divided into two groups, each associated with a typical confusion matrix consisting of true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN). Attempts to ensure fairness may seek to satisfy the following constraints as defined across the two subgroups: (1) demographic parity, i.e. equal $\mathrm{TP} + \mathrm{FP}$, (2) predictive parity, i.e. equal $\mathrm{TP}/(\mathrm{TP} + \mathrm{FP})$, and (3) equalised odds, i.e. equal $\mathrm{TP}/(\mathrm{TP} + \mathrm{FN})$. However, barring exceptional circumstances, it is generally impossible to satisfy the equality of all such metrics at the same time [154], and the above three are often cited in the ‘impossibility theorem’ of machine fairness. Therefore, trade-offs and prioritisations based on problem context are required to apply fairness in practical ML. In the abstract sense, this may mean, for instance, choosing to prioritise equality/equity for either individuals or groups but not both. Unfortunately, a theoretically fair ML model may still lead to biased decisions in practice because the decision-making process is ultimately subject to human interpretation and operational environments. Worse yet, while it depends on the dataset, enforcing outcome equality between subgroups often comes at the cost of overall model accuracy.

While fairness is difficult to achieve in practice, unfairness is much easier to identify. For instance, one can define six forms of discrimination [193]. Direct discrimination occurs when an outcome, e.g. demographic parity, is influenced by the protected features of individuals, while indirect discrimination is based on neutral non-protected features that are not independent of protected features. An example is a zip code, which, despite codifying spatial information, is often associated with ethnic and socioeconomic factors. Systemic discrimination refers to policies or behaviours discriminating against a subpopulation, e.g. if a stakeholder intentionally excludes people with disabilities during data curation. Statistical discrimination assumes individuals within a group are represented by the average qualities of that group. One can also classify discrimination as explainable or unexplainable, depending on whether an imbalance in outcome across groups, e.g. predictive parity, has a clearly understood cause.

The important point here is that, due to the subjectivity of fairness, the concept is critical to the topic of HCI in AutoML. Stakeholders must communicate to the machine what they deem to be fair, while the machine must communicate to the stakeholders whether the model satisfies this requirement. Fortunately, this endeavour can springboard off existing development in explanatory mechanisms; AutoML systems only need to incorporate some methodology to detect/identify discriminatory outcomes. To that end, one research effort proposes to use a causal model to capture underlying biases, assessing relationships between protected features and ML model outcomes [161]. Indeed, supported by explicitly causal models, the impact of ethically sensitive attributes on other attributes and ML outcomes can be presented to decision-makers, thus evaluating fairness. Of course, given the increasing attention to trust-related matters in the AI community over recent years, several AutoML vendors have already made moves in this area. Commercial frameworks such as H2O Driverless AI [133] and DataRobot AutoML [64] now include fairness metrics so that developed ML models can be evaluated in this respect. Nonetheless, the broad survey of existing AutoML tools closely linked to this monograph [258] indicates that this is very much a nascent topic, with few packages currently implementing debiasing modules. More sophisticated interfaces for the analysis of fairness will likely arise as the conversation on equitable ML continues.

2.4.6 Bias Mitigation Through Human-computer Collaboration

In real-world applications, ML models do not exist in a vacuum; end-users either interact with them directly or are impacted somehow by their outcomes. Thus, any biases internalised by an AutoML system can have adverse flow-on effects on society, e.g. by hindering the ability of individuals that are discriminated against from partaking equitably in socioeconomic activities. Naturally, there have been various ways technicians have approached mitigating these biases, and traditional methods fall under three main categories: preprocessing, in-processing, and postprocessing [63]. Unsurprisingly, as confirmed by experiment [120], combining methods at all three stages of processing can be the most effective alleviation of biases inherent within propagating data.

Before modelling, preprocessing approaches explore data transformations to eliminate discriminating factors within incoming data. In-processing methods focus on the ML model instead, imposing fairness constraints on learning algorithms during model training. Postprocessing is carried out after an ML model has been generated, usually by reassigning sample labels in holdout sets used for testing [7]. However, at present, the overall effectiveness of these techniques still seems to be lacking [56]. A significant obstacle is still the identification of biasing factors, which is why explainability research is currently so appealing in this field. Explanatory approaches boost accountability for model outcomes that usually slip under the radar for more conventional, albeit rigorous, probing methods. For instance, the integration of factual/counterfactual explanations into AutoML systems may prove useful for managing the risks arising from automation bias, clarifying reasons for decision-making outcomes and indicating which scenarios/conditions may change the results.

Now, it is challenging to decide what unbiased fairness is. Technically, as discussed in Section 2.4.5, it is actually impossible to satisfy all reasonable metrics for fairness. Therefore, generating an ML model with outcomes that are genuinely representative of a problem context and data environment requires asking a simple question: what matters? On the other hand, answering this question is no easy feat; it depends on the nuances of the social contexts in which an ML application is run/deployed, the risks associated with data curation/preparation, and the dangers of mishandling these risks. Until context-intelligent AutonoML systems arise, as speculated in Section 4, human judgement and expert knowledge will always be required to determine standards of bias/fairness in the design, implementation, operation and deployment of ML models. This human expertise may be sourced from many disciplines, including the humanities, social sciences, law, and ethics.

Of course, stakeholders must have sufficient HCI-enabled understanding/oversight of AutoML system internals to identify the reasons for discriminatory behaviour and avoid vulnerable groups suffering from bias. This scenario of necessary human intervention happens more often than some automation purists may expect. For instance, researchers investigated a model designed to assess employee churn at the Xerox corporation [187], which predicts how long workers remain in employment. The learning system identified that a key feature inversely correlating with employment duration was commute time. This association seems reasonable, and a consequent decision to maximise employee retention might be simply to prefer locals. However, after careful consideration, managers identified that commute time is indirectly a protected variable, as the company was located in an affluent area and employees with a lower socioeconomic status were based further out. Subsequently, this criterion was eliminated from model inputs. It is thus evident that biases can easily slip through an ML application without close human monitoring and other collaborative interactions.

The upshot is that AutoML systems should ideally be designed with human-computer collaboration in mind, at least for the modern era. A practical approach to human-in-the-loop decision-making involves algorithms producing recommendations or sets of options while humans double-check or select ML solutions that are both good and fair. This mode of interaction does admittedly increase human responsibility in designing, implementing, operating, monitoring and evaluating both AutoML systems and their generated ML models, so frameworks must ideally facilitate stakeholders not just identifying the biases of others but also their own. To this end, clarity is critical. Working towards transparent algorithms/mechanisms and explainable results can grant humans better judgement over how much weight to give any ML solution. As discussed in Section 2.4.3, this is partially achieved by ensuring AutoML interfaces support a diversity of analytic tools and visualisations with which human biases may be identified. Additionally, just as how ensembles of ML predictors compensate for individual weaknesses [246], it is worthwhile considering, where practical, the comparison of predictive algorithms alongside human decision-makers. While this does not always show whether human or machine is correct, discrepancies indicate that an error exists somewhere, which at least justifies the need to deploy resources for producing explanations and verifying rationale. At the current state of research and development, these are just several achievable ways that AutoML systems can be kitted out to leverage human-computer collaboration in avoiding/managing biases.

Granted, while explainability clarifies AutoML processes/outputs and can highlight biases, how these explanations are interpreted still depend heavily on human cognitive behaviour. For continued progress in this area, it is crucial to examine how humans do causal inference [12] and how they grapple with issues of ambiguity and uncertainty in learning algorithms [300]. Indeed, examining the heuristic operations of a human brain can lead to the improved design/development of ML models that are both causal and explainable [264]. Similarly, identifying factors that influence the perceptions and attitudes of humans concerning automated decision-making systems is essential beyond just the technical process of generating solutions and optimising predictive performance; these factors indicate how solution outcomes should be framed and communicated to users [170]. Some researchers have elaborated on how explanation modules can allow decision-makers to validate their thoughts and alleviate cognitive biases [312]. For instance, an AutoML system could construct/show prototype model queries corresponding to different model responses, thus mitigating representativeness bias and the illusion of similarity. Availability bias could be alleviated by displaying the prior probability for outcomes. Anchoring bias could be reduced by using rule-based counterfactual explanations to highlight input-feature importance or lack thereof. Confirmation bias could be tackled by discouraging backward-driven reasoning, e.g. by prioritising input attribution in a UI display over posterior probability or class attribution. Many proposals exist, drawing inspiration from empirical results in cognitive science, with yet another study [155] detailing techniques to address twenty common cognitive biases that impact the understanding of rule-based learning systems. In time, it is expected that these mechanisms will become increasingly common as part of AutoML technology.

2.5 The User Interface: Key Requirements

Over the previous sections, we have broadly reviewed the *who*, *what*, *how* and *why* of HCI in AutoML as it is currently seen in the early 2020s. One final question remains: given all the underlying concepts/themes that we have covered, are there any general principles for designing a good AutoML UI?

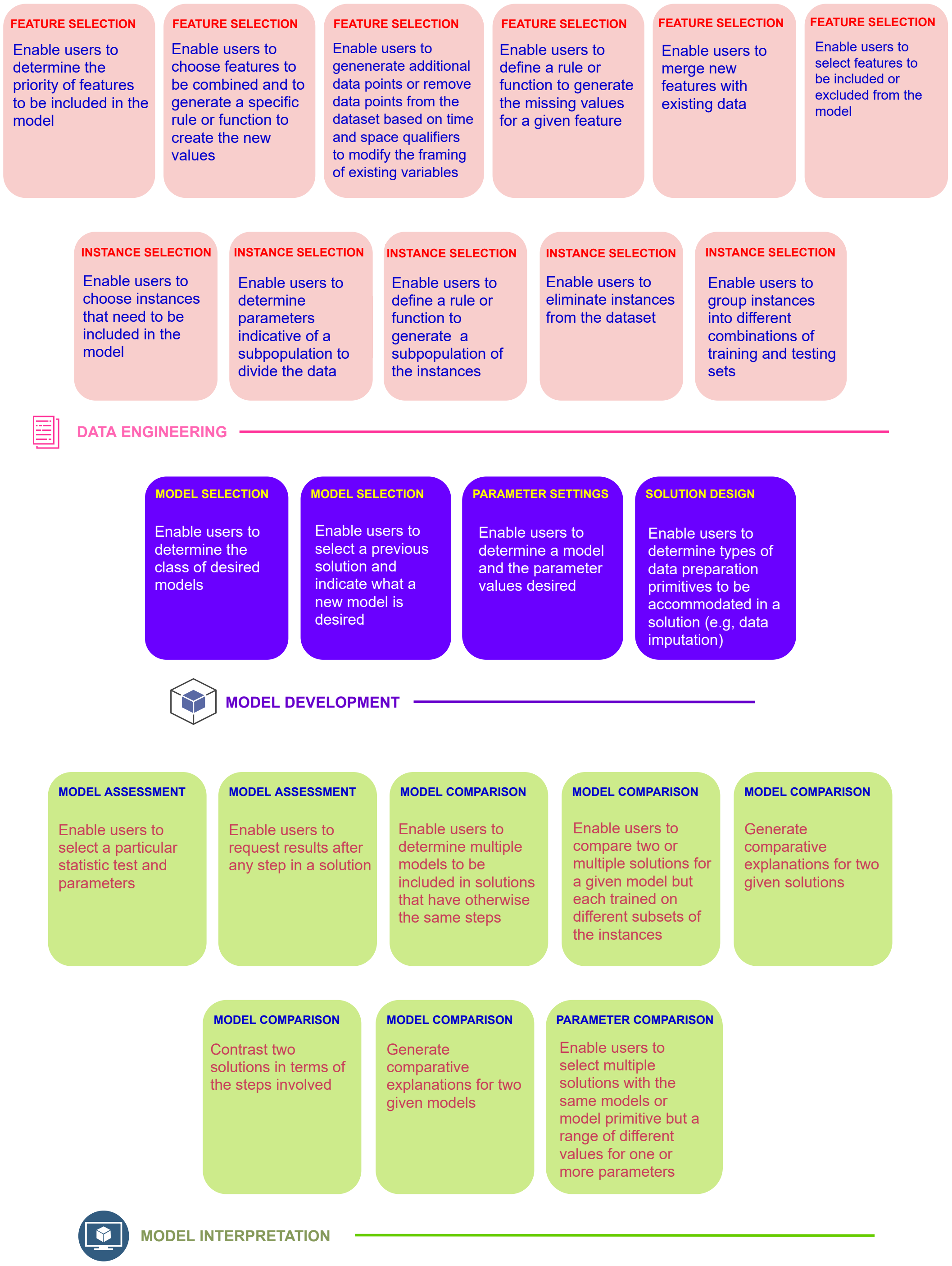
Evidently, a recurring theme in this review is that visualisation via GUI components is a natural way of fostering human understanding and interactivity around AutoML processes and generated models. Many recent studies support this notion, stating that interactive visualisation has a crucial role in the comprehension, diagnosis and iterative improvement of numerous learning models [180, 275]. Visual analytics can fill the gap between human knowledge and the insights generated by explainable AutoML systems [274]. Thus, many existing AutoML frameworks, particularly those marketed to the general public, focus on building a variety of GUIs; these aim to facilitate effective non-expert HCI at every supported phase of an ML workflow [258]. Of course, visualisation also supports technical stakeholders, enabling the inspection/control of system operations through which the learning process and ML solution search can be managed. Expectations have reached a point where experts and ML developers prefer to visualise and compare multiple ML pipelines at a time rather than inspecting them one by one [215]. Additionally, these stakeholders want to know how and why AutoML algorithms construct ML pipelines for specific problems. Indeed, as concluded by think-aloud experimental investigations [73], outcome visualisation and the provision of performance metrics are some of the most critical aspects an AutoML system should support to promote trust. Moreover, as long as human-in-the-loop collaboration remains integral to AutoML technology, explainability has additional feedback effects in enhancing the performance and reliability of ML solutions. All of these considerations provide the context in which robust UI design principles should be fashioned.

Notably, the frontier state of AutoML technology means that there is still a dearth of research into systematising the key requirements of a UI. However, with an expanded perspective, it quickly becomes apparent that there are many proposals in the literature covering guidelines/approaches for effective HCI with learning systems [6]. Many of these studies cover defensive mechanisms for avoiding undesirable actions made by a learning system, e.g. via the implementation of verification processes and control mechanisms [128], the mitigation of user-misleading interactions with adaptive systems [135], and the enhancement of transparency/explainability for mechanised behaviours [160, 232, 238]. However, research prior to the current surge of attention on trustworthy ML also exists, generally considering operational matters, e.g. balancing automated services with direct user control [129]. Historically, adaptive UIs have also received significant attention in both design principles [65] and evaluation, e.g. in terms of predictability [96]. However, it is argued that AI designers and developers continue to struggle with the creation of intuitive UIs that foster human understanding, belief and effective collaboration with automated learning systems [105]. A well-designed UI fits squarely into the topic of ergonomics, primarily cognitive; it needs to provide essential elements that assist users in reducing workload, improving work performance, and ensuring safety. Case studies indicate that the most successful UIs are usually associated with a well-defined structure of HCI based on obvious principles, validated theories, and actionable guidelines [266].

Although UI design for learning systems, examined primarily from a communication-modality perspective in Section 2.3, essentially remains a topic for the broader data-science community, there have been a handful of initial AutoML-specific forays into this discussion. One study claims that, although numerous applications may be suitably served by ML that is fully automated, humans usually possess rich domain knowledge that is valuable for constraining available data and guiding solution search procedures [105]. Therefore, the authors introduced a set of UI design principles for human interactions with AutoML systems, schematised in Fig. 7. These requirements were proposed based on how stakeholders work with AutoML algorithms and what their expectations for interactions are. Three main categories of tasks were identified, loosely corresponding to the core of the ML workflow we describe in this monograph: data engineering, model development, and the interpretation of obtained learning models. Additionally, the authors argued that human interactions with AutoML frameworks are both iterative and nonlinear. Specifically, human engagement is iterative because, over time, ML models are produced/examined, AutoML settings are reconfigured, and model search/production is rerun. Human engagement is also nonlinear because stakeholders may desire to skip steps at any time, relying on defaults for necessary operations and ignoring optional ones, or they may decide to revisit previous phases of an ML workflow at will. Therefore, UIs should ideally support these requirements. Unlike traditional fire-and-forget approaches to AutoML in academia, there is also a strong focus in the proposal on post-production evaluation, e.g. the UI must allow users to examine a specific ML pipeline, complete with its detailed steps of production, for comparison against challenger models. Accordingly, the work stresses that effective HCI with AutoML frameworks is critically dependent on *both* transparency and usability.

Of course, as of the early 2020s, the conversation on HCI in AutoML has just started. Thus, the broader concerns facing the AI community have not yet percolated into direct considerations of AutoML-specific UIs. For instance, the proposal mentioned above [105] still has a strong focus on technical performance, while establishing trustworthy ML requires grappling with many other factors, e.g. fairness, biases, and ethical concerns. Also, it is essential to remember that, at present, AutoML is still a long way from its major goals, i.e. comprehensive automation of the entire ML workflow; see Section 3. This fact is evident in that the proposed UI requirements did not systematise HCI in deployment environments, let alone consider ongoing monitoring and maintenance related to changing ML models [123].

Unsurprisingly, there are some significant challenges on the horizon for AutoML-specific UI design. By now, it should be clear that humans usually use automated decision-making systems if machine behaviours can be controlled or predicted; stakeholders expect to interrupt anything incomprehensible and potentially dangerous in a timely manner [266]. Therefore, guidelines for UI designs should factor in human understanding and control, forcing AutoML systems to explain what they did. Bonus points can be awarded for adaptive UIs that learn from user behaviour and better facilitate desired interactions in the future. However, two main complications challenge effective UI for collaborative interactions [321]: uncertainty around the capabilities of an automated learning system and complexity related to system outputs. Presumably, efforts to grapple with these obstacles will continue to filter in from the broader literature, with Fig. 8 exemplifying one such study, codifying 150 design recommendations from varied academic/industrial sources into



FEATURE SELECTION
Enable users to determine the priority of features to be included in the model

FEATURE SELECTION
Enable users to choose features to be combined and to generate a specific rule or function to create the new values

FEATURE SELECTION
Enable users to genenerate additional data points or remove data points from the dataset based on time and space qualifiers to modify the framing of existing variables

FEATURE SELECTION
Enable users to define a rule or function to generate the missing values for a given feature

FEATURE SELECTION
Enable users to merge new features with existing data

FEATURE SELECTION
Enable users to select features to be included or excluded from the model

INSTANCE SELECTION
Enable users to choose instances that need to be included in the model

INSTANCE SELECTION
Enable users to determine parameters indicative of a subpopulation to divide the data

INSTANCE SELECTION
Enable users to define a rule or function to generate a subpopulation of the instances

INSTANCE SELECTION
Enable users to eliminate instances from the dataset

INSTANCE SELECTION
Enable users to group instances into different combinations of training and testing sets

DATA ENGINEERING

MODEL SELECTION
Enable users to determine the class of desired models

MODEL SELECTION
Enable users to select a previous solution and indicate what a new model is desired

PARAMETER SETTINGS
Enable users to determine a model and the parameter values desired

SOLUTION DESIGN
Enable users to determine types of data preparation primitives to be accommodated in a solution (e.g, data imputation)

MODEL DEVELOPMENT

MODEL ASSESSMENT
Enable users to select a particular statistic test and parameters

MODEL ASSESSMENT
Enable users to request results after any step in a solution

MODEL COMPARISON
Enable users to determine multiple models to be included in solutions that have otherwise the same steps

MODEL COMPARISON
Enable users to compare two or multiple solutions for a given model but each trained on different subsets of the instances

MODEL COMPARISON
Generate comparative explanations for two given solutions

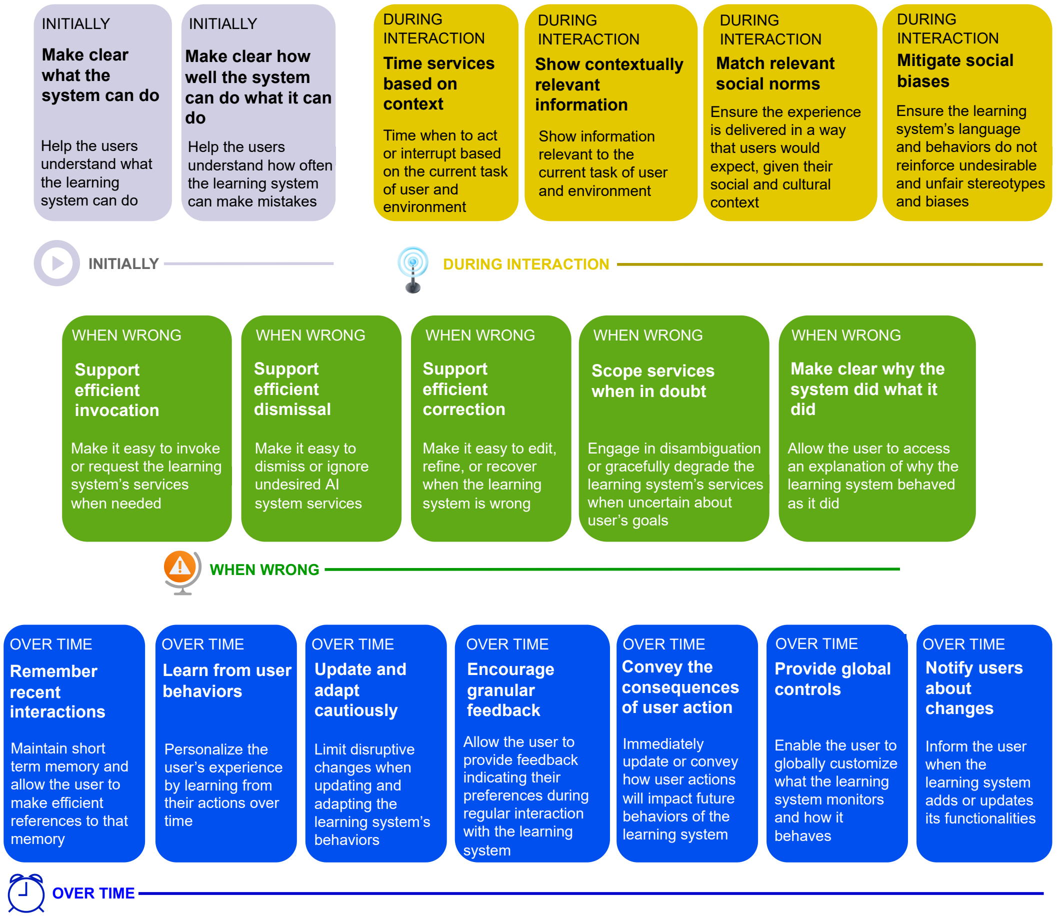
MODEL COMPARISON
Contrast two solutions in terms of the steps involved

MODEL COMPARISON
Generate comparative explanations for two given models

PARAMETER COMPARISON
Enable users to select multiple solutions with the same models or model primitive but a range of different values for one or more parameters

MODEL INTERPRETATION

Fig. 7. Conceptual schematic of HCI-related functionality that an AutoML UI should support [105].



INITIALLY
Make clear what the system can do
Help the users understand what the learning system can do

INITIALLY
Make clear how well the system can do what it can do
Help the users understand how often the learning system can make mistakes

DURING INTERACTION
Time services based on context
Time when to act or interrupt based on the current task of user and environment

DURING INTERACTION
Show contextually relevant information
Show information relevant to the current task of user and environment

DURING INTERACTION
Match relevant social norms
Ensure the experience is delivered in a way that users would expect, given their social and cultural context

DURING INTERACTION
Mitigate social biases
Ensure the learning system's language and behaviors do not reinforce undesirable and unfair stereotypes and biases

WHEN WRONG
Support efficient invocation
Make it easy to invoke or request the learning system's services when needed

WHEN WRONG
Support efficient dismissal
Make it easy to dismiss or ignore undesired AI system services

WHEN WRONG
Support efficient correction
Make it easy to edit, refine, or recover when the learning system is wrong

WHEN WRONG
Scope services when in doubt
Engage in disambiguation or gracefully degrade the learning system's services when uncertain about user's goals

WHEN WRONG
Make clear why the system did what it did
Allow the user to access an explanation of why the learning system behaved as it did

OVER TIME
Remember recent interactions
Maintain short term memory and allow the user to make efficient references to that memory

OVER TIME
Learn from user behaviors
Personalize the user's experience by learning from their actions over time

OVER TIME
Update and adapt cautiously
Limit disruptive changes when updating and adapting the learning system's behaviors

OVER TIME
Encourage granular feedback
Allow the user to provide feedback indicating their preferences during regular interaction with the learning system

OVER TIME
Convey the consequences of user action
Immediately update or convey how user actions will impact future behaviors of the learning system

OVER TIME
Provide global controls
Enable the user to globally customize what the learning system monitors and how it behaves

OVER TIME
Notify users about changes
Inform the user when the learning system adds or updates its functionalities

Fig. 8. General design guidelines for effective interactions between humans and learning systems [6].

18 general design guidelines [6]. These guidelines cover best practices for immediately clarifying system capability, managing interactions, tackling errors, and dealing with ongoing operations. Similarly, the IBM Design website[4] also provides guidelines for design foundations, albeit from broader social perspectives discussed in Section 2.4, e.g. ethics, fairness, and accountability.

Regardless, it seems inevitable that these low-level abstractions and implementations of UIs will, in time, solidify within the AutoML technology that is publicly available. As mentioned before, the steady expansion of the automation endeavour along the entire ML workflow, combined with escalating concerns around trust in ML, will require system developers to grapple with these facets one way or the other. It may be informative then to zoom out of the details for a while and forecast, with a high-level perspective, what happens to HCI once humans no longer *need* to be involved in any system operations.

[4] https://www.ibm.com/design/ai/

3 INTERACTING WITH AUTOML SYSTEMS: CONSTRAINED BUT FULLY AUTOMATED

Bluntly speaking, the concept of HCI was never fundamental to the initial drive for AutoML. After all, a primary motivator for automation is ensuring that operations of interest can be processed with as little human interaction as possible. To this end, the field of AutoML has made great strides, yet it remains far short of its end goals. In most modern AutoML frameworks [258], humans still participate heavily in configuring the system by, for example, controlling the selection of data transformations, ML models, and HPO algorithms [323]. Moreover, not only is the technology *semi*-AutoML in practice, many existing packages focus only on subsections of the whole ML workflow, e.g. feature engineering, HPO, or NAS [70, 148, 258, 331]. Therefore, at present, most AutoML services are seen as no more than tools to assist technical users in developing ML models for practical problems. Accordingly, ergonomic approaches to HCI have arisen as a necessity, not a desire, primarily to support the technical control that semi-AutoML packages require. That is not to say that there are no HCI mechanisms devoted to the broader social issues discussed in Section 2.4. However, the roles and modes of human interactions are predominantly driven by getting AutoML to work and work well.

Of course, the democratisation angle of AutoML remains. The field strives to enable domain-knowledgeable stakeholders without a technical background in data science to engage in and benefit from ML theories/practices [105, 261, 319]. To pursue this agenda and truly remove the need for an ML technician, an AutoML system must be virtually end-to-end, capable of processing almost the entire ML workflow independently. We exclude the stage of problem formulation and context understanding from this requirement; see Section 4. Essentially, given an input dataset, a genuine AutonoML system must be able to find/construct an optimal ML pipeline on its own, deploy it, and then maintain its performance over time within a production environment. It should also be capable of dealing with diverse input types, such as images, sounds, text, and tabular data. As of the early 2020s, there are only a few AutoML products that are sophisticated enough to even approach these requirements, such as AutoML by DataRobot [64], Driverless AI by H2O [133], Fully-AutoML [331], and Auto-sklearn 2.0 [87]. The first two examples, AutoML by DataRobot and Driverless AI by H2O, are commercial tools marketed as automating the entire ML workflow, from data engineering to the MLOps phase, complete with simple forms of monitoring/maintenance. The other two examples, Fully-AutoML and Auto-sklearn 2.0, only focus on data engineering and model development. All are impressive tools/systems, but, at the current time, it would not be advisable for a client to leave their ML project entirely up to the machine without any oversight.

Granted, we do not necessarily advocate the removal of human intervention. However, there is no reason not to expect the default behaviour of AutoML frameworks to get better and better over each new generation of research and development. So, we now imagine that an AutonoML system with sufficient technical capability is finally developed, one that can only solve well-defined ML problems – its understanding of context is ‘constrained’ – but one that can do so with full automation. Does the spirit of HCI change? To consider this question, Section 3.1 first discusses the possible ways in which AutoML theory/technology will progress to the ideals of AutonoML; improving search strategies is crucial to this advancement. Section 3.2 then considers what the roles and modes of human interactions are likely to be when engaging with these fully automated systems. Finally, we consider what lies beyond constrained ML problems in Section 3.3, motivating an incoming discussion on AutonoML within open-ended environments.

3.1 Served by Superior Search Strategies

In general, the core aim of ML is to find a mathematical model that best approximates some desirable query-response function. Naturally, a great deal of human effort goes into concretising this desirable function and the associated data environment in which it hides. For now, we treat this initial phase of an ML workflow, i.e. problem formulation and context understanding, as solely the domain of human cognition. In so doing, we assume that, by the time an ML problem is passed to an AutonoML system, it is well defined and firmly constrained. Given this assumption, it is reasonable to state that all remaining *operational* human interactions go into supporting the search for an optimal ML solution. These can involve transforming the data environment, defining searchable spaces for HPO, supplying the right computational resources, considering how best to adapt models for a changing desirable function, and so on. It stands to reason then that full automation of the ML workflow depends critically on improving search strategies, for every AutonoML mechanism, to the point that humans need not guide them.

Historically, advances in optimisation theory/practice have already had an enormous impact on data-science applications and technology. Notably, these advances have often shown that expert knowledge is unnecessary – sometimes inhibitory – to effective solution search in various narrow domains, e.g. protein-folding [137], Atari-based games [113], Go [267, 269], Chess [268], and so on. A pioneer of modern reinforcement learning, Richard Sutton, argues in a blog post that “the biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin” [283]. Certainly, machines are not beset by human cognitive flaws and limitations, nor do they struggle with the same learning curves [196]; again, we emphasise this is in the context of narrowly defined ML problems. Modern computational resources are also vast, even if they fall short of the postulated exaFLOP that a human brain operates at, and leveraging them efficiently/intelligently can achieve significant outcomes.

As an example of how AI exceeded human skill by relying on superior search strategies, consider the game of Go. Traditionally, ML efforts at mastering games have relied heavily on human expertise. Even AlphaGo [267], the first computer program to beat a professional Go player, involved a deep neural network trained in a supervised manner on previous games played by human experts. However, it also drew benefits from self-play; after a certain level of proficiency, it played millions of games against itself and used reinforcement learning to improve its capabilities. During this self-play phase, it was essentially provided only the rules of the game and a metric to evaluate board state, e.g. win-loss. In effect, it ceased to rely on handcrafted domain insights and human heuristics for making moves, eventually developing its own tactics for optimal gameplay.

Of course, an initial expert-knowledge bootstrapping can still influence a subsequent self-play process. So, the successor, AlphaGo Zero [269], eliminated human expertise embedded in the training data entirely. Its search environment was constrained by the context of the problem, i.e. the rules of the game, a performance metric, and the input structure of data corresponding to the board state. Then, starting from entirely random moves and initialised parameters, the system reached significant proficiency at the game, able to beat top human players within days, and achieved all this without any human data or guidance. Without human cognitive biases and preconceptions related to the game, AlphaGo Zero was effectively able to explore non-standard tactics beyond conventional Go knowledge. Since then, AlphaZero [268] has only continued this trend of advancement, able to employ a single network architecture to search and learn at expert level for multiple games, i.e. Go, Shogi, and Chess. These achievements demonstrate the potential of generic automated search-and-learn algorithms to surpass human performance in narrowly defined problem contexts. Unsurprisingly, self-play strategies have also been engineered into AutoML systems, e.g. AlphaD3M [72].

Crucially, the above example shows that the key to significant advances in automated search relies on high-level insights, e.g.-the notion of adversarial self-play to explore a space logically but, to humans, unintuitively. In a similar fashion, ignoring revolutionary changes in computational hardware, AutoML is unlikely to make sufficient progress towards full automation without significant advances in theory/practice. Search spaces for HPO are typically immense and multi-dimensional, meaning that random search has often been found competitive with principled optimisation methods [100]. A possible saviour then might be found in the notion of meta-learning, whereby meta-knowledge, experience built from encounters with other ML problems, is used to supercharge advanced search strategies, e.g. by warm-starting searches or recommending parameter values. This principle is highly appealing, grounded in lifelong biological learning, and is likely a hard requirement for opening up cross-context capabilities for AutoML; see Section 4. Unfortunately, the current accumulated successes of meta-learning have been inconsistent at best, e.g. in terms of recommending search spaces for ML models [4, 208]. It is presently unclear which settings of an AutoML system should be tuned similarly for similar ML problems, nor is it clear how to define that similarity between ML problems effectively. Additionally, while the field currently exists in an era of big data, the AI community may not have enough instances of *datasets* to begin building robust meta-models.

Despite all the obstacles, it seems obvious, conceptually speaking, that meta-learning should be central to advanced AutonoML. As early as 2009, a generic architecture was proposed to fully automate the process of developing, deploying and maintaining online predictive models [139]. It was an attempt at AutonoML before the term 'AutoML' was even propagated in the mainstream. Relevantly, meta-learning was considered an integral part of ML solution search alongside local learning and ensemble learning, with optional support for the insertion of expert knowledge. Although modern AutoML systems are arguably more advanced in various aspects, this framework remains a possible hint of how fully automated AutonoML may eventually appear. Its generic nature allows different data-engineering methods, predictive models and adaptation strategies to be easily plugged in, with automated interaction channels among these components explicitly defined. It supports both batch learning and incremental learning in a single framework and, uniquely among modern AutoML offerings, provides multiple complex adaptation approaches for lifelong learning. That said, the research did not have an agenda of forced automation, so the architecture also describes interaction methods between system operators and various elements within the whole framework. Certainly, technicians can access specific modules and manually intervene in system operations at will, with most interventions relating to process control, parameter configuration, and performance evaluation.

There are a handful of other systems that also incorporate meta-knowledge, such as the Fully-AutoML framework [331]. Without human guidance, this system can search a coarse-grained space of data-engineering and model-development components to construct an ML pipeline for different types of input data, e.g. image, text, video, speech, and tabular. Most importantly to this discussion, the proposed framework has an offline phase, during which an evolutionary algorithm searches for pipeline configurations with good empirical performance on various data sets, subsequently storing them within a given repository. These pipelines are employed as lifelong knowledge anchors and are associated with various meta-features, including descriptive characteristics for the datasets they solve, as well as ML task type, domain, and so on. The promising ML pipelines generated from previous experiences are then used to initialise a population of solutions that an evolutionary algorithm works with to propose a good candidate for a new ML problem. Recommended pipelines can consist of ML components tasked with various operations: data preprocessing, feature engineering, model selection, hyperparameter optimisation, and ensemble aggregation. Similarly, though not by evolutionary means, Auto-sklearn 2.0 [87] also employs meta-knowledge to warm-start ML pipelines of preprocessors/predictors and their hyperparameter configurations.

For now, meta-learning mechanisms remain rare amongst AutoML services, so it is difficult to assess whether their present implementation is generally effective or simply proof-of-concept. Indeed, it is time-consuming and difficult for humans to curate an appropriate collection of meta-knowledge, while automated curation risks the inclusion of low-quality meta-data. In addition, the effectiveness of meta-learning appears very sensitive to many factors, with empirical results indicating that minor differences between previous data sets and a novel ML problem can degrade performance severely due to biases in data [17]. Nonetheless, the promise of meta-learning to accelerate and enhance search strategies continues to draw research, which grows ever more sophisticated. For instance, to enhance the generalisation of meta-learning, a Wasserstein Generative Adversarial Network [10] was recently proposed to act as a surrogate model for the generation of meta-data [181]. A self-adaptive meta-model trained on this artificial meta-data was then investigated, seeking a good trade-off between accuracy and run-time.

Overall, while the above examples are only initial forays into AutoML-specific meta-learning, there is no in-principle reason to doubt that search strategies will continue to improve. In time, default AutoML operations will likely become sufficiently performant to no longer warrant technical oversight. However, not all phases of an ML workflow are equal. The pathway is arguably the longest for the stage of continuous monitoring/maintenance, and previous comprehensive reviews found research/development efforts here to also be the most nascent [70, 148], even though many real-world ML models operate in dynamically changing environments. In general, the design and implementation of adaptive learning systems face six essential challenges [334] that remain incompletely addressed: making adaptive systems robust and scalable, tackling realistic data, enhancing usability and trust, effectively integrating expert knowledge, considering different application demands, and translating algorithms into software. Unfortunately, this remains largely irrelevant to modern AutoML, as a survey of packages/services found that most still focus on static data [258]. Those that do not constrain their focus tend to have simple mechanisms for adaptation, although there is insufficient evidence to judge whether that simplicity is a negative.

Regardless of the present state, it is inevitable that AutoML will eventually have to grapple with adaptation, if only because real-world applications demand it. Indeed, in the wake of the generic proto-AutonoML architecture proposed in 2009 [139], related efforts have shown the utility of complex adaptive ML systems in the process industry, specifically with soft sensors [15, 143, 144]. Consequently, some researchers have decided that, if AutoML packages are fire-and-forget, perhaps it is possible to call them on demand as part of an adaptive strategy [248]. This thread of research was recently progressed even further, exploring the artificial combination of several open-source AutoML toolboxes and a drift-detection mechanism [51]. However, while valuable, such approaches remain workarounds; genuine AutonoML systems will probably need to internalise adaptive mechanisms if they are to operate in any sophisticated manner. Accordingly, other efforts have explored how best to automatically develop/deploy flexible adaptive strategies [14]. Ultimately, the takeaway here is that there are many interesting technical challenges left for researchers/developers to tackle in the field of AutoML. However, to contemplate HCI, it is not unrealistic to fast-forward and envisage the end-point of full automation, at least in terms of well-defined ML problems.

3.2 Roles and Modes When Human Involvement Is No Longer Required

Technical and business stakeholders shift their focus to problem formulation and context understanding. This reprioritisation is the obvious consequence of engineering AutonoML frameworks capable of fully mechanising all ML workflow phases dedicated to ML solution search. Technicians, in particular, are freed from a wide range of tedious tasks, including managing the selection/generation of features, recommending model types to evaluate, and guiding search through HPO configuration spaces. Instead, computers are assumed to automatically run these operations with sufficiently good performance, subject to the bounds of computing memory, time, input types, and solution-acceptance criteria. Accordingly, human resources can be redirected to more innovative zones, e.g. refining ML task definitions/constraints, better understanding how problem and context interrelate, sourcing more informative data sources, etc. In essence, stakeholders focus on building the best sandbox in which AutoML will proceed to play. For example, the operational environment of learning models playing Chess or Go is restricted by specific invariant game rules, alongside input-data structures representing the size and type of game boards involved. For diagnostic models used in radiological imaging, the learning environment may be constrained to specific disease types and scan formats.

The redirection of human resources can be particularly valuable, as some settings may require substantial effort in defining the problem context, especially for complex predictive/prescriptive models that manage multiple intertwined tasks. For instance, the 'brain' of a self-driving car will be subject to many factors such as physical response times, traffic laws, speed limits, road signage, lane markings, and weather conditions. The point here is that stakeholders no longer instruct the automated algorithms on how to solve a problem; they merely dedicate time and effort to ensuring that definitions/constraints are sound and watertight. These include appropriate metrics for evaluating outcomes and performance, e.g. in a reinforcement-learning setting. The AutoML system is then expected to leverage available computing power and conduct search/learning processes without the need for any human intervention, possibly supported by meta-knowledge. Without directly inheriting the biases and limitations of human intelligence and cognition, AutoML may discover entirely new ways of solving a problem that humans have never seen before [196]. Technicians can, of course, revisit a search history to validate results according to predefined criteria and then choose whether to accept them.

The other clear advantage of an AutonoML framework that is self-managing and effectively end-to-end is that the role of an end-user can also shift. With the operation of ML applications ceasing to require technical expertise, ML solutions no longer have to be built for an end-user; they can be built *by* an end-user. This assertion is simply the logical culmination of the democratisation drive behind AutoML. Research suggests that it is already feasible with current AutoML technology to give end-users a degree of that agency, allowing them to build models for their own domain-specific datasets and initial settings [171]. Admittedly, the empirical results of the study are preliminary and only involve basic models. Nonetheless, this role is likely to evolve in parallel with UI improvements and simplifying representations of AutoML. For instance, if stakeholders currently desire explainable ML models from AutoML systems, they are often forced to interact with low-level details, deciding whether configuration spaces should include/exclude predictors such as decision trees or multilayer perceptrons. In contrast, for fully automated AutonoML frameworks, end-users should need to do no more than state a preference for explainability, leaving all the rest up to the system. Many such advancements are, in principle, technically trivial, e.g. translating one marked checkbox into a series of true/false attributes associated with ML algorithms in the codebase. The onus in these particular cases is on developing robust ontologies to map human desires into machine-interpretable operations.

The strengths and weaknesses of operators and AutonoML systems become complementary. This eventuality is closely related to the previous one. Rather than humans and machines both wrestling with solution search, entities are now able to solely prioritise the strengths of their biological/mechanical intelligence, covering for the weaknesses of each other. In recent years, this perspective has become particularly popular, eschewing AutoML as an outright replacement for human roles. Instead, the purposes of AutoML are increasingly seen as augmenting the capabilities of users [183] and countering the constraints/limitations of human cognition [158]. Naturally, it is still the role of domain experts to precisely define/constrain a problem and its context, as the presently hypothesised form of an AutonoML system has neither individual agency nor a versatile understanding of concept. However, in the ideal case, operators do not expect to directly interfere with the internal mechanisms of their AutonoML 'assistant'.

For an example of this duality, consider the case of medical diagnostics, where an ML task may be to map particular patients to sets of treatments, with careful consideration of potential side effects [179]. Maintaining concentration for extended time periods is challenging for humans, especially with high-stakes monotonous tasks, e.g. reading through large amounts of clinical data. Computers do not suffer this form of processing fatigue. On the other hand, despite their extremely effective search strategies, AutoML systems have neither the rich domain knowledge that humans have acquired nor the cognitive mechanisms to form connections between concepts. This lack of context-awareness can make it impossible to understand the implications of data and outcomes. For instance, as noted by the medical diagnostics study, human experts can identify that a missing data feature might be a critical indicator for unmeasured information or hard-to-quantify objectives concerning a minority subgroup expressed in a dataset. Specifically, this is an outcome of humans applying the context understanding that technically proficient AutoML systems are incapable of [319].

The primary remit of an ideal but constrained AutonoML system is also confined to four out of five phases of an ML workflow. One can obviously speculate, but, for now, such systems do not have the same flexibility to engage with a real-world data environment as humans. For example, clinicians can adjust rigid prescriptions fired out from an ML solution by collaborating with patients and taking in their feedback. Clinicians can also come up with creative ways to alleviate side effects. This argument for rejecting automation as an outright replacement for human roles is common. For instance, even though an automated learning system could theoretically observe and interpret X-ray images, the expertise and capabilities of a radiologist go far beyond these functions [9]. A radiologist will regularly handle patient-facing tasks, e.g. ultrasound, fluoroscopy, biopsy or consulting, and they will also handle multi-disciplinary work, e.g. auditing results and reviewing discrepancies [121].

Ultimately, a human brain and an idealised AutonoML system, without context understanding, are both complementarily valuable for decision-making. In critical high-stakes fields, these decisions should likely remain the responsibility of qualified humans, but, even there, AutonoML predictions/prescriptions may be supplementarily helpful as long as they are weighted accordingly. In essence, humans and AutonoML systems will ideally reach higher performance in both speed and accuracy when acting together than apart. For example, an automated driving support system may be better at keeping a car within a lane and identifying sudden hazardous situations. At the same time, human drivers may be better equipped with context understanding to make significant routing decisions or select responses to certain hazards [179].

Humans become mentors and supervisors to AutonoML systems. This eventuality ensures the effectiveness, safety and trustworthiness of such systems [319]. After all, proficient search is not perfect search. We have just noted how a lack of context understanding is limiting, and the discussion around fairness ambiguities in Section 2.4.5 supports this, but there are still technical complications in ML that will likely never fade. For instance, without a complete analytical description of a search space, it is impossible to know if a locally optimal ML solution is near a global optimum. There are many other challenges, and these, including necessary trade-offs, are mentioned in a previous review [148]. Moreover, it may not even be the AutonoML system that is guilty of flawed ML solutions; Section 2.4.4 highlighted that it is easy for human cognitive biases to be internalised by a machine.

The upshot is that responsible usage of AutonoML requires oversight by whichever stakeholders are running an ML application. Beyond just validating outputs, humans also need the ability to review an entire ML workflow to comprehend/justify why specific models and hyperparameters were selected and how predictive outcomes were derived [319]. This careful appraisal reduces the chance of technical flaws arising in the system, e.g. a performance metric that does not work as advertised. It also improves the outcomes of AutonoML along the lines discussed in Section 2.4, e.g. ensuring that system outputs are fair, nondiscriminatory, ethical, legal, and harmless to minority groups in society. In some cases, this will likely involve the correction of data bias to avoid its propagation.

So, how is this different to proposals made for modern-day AutoML? It is simply a matter of degree. The freeing up of technical labour means that more human resources are available for this oversight, improving the verification of outcomes according to various pre-defined criteria, e.g. safety and security. This migration of labour has already been seen to a limited extent with current AutoML tech, where the effective automation of the model-development phase has shifted stakeholders to monitoring/auditing outcomes alongside focussing on problem definition, data collection, and deployment [323]. It stands to reason then that human resources will continue to be redistributed once AutonoML systems become proficient at transforming data, optimising ML models, embedding ML solutions within production environments, and maintaining them. Naturally, the flow-on effects of increased oversight include boosting confidence in the use of AutonoML and promoting further uptake.

All stakeholders become capable of improving AutonoML performance. We again emphasise that full automation in constrained environments means that an AutonoML system can consistently find, deploy and maintain reasonable solutions to arbitrary but well-defined ML problems. It does *not* mean that solutions cannot be refined by including expert knowledge that a meta-learning module may not have encountered yet. So, it is a legitimate role for technicians to continue guiding AutonoML search, especially if they are privy to useful external information. Certainly, empirical results show that, following the inspection of HPO search spaces and ML models available to an AutoML tool, technical users are reasonably effective at identifying hyperparameter ranges in which better ML solutions may exist [313]. Any decent modern AutoML package will support the subsequent exploitation of a refined search space. However, a consequence of full automation is that the line between ML experts and non-experts is now blurred. As mentioned earlier, in terms of designing robust ontologies, a significant portion of this ideal automation can be achieved with good HCI design. Most low-level technical functions are already present in a system; their operations merely need to be organised and bundled into accessible high-level interactions. Thus, if an AutonoML system is automated correctly, it should be far easier for end-users without an ML background to generate solutions and provide feedback via such interactions [5, 322]. For instance, while technicians are capable of wrestling directly with complex HPO subspaces, a search space could be constrained by simply having a non-expert approve/reject a model recommendation. This process is similar to having a single evaluation inform a Bayesian optimiser where it should next sample solution space.

Granted, the AI community is in the midst of an ongoing debate about whether computerised learning should adopt human expertise or strike it out alone. We noted in Section 3.1 that ML performance in certain applications skyrocketed once the limitations of human knowledge/intelligence were eliminated from the associated approaches. Some have even argued that it is general methods leveraging computation, not human expertise, which is responsible for the greatest successes in AI over the past 70 years [283]. However, others argue that all successful AI systems have "required substantial amounts of human ingenuity", even if that intelligence has supported automated algorithms in varied ways, e.g. by engineering specialised network structures and defining particular training regimes [38]. Unsurprisingly, these counterarguments raise many examples showing where ML technology falls far short of human capability. For instance, a modern recognition system in an autonomous driving car was easily fooled that a traffic stop sign with white stickers was a 45 mph speed limit sign. The neural networks involved could not identify nuances that humans easily handle, instead relying on generic patterns, i.e. stop signs are red and speed limit signs are white.

Extrapolating current research progress in AutoML and AI, it is likely that full technical automation will be attained ahead of any proficiency in context understanding; see Section 4. So, in practice, it is likely that stakeholders will continue to benefit from encoding their rich expertise for use by AutonoML systems, e.g. via suggestions of promising model architectures or the refining of constraints. This requirement may even be essential for high-stakes applications to avoid errors based on a lack of prior knowledge. A hint of how improved knowledge insertion may work can be gleaned from research involving an ML application that predicts patient mortality risk in an intensive-care unit [103]. The authors of the study were keen on automatically extracting problem-specific expertise from domain experts and integrating this into ML models via human-computer collaboration. Specifically, a gradient boosting model was used to transform raw input data into a set of simple rules representing sample subpopulations, each identified by some condition being satisfied. Clinicians were instructed to assess the relative mortality risk for each rule-associated subpopulation compared to that of the whole. The differences between human assessments and those made by data-driven ML models were then computed and ranked. Finally, a new model was generated, penalising unreliable rules with high levels of human-computer disagreement. In this way, the ultimate model hopefully contains the best of both worlds, having incorporated human expertise. Notably, the clinicians in this study did not need to understand ML to improve the model, nor did they have to generate data instances to represent their compiled knowledge. One can only presume that an ideal AutonoML system will similarly facilitate the direct inclusion of expert knowledge from those that do not understand ML.

Continuing legal responsibility forces humans to ensure AutonoML systems are transparent. This assertion comes down to the fact that, while the debate around AI and legal liability has recently surged liability has recently surged [152], the full automation discussed in this section is unlikely to trigger any revolutions in jurisprudence. A proficient ML model optimiser/deployer/maintainer is still a machine following specific orders, i.e. it solves a strictly defined ML problem. It does not have the individual agency to formulate a problem of its own. So, while humans may find themselves content to let AutonoML do all the technical work, legally conscious individuals will be forced to maintain stringent oversight. This interaction may include overriding decisions made by AutonoML systems in emergencies. In short, legal responsibility ensures that human involvement cannot be entirely removed [31]. A simple analogue to an AutonoML system, perhaps one day a direct example, is an autopilot system in a commercial flight. Such a mechanism helps manage tedious and repetitive tasks, thus reducing pilot fatigue and operational error during long flights. However, although the manufacturer of such a system may be liable for mechanical failures, no social contract has granted the autopilot responsibility for the entire flight. It is merely a machine that assists a human in solving a problem, i.e. flying, and its assistance is to be discarded when no longer helpful to the actual pilots.

Now, admittedly, jurisprudential issues become murkier if an autonomous system becomes as reliably capable as a human for handling general decision-making tasks. Other than individual agency, where is the distinction? If a self-driving car crashes with the expected frequency of a standard driver or even more rarely, is the manufacturer any more liable than the parents of the driver? Such debates are out of scope for this monograph, let alone this section. An AutonoML system perfected for constrained environments cannot be expected to handle concepts beyond the bounds of a formulated problem any more than an autopilot is expected to manage every malfunction and tropical storm. Indeed, as is presently the case with AutoML [319], human involvement will likely be necessary for even fully automated AutonoML to satisfy all the rigorous requirements of a practical application. The consequence of all this is that legal liability will be just another socioethical factor that prevents society from accepting black-box designs for AutonoML. Human operators will expect transparency to justify their usage as appropriate and legally defensible. This notion entails having access to AutonoML processes, understanding them, and knowing system limitations. It also means knowing to what extent model outcomes can be trusted and in which contexts they can be appropriately applied [179].

AutonoML teaches humans new tricks. Thus far, the implicit focus of developing fully automated AutonoML systems, with the condition of constrained contexts, has been to produce/deploy/maintain a 'good' ML model. However, once the limitations of human heuristics have been set aside, it can become valuable to observe *how* the AutonoML system performs its job. Does it chain ensembles when constructing an ML model? Does it provision computational resources across a cluster in an unbalanced fashion? Does it adapt local elements of an ML solution to blend short-term and long-term memory? All of these are potential insights for how to conduct ML applications better. Of course, some may argue that if the AutonoML system performs so well, then maybe automation bias becomes justified; see Section 2.4.4. There may be no point in humans learning how to do ML better, particularly if the AutonoML system records these insights with a robust meta-learning module. However, as stated before, we do not treat this idealised form of AutonoML as perfect. Stakeholders may identify patterns, trends and correlations that the system misses, feeding them back in for future runs. Likewise, human cognition may be able to distil AutonoML complexities, e.g. important features arising from deep neural networks with arcane connectivity, into conceptually simplified pieces of knowledge. Even disregarding the use of tricks learned from AutonoML to improve AutonoML, such insights may still feed back into improving problem formulation. This outcome is especially useful when AutonoML has no context understanding. For instance, if an AutonoML system consistently applies dimensionality reduction to throw out dataset features prioritised by data-collecting surveys, perhaps valuable knowledge lies in the neglected features; this may inform future directions for context refining and data curation.

Although AutoML is too nascent to suggest how often it will teach humans how to do ML better, there are many analogous examples among automated learning systems and within various fields. Medicine is but one area that is ripe for ML-derived insights, in which new drugs and treatments can be found by searching and matching elements in ways that are entirely unintuitive to human experts [196]. Unfortunately, in many cases, associated learning systems cannot provide human-understandable explanations of how novel solutions have been found. The onus is then on human experts to validate the results, which requires high levels of creativity, problem-solving, cognitive skills, and domain expertise. However, once confirmed, lessons can be further extrapolated into novel discoveries [115], enriching existing knowledge within a specific domain. As mentioned in Section 3.1, the games of Go and Chess are also arenas where automated learning systems have outperformed humans with unconventional insights. For instance, in Chess, AlphaZero [268] has managed to surprise players with its focus on innovative strategies, e.g. h-pawn thrusts [198]. Similarly, Lee Sedol, world champion and a competitor of AlphaGo, reportedly improved their playing style after examining the moves AlphaGo found during their fifth match [196]. Thus, it seems realistic to assume stakeholders will also find valuable insights when exploring the search history of an AutonoML system, e.g. novel modelling approaches, innovative architectures, varied resource provisioning strategies, and ML pipelines that are well suited to particular kinds of problems [319]. The integration of explainability mechanisms discussed in Section 2.4.3 will presumably further enhance this gleaning of insight.

3.3 What Lies Beyond the Constraints

Thus far, we have envisioned a state of AutoML technology where a machine can autonomously solve a tightly defined/constrained ML problem. This scenario is equivalent to fully automating every stage of an ML workflow from data engineering onwards. However, it is unlikely that the endeavour will end there. Many real-world problems are fluid and cannot be neatly constrained.

Alternatively, even a firm singular high-level agenda may translate into multiple ML tasks. For instance, an ML application dedicated to optimising an investment portfolio may aim to predict the best neighbourhoods for property purchases one day and forecast energy market trends the next. In both cases, one could try to set up multiple ML applications for sub-problems or problem variants across different contexts. However, not only is this arrangement highly redundant, cross-context information becomes difficult to leverage. For example, active learning may identify that a client is only interested in 'green' suburbs, producing knowledge that would influence better recommendations for energy-company investments. Thus, with respect to ML problems, the grand aim is clear: one AutonoML system to process them all.

Going beyond constrained environments is, of course, no easy feat. Automated systems can struggle within problem contexts possessing weak structure, consistency, guidance, and rules. In such semi-structured and unstructured environments [158], rare/undesirable events and unforeseen changes in goals may substantially distort, one, a function that ML is attempting to approximate or, two, the space of data that samples this function. Importantly, we emphasise that the changes must be dramatic and related to context; simpler forms of data/concept drift are presumably handled by the adaptation mechanisms of a fully automated AutonoML system. In essence, challenges only arise when knowledge acquired via proficient search/learning on supplied datasets cannot address emerging situations. That is not to say that all of these challenges require complex countermeasures in modern ML. For example, if a visual model distinguishing between fish and coral starts to encounter something challenging to classify, the simplest method is to attract the attention of humans and have them introduce a new label, e.g. oceanic plastics. Unsurprisingly, many mechanisms have been proposed to automate such a process, covering novel data detection [138], automated labelling [247], and the growth of models to accommodate unexpected forms of new data [95, 151]. However, no mechanism currently exists for managing the entire spectrum of situations in semi-structured or unstructured environments where a domain and inputs are inconsistent over time. Even ML adaptation in structured environments is currently onerous, with humans often involved in reviewing performance degradation, assessing the effectiveness of automated adaptation mechanisms, and confirming a model update if necessary. So, adapting to changes in unstructured environments can only be more labour-intensive, time-consuming, and costly.

Herein lies a fundamental weakness of traditional ML. Inductive learning requires a *lot* of data with repeated presentation of concepts in order to develop a decent model. The era of big data has undoubtedly made this feasible for many ML applications, but the process can still be intensely time-consuming and energy-intensive, even on modern computational hardware. Moreover, it can be tough to collect sufficient data for many other ML applications, even with synthetic support by a generative model, and this challenge amplifies for environments where concepts can be fluid, e.g. those that are complex, messy, semi-structured or unstructured. This state of affairs is unfortunate as, given sufficient computing resources and data, automated search and learning can easily outperform models constructed with expert knowledge [290]. This sentiment recurs in the literature, and the superiority of automated search/learning over expert knowledge-based systems is exemplified by research into adaptive soft sensors for well-defined industry process problems [139, 143, 144, 248]. At the same time, the immense costs are also well-evident, particularly when considering state-of-the-art DL approaches used for non-tabular input data, e.g. images, sounds and text. Deep neural networks have proved greatly impressive by unlocking new applications and pushing the known limits of predictive accuracy. However, they can be substantially less computationally efficient than knowledge-based learning systems; a training time of weeks and even months is not unusual. Consider the celebrated NLP model, GPT-3, which cost about $4 million to train [290]. Although a flaw was retrospectively identified in the preparation of training/testing data, i.e. concerning overlaps, the developers decided against retraining the model due to its high cost [40]. This hindrance to fundamental progress in ML can be found across the board. In medicine, an AutoML algorithm supporting the construction of an automated diagnostic system needs to search through as many variables as possible to find out which features are important, requiring a large amount of data and inducing high computational costs. In contrast, models trained with expert insight, e.g. with radiologists or oncologists identifying essential features based on their experience, can be fashioned much faster and, if the domain knowledge is accurate, reach a high level of performance very efficiently [290].

As of the early 2020s, the debate around the future of ML and broader AI continues. The pivotal successes of DL have seemingly become less frequent and more incremental. Without any revolutionary acceleration of computational hardware on the visible horizon, e.g. photonic or quantum computing, some argue that data-driven learning should progress towards knowledge-based learning. However, there are also good reasons why the term 'expert system' faded from the data-science lexicon in the 1990s. There are numerous examples where expert insights failed to deliver better solutions than data-driven learning, e.g. in toxicity prediction [41] and biomarker-based water-pollution prediction [42]. The human cognitive biases discussed in Section 2.4.4 can also have much more negative impacts on knowledge-based systems over data-driven learning, e.g. via the misidentification of relevant variables and inclusion of faulty rules. Thus, modern proponents of knowledge-based learning rarely hark back to the extreme of traditional expert systems; the suggestion is that the best of both worlds should be fused. For instance, perhaps a mechanism based on human cognition/reasoning may be what enables the automation of context understanding, supporting the optimisation of ML pipelines for fluid ML problems with a flexible leveraging of domain expertise. Indeed, maybe this is the key to strengthening the adaptability of an AutonoML system and enhancing its general applicability. Whatever the case, whether by a purist form of induction or by a fusion with knowledge-based approaches, AutonoML research will eventually grapple with learning in open-world environments. The question arises: will this force HCI to evolve even further?

4 INTERACTING WITH AUTOML SYSTEMS: OPEN-ENDED ENVIRONMENTS

Research along traditional lines of ML seems all but destined to advance AutoML until the technology can solve ML problems to a high degree of performance without human involvement. This was the focus of Section 3. However, it is much less clear how to take AutonoML further. Data-driven AutoML systems that excel at searching in constrained environments are unlikely to handle moving between dramatically different domains and tasks. Extrapolating present-day approaches, these systems will struggle to extract learned knowledge from one task and apply it to another. In contrast, humans are very proficient at acquiring new knowledge from minimal data, abstracting and generalising with typical ease. A major long-term challenge then is translating capabilities typical of human reasoning to automated learning systems [179], e.g. generalisation, an understanding of normativity, the detection/use of causal factors, and so on. Success in this endeavour may bring AutonoML more in line with a genuine assistant than a tool, and HCI may start to be framed more in terms of collaboration over mere operation.

Crucially, as displayed in Table 3, this advance in AutonoML comes primarily down to one section of the ML workflow. More specifically, leaving problem formulation in the domain of humans, we consider the challenge of automating the second sub-phase: context understanding. This aim is a lofty goal, but plenty of research/debate exists around this topic in the broader literature. We merely recontextualise these discussions within the field of AutoML. Progress in this area would allow AutonoML to understand problems better, consider them from different perspectives, and require far fewer human interventions in complex open-ended environments. Systems would be particularly suited to managing clusters of similar tasks, but they would also be capable of adapting to novel

Table 3. The prospective evolution of automation in ML.

Stage in ML workflow	Need for human involvement		
	AutoML: Current practices (Section 2)	AutoML: Constrained but fully automated (Section 3)	AutoML: Open-ended environments (Section 4)
Problem formulation & context understanding	Yes	Yes	Partly
Data engineering	Yes	No	No
Model development	Yes	No	No
Deployment	Yes	No	No
Monitoring and maintenance	Yes	No	No

unseen contexts, transferring knowledge where relevant. As an example, an AutonoML system for a lane-keeping driving task might generalise its learned knowledge and adapt to various encounters beyond what it was trained for [167], e.g. in terms of road profiles, terrain, and lighting conditions. Of course, this theoretical discussion will remain carefully grounded for AutoML practitioners and current perspectives, striving to suggest what forms of HCI to prepare for when designing the next generation of AutoML. We will avoid delving deeper into general-purpose learning systems, which verges on artificial general intelligence (AGI).

So, to address the topic of AutoML and open-world learning, Section 4.1 will first explore what makes open-ended environments so challenging for ML. A possible pathway to improving AutonoML will be discussed in Section 4.2: the incorporation of human-style reasoning. Current thoughts on the plausibility of this future will be reviewed in Section 4.3. We will then assume the advance to automated context understanding is inevitable, proceeding in Section 4.4 to consider how HCI between stakeholders and AutonoML systems is likely to evolve. Finally, Section 4.5 will examine what scope there is to optimise interactions for this novel state of technology.

4.1 The Challenges of Learning in an Open World

Most popular ML algorithms in the present day operate with a closed-world assumption, where models are generated from a given training set without considering historically acquired knowledge or any other contextual insights [55]. Therefore, ML algorithms need a large amount of training data to effectively learn and optimise predictive/prescriptive performance. Frequently, for supervised learning, the training data is considered a representative sample of what will be encountered during testing and production. Thus each testing/production sample is typically only classified into classes learned from training sets [85]. Admittedly, some methods may be sophisticated enough to construct new classes for poorly classifiable data. However, the real world is open-ended, where no data perfectly reflects all the dynamic changes and events that may occur within a data environment.

In effect, there are unlimited possibilities and no fixed rules to anticipate all requisite information needed to maintain predictive/prescriptive performance in any given situation. Accordingly, it is virtually impossible to yield a perfect training set for every function that an ML model needs to learn. The challenge then is deciding how best to loosen the constraints of well-defined problems so that AutonoML systems can work effectively in real-world scenarios with the aim of continuous lifelong learning. Operating conditions and contexts usually change over time within an open-ended world, so the key to progress seems to rely on improving the first phase of the ML workflow: problem formulation and context understanding. Ideally, the goal is to equip AutonoML systems with advanced mechanisms capable of understanding operating contexts better, enabling more autonomous operations supported by non-technical collaboration with human operators. This gradual relaxation of closed-world constraints is a common agenda across the broader field of AI. Many ongoing data-driven research efforts aim to acquire, generalise and transfer learned knowledge across different input spaces of the same task or related tasks within open-world scenarios.

One thread of research considers the situation where a single set of training data does not cover every angle of an ML problem. Specifically, multi-view learning samples a data environment from multiple perspectives, in which each view is represented by a distinct set of features, thus enhancing the performance of an ML model. For instance, training data for an ML problem processing a video source can be formed from two different perspectives, i.e. the image signal and the audio signal. Both views are non-overlapping, yet the signals will be correlated in a highly complex manner, and both will need analysis to thoroughly understand the meaning within the data source. This research direction has many motivations, but one of them is enabling an automated system to examine and leverage every angle of provided data without changing the problem definition. Numerous approaches for multi-view learning are listed elsewhere [281].

Another thread of research investigates continuously changing environments, although still mainly within the confines of a previously defined ML task. The goal is to enable an AutonoML system interacting with its environment to learn from its experiences, thus accumulating, organising and transferring knowledge over long periods. This long-term form of continuous learning, accommodating new knowledge while retaining previously learned insights, is also known as lifelong learning. Many approaches tackling lifelong learning draw inspiration from biological systems, and they have been reviewed elsewhere [220]. We re-emphasise that, to a large degree, lifelong learning is already expected to be a characteristic of constrained but fully automated AutonoML; see Section 3. It is the *extension* of adaptive mechanisms to the phase of context understanding – consider the dotted looping arrow from the monitoring and maintenance stage in Fig. 1 – that defines the lifelong learning required for open-ended environments.

As Section 3.1 indicated, an ideal approach to information transfer between contexts is the use of meta-knowledge, where knowledge about how learning was achieved for one ML problem should prove educational in enhancing the performance of ML within a similar setting/domain. This process of meta-learning, along with its achievements and applications, has frequently been reviewed [4, 172, 304]. In the context of AutoML, meta-learning can be used to recommend good values for system variables, ranging from low-level HPO search-space constraints [208] to high-level mechanisms, e.g. which optimiser to use. A closely related notion is transfer learning [3, 218, 333], which typically involves copying and pasting an entire ML pipeline or DL network with a solid performance on one ML/DL problem into the solution space of another. Most of the transferred pipeline/network is often kept invariant to subsequent retraining, with only the interfacing components/layers being tunable. In theory, this approach is reasonable, as, for example, it seems that initial layers for feature identification/processing in a deep neural network should not change dramatically whether a model is identifying animals or objects. Presumably, the last layers correlating feature combinations with concepts are what vary. However, both meta-learning and transfer learning suffer the same issue. Their effectiveness depends on the number of commonalities between domains. For human learning, this makes intuitive sense; the skills acquired from swimming do not readily apply to playing a guitar. The challenge lies here: the degree of similarity between problems/contexts remains massively difficult to define robustly. Thus, a softer form of transfer learning exists, called domain adaptation [282], which typically does not have to care about varying feature spaces, only distribution/domain shifts. Domain adaptation is not open-world learning in the way this section defines it, but it is still an incremental loosening of problem constraints, the successes of which may feed into broader progress.

By now, it is evident that there is plenty of research activity in broadening the adaptive applicability of ML and, by proxy, AutoML. However, these threads continue to struggle with training/adapting ML models for real open-world problems despite many remarkable achievements. In a recent IEEE Spectrum interview, distinguished ML researcher Michael Jordan stated that, "for the foreseeable future, computers will not be able to match humans in their ability to reason abstractly about real-world situations" [229]. Unlike ML algorithms, humans can learn effectively and adaptively in open-ended environments from only a few presentations of concept [55]. Naturally, there is no consensus as to how this is done, e.g. whether this capability is unlocked by the processing power of the brain, the advanced mechanics/architectures of biological neurons and their networks, or simply the accumulation and effective leveraging of knowledge over a lifetime of learning. Perhaps it is a mixture of all three. Nonetheless, phenomenologically, these abilities appear to manifest as the use of semantic representations and high-level reasoning. Thus, if ML/DL algorithms do not have the complexity yet for such high-level phenomena to arise emergently, then perhaps the shortcut of engineering these qualities into an AutonoML system will make do. Such a mechanism would make AutonoML better suited for open-ended environments by enabling the accumulation and organisation of historically learned knowledge, employing it via the form of reasoning on new ML tasks/problems [16].

The core requirement of open-world learning systems is the ability to identify and learn unknowns in order to become more knowledgeable. In particular, AutonoML systems may need to construct models of the world to support prediction and planning, explanation, abstraction, and the flexible generalisation of knowledge from observations [165]. This capability goes beyond simple pattern recognition based on many previously determined tasks. For instance, toddlers often observe the world and generate a mental model regarding how it works, then they take action and adopt the outcomes of that action to refine their mental model [278]. These steps are repeated until they sufficiently interpret the world. Indeed, through continuous training on non-stop streams of data encountered every day, supported by ongoing background optimisation during sleep time, toddlers accumulate rich knowledge to help them better perceive and understand various contexts in the real world.

Immediately, this hints that the very nature of HCI within AutonoML might need to change. Although Section 3 suggested meta-knowledge bases could be required for closed-world AutonoML, these would likely be developed/expanded *reactively* while a system is encountering/solving a problem, with maybe a few one-off sessions of offline processing. On the other hand, a model of real-world contexts and their relationships is at least an order of magnitude more complicated to generate. In essence, AutonoML systems may need to *proactively* develop these context models, in which case they are unlikely ever to be switched off. This paradigm is vastly different from what Section 2 describes as the most common present use of AutoML, i.e. a tool to be picked up when an ML application requires it. In fact, given how complex and expensive it will likely be to generate a context model for the real world, it may be that one will be shared between multiple instances of an AutonoML system, with each contributing to it in their off-time. Accordingly, the systems are likely to be 'always on', which has implications for the *constancy* of HCI; see Section 4.4.

Now, it is argued that one of the strengths of human-level intelligence that suits it for open-world learning is the ability to generate information from raw data and systematically organise it into abstract high-level concepts [99]. This knowledge synthesis occurs at a different level to the typical curve-fitting that ML applies to raw data. Substantial portions of the data science community advocate this reasoning-based form of learning as a way forward for more capable AI; this includes AutoML. Additionally, in an open-ended environment, AutonoML must be equipped to make decisions with incomplete and uncertain information. Such scenarios have previously been called commonsense informatic situations [191]. Thus, although meta-learning and transfer learning based purely on data-driven approaches have shown promise for open-ended problems, the challenges of learning in an open world seem to require more sophisticated strategies leaning towards human-based reasoning. We cannot state with certainty whether this is the best pathway for advancing AutoML, but the intensity of recent research activity aimed at integrating high-level reasoning mechanisms into data-driven learning paradigms warrants review, if only to characterise/contextualise the advanced systems that HCI may have to accommodate.

4.2 One Possible Future: AutoML and Reasoning

The path to automating context understanding and relaxing the restrictions on open-world learning may require AutoML to eventually integrate some form of mechanism for reasoning. Three components of human intelligence are particularly relevant to this endeavour: (1) the synthesis of conceptual knowledge from data, (2) reasoning based on this conceptual knowledge, and (3) cognitive mechanisms that revise and adjust the acquired conceptual knowledge. So, to review research efforts/opinions in this area, Section 4.2.1 begins by examining how humans reason and make decisions. As detailed by Section 4.2.2, this potentially motivates a shift in how ML is approached, evolving from a data-driven focus to one that is knowledge-driven. Various types of reasoning will then be elaborated by Section 4.2.3, before Section 4.2.4 concludes with a brief overview of how artificial models of cognition may be generated.

4.2.1 How Humans Make Decisions

Reasoning is a complex concept, and neuroscientists continue to debate many facets of how the human brain engages in cognition. Accordingly, this section is not intended to be a comprehensive review of current literature on the topic. However, there is a rough consensus that human cognition operates on multiple levels, although the jury is out on whether these levels are discrete modes of operation or merely distinct phenomena along a continuous spectrum of neural processing. For the purpose of contemplating open-world AutonoML, the low-level details do not appear to matter. Thus, we focus here on the popular dual-process model of cognition, also known as the Kahneman theory of thinking fast and slow [145]. According to this theory, human reasoning and decision-making are enacted by collaboration between two 'systems' of thinking. System 1 involves thinking that is fast, instinctual, associative, intuitive, and low-effort. System 2 involves thinking that is slow, deliberate, logical, rational, and reasoned. Several discussions of AI have been grounded in this theory [241].

In greater detail, System 1 is said to involve humans using experience, biases and evolutionary heuristics with unconscious cognitive shortcuts to generate quick responses [272]. Within the context of AI, this form of thought is often associated with the pattern identification and curve fitting of standard ML. It is not that *generating* the approximating model is fast; the speed usually refers to ML inference, where a novel instance of data can be quickly converted into a prediction/prescription by the approximating function. A human example of System 1 processing is rapid object identification, usually employing inductive reasoning and the representativeness heuristic to leverage similarity with previously seen objects [312]. Naturally, the priority of speed over robustness for System 1 means that determinations tend to have weaker validity over other forms of deductive reasoning. The results of System 1 can also lack explainability, being based on heuristics, unconscious biases, memories, pattern matching, and other shortcuts. Pareidolia is an obvious example of errors in associative thought. Nonetheless, despite a tendency to be loose with accuracy and precision, System 1 remains essential for efficiently updating and maintaining personal models of the world [145], constantly integrating a stream of incoming information into background beliefs [273] so that effective responses can be formulated for many daily stimuli. Admittedly, unlike most forms of traditional ML, it is argued that System 1 in humans is also capable of conducting basic causal inferences to interpolate/extrapolate in the case of incomplete knowledge [33]. This causal reasoning applied by System 1 thinking is kept at the level of data, while more complex assessments of causality are thought to occur at higher levels.

In Kahneman theory, this leads to System 2. Theoretically, when facing a novel problem that is too complex to be solved by System 1, such as complicated mathematical calculations, System 2 is triggered and takes over. It accesses extra cognitive resources and sophisticated logical reasoning mechanisms to form thoughts in a series of steps [145]. During this procedure of extending explicit knowledge and beliefs, System 2 thinking is slow, step-wise, deliberate, analytical, rule-based, controlled, and resource-heavy. In the case of humans, complex reasoning processes cannot be utilised without prior knowledge acquisition from many different domains. This expertise is required to systematically capture/use concepts that are semantic, logical, or mathematical [312]. It is also needed to manipulate symbolic and abstract rules [111]. Because inference here can be much more involved than with inductive approaches, System 2 is argued to subdivide large tasks into multiple smaller parts. High-level cognition also appears to prioritise long-term memory, while instinctive low-level cognition typically leverages short-term memory [145].

As with humans, any AutonoML frameworks that incorporate the principles of Kahneman theory are expected to have distinct levels of cognition acting in collaboration. Proposals in the field of AI vary, but System 2 is often envisaged as being supported by the fast-thinking System 1. For instance, given the vast solution spaces that System 2 can encounter, heuristics provided by System 1 are well-placed to provide constraints and focus only on the most promising subspaces [33]. In the opposite direction, once complex concepts are sufficiently established and refined by System 2, some of this knowledge may be translated/simplified into a form that System 1 can use. This development is often associated with the notion of once-complex tasks becoming ‘second nature’, i.e. no longer requiring concentrated thought. Examples of this are many, e.g. developing motor control for bike riding or learning to read a natural language.

The overall hypothesis here is that it takes sophisticated thinking/reasoning mechanisms, such as the collaboration of two distinct levels, to provide humans with a competitive advantage in cognition and decision-making over, for instance, animals. Indeed, animals may have a significantly less developed form of System 2 thinking, aligning with a predisposition for instinctive behaviour. In contrast, humans can reason at multiple levels of abstraction, adapt efficiently to dramatically new environments, understand wide-ranging contexts, translate/generalise knowledge-driven skills from one problem to another, etc. Accordingly, to enhance the capabilities of AutonoML, it appears desirable to engineer frameworks with these multi-level principles of cognition. Of course, computer architectures are not identical to biological brains, so future AutonoML engineers may have the flexibility to improve upon the designs of nature. For instance, in humans, Systems 1 and 2 are often categorised by local and global forms of processing, respectively, i.e. in terms of which parts of a brain are involved. A natural question to ask is as follows: would an integrated AutonoML framework [148] mimic these distinctions? Additionally, it is suggested that the cheap form of System 1 processing is done in parallel within a human brain, while System 2 requires plenty of attention/concentration, operating serially. Sure enough, conventional wisdom is that few humans are effective multi-taskers when it comes to deliberative thought. Nevertheless, perhaps an AutonoML framework does not need to be similarly restricted, and computers may support the parallelisation of high-level cognition.

Importantly, we note that, while multiple levels of cognition do advance human capabilities, cognitive biases at all levels can impact solutions, forcing sub-optimal outcomes and incorrect decisions [312]. Typical heuristic biases and their negative impacts have previously been highlighted when studying the medical decision-making process of clinicians [228]. Granted, as discussed in Section 2.4.6, these same factors may be mitigated in autonomous systems by the integration of explanatory mechanisms and collaborative interactions with stakeholders. Nonetheless, it is probably a mistake for AutonoML to directly mimic how humans make decisions. More realistically, biological processes should be studied as an inspiration when designing the behaviour of AutonoML systems in the future.

4.2.2 The Paradigm Shift from Data-driven to Knowledge-driven

As hinted by Section 4.1, the reason why humans are so effective at open-world learning may partially be due to one fact: humans virtually never start learning from scratch. This statement may seem patently untrue, as humans frequently encounter new situations for the first time. However, unlike a modern AutoML system, they are hopefully not rebooted prior to each new encounter. The above assertion does not even refer to the persistence of a traditional meta-learning module in the human brain, e.g. one that aids learning how to ride a mountain bike based on prior experience with a road bike. No, long-term experience has equipped humans with much more abstract general knowledge [164], e.g. numbers, time, space, physics, psychology, etc. These concepts represent pervasive principles inherent to real-world environments, e.g. the mental/behavioural states of entities, the observed physical laws of reality, and even more abstract properties like compositionality and causality [34]. Such abstractions form links between disparate environments, meaning that even an apparently novel problem context can be, to some degree, understood by a human. For instance, most humans recognise that jumping off a skyscraper without a parachute is typically a bad idea. They usually do not have to experience/observe repeat presentations of this concept to reach enlightenment. Thus, the integration of fundamental abstract concepts into ML systems, such as notions from physics and psychology, may be essential for elevating AutonoML to open-world learning [164]. To date, debate continues around how best to achieve this, but a notable portion of the data science community fears that the modern standard of approaching every ML application from scratch and relying only on data-driven approaches is likely to be “an alarming mistake” [187].

A recent study argues that most modern forms of ML, especially DL, are poor at generalising knowledge beyond the space of training data, meaning that they cannot be trusted in open-ended environments [186]. Consequently, the study motivates the development of an autonomous system that can regularly acquire, represent, and modify abstract knowledge. The goal is for the system to use this learned knowledge in deducing, constructing and updating internal cognitive models of an external working environment. To be fair, many present-day ML/DL algorithms effectively generate knowledge within a constrained environment. However, they fail to capture many aspects, e.g. the way that the physical world works, how objects interact, and how causal relationships manifest in dynamically changing environments [330]. Several research efforts have honed in on this last point specifically, attempting to statistically extract causal relations from changes in data distributions [23] and automatically learn modular structures representing the dynamics of a working environment [108]. The concept of causality is essential to human reasoning, often impacting the way new concepts are generated and learned [236]. In addition, causal knowledge also enables human beings to specify what is critical and infer what is unknown [230]. For this reason, equipping AutonoML systems with mechanisms for the extraction of causal relations from data may be an initial step towards open-world learning.

Now, assuming that the AutonoML agenda does begin to extend from data-driven approaches to knowledge-driven learning, a fundamental question is as follows: how should abstract concepts at a higher level than data be represented? There is an obvious answer: with symbols. Indeed, animal cognition is argued to use symbolic representation for the storage and retrieval of information [97]. This theory is particularly well-supported for humans, with plenty to suggest that symbolic reasoning is vital to deliberative cognition [186]; consider how often the communication of thought processes is framed in deductive logic operating on symbolic concepts. Therefore, equipping AutonoML systems with a deeper understanding of an open-ended working environment seems likely to rely on integrating human-like mechanisms that acquire, represent, organise and manipulate abstract symbolic knowledge. In effect, open-world AutonoML may need to embrace a hybrid learning paradigm that can discover, extract and generalise abstract knowledge from large and noisy datasets while also reasoning with this learned knowledge to better tackle new ML tasks. It is certainly a recurring opinion, in both the past and present, that AI architectures could potentially benefit from the best of both worlds [189, 206], i.e. the powerful large-scale learning ability of data-driven methods and the generalisable applicability of abstract knowledge in symbolic format.

Another potential strength of knowledge-driven learning is an improved ability to handle uncertainty and missing values. Crucially, in real-world contexts, data is usually dependent on imperfect observation, thus being subject to noise and incompleteness. Admittedly, the typical point of an ML algorithm is already to fit and interpolate/extrapolate, leveraging model-specific choices for how inductive reasoning is applied. However, in an open-ended environment, it is much easier to encounter a novel observation that has no comparable instance of data on which to anchor inductive reasoning. Thus, an ideal open-world AutonoML system should lean on deductive reasoning to fill in the gaps, making assumptions, inferences and deliberation based on prior conceptual knowledge, updating and revising where need be. Relying on pure search, no matter how proficient or supported by association-based meta-learning, is unlikely to be efficient or effective.

As an example, consider an ML application tasked with predicting the socioeconomic classes of people – each dataset is specific to a nationality – based on their yearly expenditures across a range of consumer goods. Searching through an infinite space of possible data transformations to maximise classification performance seems infeasible. Selecting data transformations based on similarities between country-specific datasets may be better, but it faces the usual challenges of traditional meta-learning described in Section 4.1. In contrast, linking a country to a knowledge-base of economic data may present a more expansive yet well-reasoned pool of data transformations, e.g. normalising expenditures based on purchasing power parity and domestic price indices. Now, consider that a new dataset differs not in nationality but feature space, e.g. income streams are listed instead of spending. Traditional AutoML would likely need to search for an ML pipeline from scratch, while meta-learning would have no prior experience to recommend data transformations. However, conceptual knowledge, such as linking income/expenditure feature labels to appropriate economic data, may preserve the performance of the original ML pipeline; an AutonoML system may simply attach a new transformation related to country-specific household debt levels.

All of these knowledge-driven improvements boil down to relaxing the need for HCI in the first stage, second sub-phase, of the ML workflow in Fig. 1. By working towards automated context understanding, humans no longer need to be as rigid and constrained when defining a problem statement. Concurrently, AutonoML systems and the ML solutions they support become much more robust and capable. For instance, it is worth noting that, although the famed GPT-3 language model [40] achieves impressive outcomes on NLP tasks, many limitations have been attributed to it, e.g. ingrained biases, a propensity for cherry-picking, a lack of semantic coherence in outputs, and unreliability [62]. It is far from a system with an open-world level of intelligence because it is primarily based on textual statistics and probabilistic matching. It has no discernible understanding of underlying concepts, causal relationships, and other facets with which human cognition could easily grapple. We stress the word 'discernible' here, given how contentious the phenomenon of understanding is; consensus theories of human cognition attribute both the fast and slow thinking discussed in Section 4.2.1 to the same fundamental concept of a biological neuron. Nonetheless, many argue that appreciating the statistical properties of a concept manifesting is not equivalent to understanding that concept [186]. To acquire true understanding, or at least mimic the functional phenomenon, AutonoML seemingly needs to operate with prior abstractions of concept under uncertain conditions. Naturally, if deductive inference is based on previous truths, a question then arises: what kinds of prior knowledge should be made available to an autonomous learning system? Data scientists have argued that these should encode principles about time, space, causality [187], compositionality, intuitive physics, and psychology relating to both individual agents and groups [164]. Whatever the case, this 'mental model' of the universe will need to be curated/integrated carefully, given the significant potential for computational bloat.

A final comment is that one of the big HCI issues of modern AutoML, i.e. the topic of trust that was discussed in Section 2.4.1, will remain relevant for knowledge-driven learning. Outcomes need to be interpretable/explainable, especially when the risk-managing constraints of a closed world fall away; see Section 4.4. Fortunately, symbolic logic and similar forms of reasoning are already typically perceived as more transparent and believable than inductive extrapolation due to various potential properties, e.g. logical consequence and deductive closures. Indeed, there are many human-centred reasons why knowledge extraction has been proposed to enhance communication between users and an autonomous learning system [99]. Improved explainability is just another potential bonus of integrating knowledge-driven approaches into AutonoML systems.

4.2.3 Types of Reasoning

Even though the implementation of an open-world AutonoML system is unclear, in terms of whether mechanisms should be discrete and prescribed [148], a functional dichotomy is becoming apparent. Traditional ML is associated with System 1, fast inference, inductive principles, and data-driven learning. Automated context understanding is associated with System 2, slow inference, deductive principles, and knowledge-driven learning. Traditional ML typically attempts to interpolate/extrapolate approximate probability density functions for an entire domain, processing large amounts of data. Automated context understanding prospectively attempts to encode concepts from a problem setting and data environment into formal logic, often compacted representationally into symbolic logic. Each has its own particular strengths and weaknesses.

For an open-ended world, it is infeasible to make a nuanced global fit of a function, so the capabilities of ML are weakened. Accordingly, an AutonoML system should potentially support local-space approximations with inferences formed from global-space abstractions of knowledge, thus extrapolating beyond supplied data distributions without needing large training samples [293]. Many other inspirations can be drawn from human intelligence, which is effective/efficient for open-world learning. Humans frequently reason when considering action strategies, taking into account many relevant actors and events, both transient and continuous [191]. Of course, the dichotomy discussed above is a simplification, and there are many nuances to reasoning that may be useful in the context of open-world AutonoML. We briefly mention reasoning of the following types: low-granularity induction, causal deduction, symbolic, common-sense, and 'exact' combinatorial.

When discussing induction, as in making broad generalisations from specific observations, it is essential to emphasise that knowledge exists in a spectrum of scope and detail. Unlike other animals, humans have the known ability to understand and communicate high-level information easily [117]. This quality is primarily due to the capability of abstracting high-level concepts from raw data. In addition, human intelligence can make inferences at different levels of abstraction to focus only on specific sets of features, disregarding unimportant information [335]. Now, while the last few sections have stressed the benefit of using deductive reasoning for conceptual knowledge, in apposition to the inductive reasoning applied by ML at the data level, nothing precludes induction from being applied to the higher level of knowledge as well. Plenty of research exists intending to extend learning algorithms to various granularity levels, e.g. as proposed by a granular computing framework [18, 150], generalising abstract/causal rules via the process of induction.

On that note, causality is a beneficial concept to have a handle on when forging predictions/prescriptions. A weakness of traditional ML is that it constructs models inductively based on correlation, which means that ML solutions can fail dramatically under data/concept drift if a feature-label association is coincidental and the underlying causal factor remains unknown. In contrast, once a causal relationship is determined, it can usually be converted easily into a logical consequence, allowing subsequent inferences to be deductively made. Of course, algorithms still need to extract rules/knowledge from raw data to reason at a higher level with hierarchical structures of causal insights. One way to do so is with probabilistic programs. For instance, a recent study used a Bayesian strategy to convert raw data into causal knowledge, handling noise with probabilistic semantics [163].

Once an AutonoML system begins to dabble in high-level deduction, this reasoning will usually be expressed in terms of formal logic. In fact, at a low level, decision trees are an explicit example of formal logic being applied, typically represented in the form of ‘if-then’ rules. Here, we remind that, while ML constructs models inductively, applying the model to new queries is technically a deductive process, i.e. ML training converts specific information to general information, while ML inference works oppositely. Regardless, combining and aggregating logical theorems at a higher level can result in highly complex structures. Thus, humans usually employ symbols to encode such concepts for both convenient communication and the atomic basis of extended reasoning [286]. Unsurprisingly, many studies have focussed on embedding symbolic rules and knowledge into traditionally data-driven ML/DL algorithms so that they may infer and capture compositional/causal structures from data [99]. One proposal considers adopting a three-level causal hierarchy to make ML algorithms more robust and autonomous [222], i.e. managing association, intervention, and counterfactuals. As mentioned above, most current implementations of ML only leverage purely statistical relationships, so they only support the lowest level of association. However, it is argued that if symbolic rules can be integrated into models such as neural networks, ML will become capable of supporting inference involving intervention and counterfactuals [99]. In essence, applying symbolic manipulations seems to be very convenient for, if not a prerequisite to, understanding cause-effect connections. Naturally, given a dichotomous perspective of fast and slow thinking, one possible approach is to allow data-driven learning algorithms to interact with symbolic extraction/reasoning mechanisms in some hybrid manner, e.g. a neural-symbolic framework [25].

Finally, the topic and lexicon of cognition are vast, so many other variant terms are associated with reasoning, each with their own nuanced vision of what ML should support, even though aspects overlap with the concepts already discussed. For instance, any artificial system competitive with human intelligence is arguably required to operate efficiently in common-sense informatic situations [191]. This notion usually refers to decision-making when information is uncertain or incomplete. The intention here is that AutonoML systems will need the ability to jump to conclusions based on assumptions, using approximate concepts/theories and, ideally, maintaining a record of the confidence levels for each decision. Such conclusions may then be updated and adjusted during learning processes when new instances are observed or novel knowledge is extracted [99]. Practical principles to support common-sense reasoning include Bayesian inference, non-monotonic logic, and fuzzy logic. Oppositely, when an ML application is rich with knowledge and observations are both complete and certain, ‘exact’ combinatorial reasoning, i.e. based on counting, may find its own purpose. Given that combinatorial reasoning could potentially correct conclusions generated by approximate reasoning, it is clear that an AutonoML system would benefit from implementing various forms of logic, working in concert to cover a broad spectrum of possible ML problems/contexts. Admittedly, high-level AutoML operations are already very resource-expensive in the modern era, so the true challenge of employing cognitive mechanisms may not reveal itself until the engineering/implementation phase.

4.2.4 *Cognitive Models of Learning*

Given that the notion of incorporating high-level cognition in AutonoML is presently speculative, although well reasoned, we do not dwell on matters of implementation and engineering. However, it is worth noting that, in the broader AI literature, plenty of relevant thought is being paid to artificial cognitive models of learning and how they should function. For instance, one recurring question is how such systems can simultaneously enrich and leverage high-level knowledge. Human cognition can again act as an inspiration for this process, which accumulates precise/complete knowledge through continual operation. Specifically, it obtains, encodes and revises knowledge based on experience or instruction, inferring and deducing new knowledge by the combination of existing concepts and rules [166]. One example of a computational system that simulates this mental procedure is Cascade [303], using rules to represent knowledge and incorporating an impasse-repair-reflect cycle. During the learning process, Cascade operates normally until it hits an impasse, i.e. where existing knowledge is insufficient to solve a problem. This event then triggers a repair phase, where current rules are modified or new ones are constructed from the combination of previous rules, with common-sense knowledge potentially involved. Finally, the system reflects on the validity of the suggested rules via a self-explanation form of reasoning.

Beyond operating procedure, AutonoML engineers will also need to decide what an open-world system will need to be shipped with. Indeed, while Section 4.2.3 discussed diverse forms of reasoning with knowledge, earlier sections highlighted arguments that a cognitive model should already be equipped with baseline information about sets of entities and relevant properties, including time and events [186]. Such proposals suggest that it should not be overly challenging to construct internal cognitive models automatically, based on foundations that include hybrid model architectures, prior knowledge, and reasoning techniques. Finding effective algorithms to surmise causal relationships from data may also be particularly valuable, providing strong priors for learning/reasoning as well as the generalisation of new knowledge [164]. For the time being, most ML/AutoML research and development have overlooked these matters, focussing on what is perhaps the easier challenge: end-to-end learning with convenient assumptions, e.g. the availability of large datasets. This deliberate neglect of building cognitive models within learning systems may stymy progress in the long term. For example, NLP models based on Transformer architectures [305] can achieve state-of-the-art results, but their reasoning ability is poor. Without cognitive mechanisms and a conceptual model of the world in which the language is based, such applications lack interpretability and genuine understanding. The text generation capability of such NLP systems, while impressive, quickly ends up being identified as human mimicry; the lack of context understanding means such approaches will easily struggle with the Turing test. Thus, it seems inevitable that AutonoML research will have to tread down the path of integrating cognition mechanisms, or equivalent, if it intends to unlock the full power of automation on a maximal spectrum of ML applications.

4.3 A Brief Debate on the Plausibility of This Future

Any speculation on the future of ML will always be debated, so, before we consider the full implications of open-world AutonoML on HCI, we first review opinions on whether automated context understanding and the integration of high-level cognitive mechanisms are plausible directions for future research. At an initial glance, such a pathway does not seem overly attractive to most of the modern data science community, although it is unclear whether that is because of a cost-benefit analysis or simply tunnel vision. Much cutting-edge DL research focusses on automatically building massive pre-trained data-driven models and then adapting their learned knowledge to specific tasks. This philosophy has even been recently branded as one of 'foundation models', discussed primarily in terms of very large neural NLP networks [32]. The movement has been motivated by the initial success of the OpenAI GPT-3 model, consisting of approximately 175 billion parameters [40], as well as Switch Transformers, consisting of nearly 1.6 trillion parameters [84]. Certainly, data-driven learning models have accumulated many significant achievements, being both flexible and capable of discovering complex statistical relationships between features in large amounts of data.

However, as expressed by a growing portion of the community, the rapid gains of the DL era have slowed. For instance, despite the dramatically successful approach of AlphaGo and its successors for relatively constrained problems, it is claimed that the overseeing institute, DeepMind, has not achieved many equivalent breakthroughs for ML applications involving, for example, cancer or clean energy [91]. To some extent, this is due to ostensibly unpredictable dynamics in real-world data environments, to which induction-based learning is vulnerable. Other disadvantages include a reliance on large datasets, a heavy consumption of resources, and a lack of explainability and trustworthiness [126]. In essence, substantial progress may require insights and paradigms that are novel or, at the very least, new to the modern era of computational hardware.

A fundamental limitation of ML remains that most correlations are not causal, even if the former may indicate the latter [91]. Indeed, current data-driven learning models work almost entirely in association mode; see Section 4.2.3. It is argued then that, for an autonomous learning system to pass the Turing test and exhibit an intellectual level comparable to human intelligence, such a system must have mastery over causal knowledge [223]. Others go further and claim that these machines fail the Turing test because they do not inhabit the material/social world that human beings live in, thus having no means to understand it [91]. This assertion is standard, with similar claims that understanding the environment of a human does not rely on knowing the chemical reactions in the brain or 'soul' of a person; one must empathise with their circumstances [71]. Of course, some of the discourse veers into topics beyond the scope of this monograph. An AutonoML framework does not necessarily need to pass a Turing test to solve ML problems proficiently. However, automated context understanding may ultimately rely on interacting with humans and an open-ended world to both acquire contextual knowledge and stay abreast of dynamic changes.

On that note, the importance of dynamics and adaptation to the future of AutoML cannot be overstated. These topics were discussed in Section 3 at the data level, e.g. in terms of data/concept drift that do not break well-defined problem constraints. However, at a *higher* level, humans can swiftly respond to changes in a working environment, able to identify rare events and novel concepts after only a few presentations within observed data. The critical factor that discriminates this human capability from current state-of-the-art AutoML is the ability to reason with limited examples, leveraging rich contextual knowledge. In particular, deliberative decision-making does not typically rely on fine-grained data, instead generalising conceptual notions for subsequent use in reasoning. For instance, studies suggest that children utilise high-level abstractions to reason about the distributional and dynamical properties of objects, discovering the underlying causes of observed events and using this knowledge to guide solution search when facing new tasks [294]. As a result, the acquisition of rich domain knowledge and the compilation of conceptual models, combined with subsequent reasoning, is asserted to be a much better form of learning than the pattern recognition employed by current ML algorithms [197]. Moreover, adaptation on a higher level seems to be most successful if there is a ready-to-use knowledge base to leverage, e.g. by building hierarchical priors to support learning-to-learn [36] or constructing abstract models of the world to enable learning new notions with the aid of previous experience and similar concepts [149].

Importantly, while an increasing number of researchers have opined for the integration of high-level cognition into smart autonomous systems, some have urged caution about directly imitating human intelligence [158]. While certain human capabilities are phenomenologically appealing, there are fundamental distinctions between biological brains and computational systems, e.g. in terms of organisational architecture, connectivity, speed, scalability, adaptability, and energy consumption. Furthermore, the development of an AutonoML system is arguably limited only by modern constraints of physics/technology and the ingenuity ceiling of engineers. In contrast, for all of its merits, biological evolution is slow when adapting to the complex hierarchy of modern-era desires that correlate with biological fitness, e.g. solving societal problems rather than focussing on immediate survival. For instance, humans did not evolve to process large-dataset ML applications mentally, and there is evidence that biases and cognitive limitations make humans particularly bad at some forms of reasoning [136]. Presumably, such cognitive processes are not a biological imperative. So, while human intelligence remains far advanced in open-ended environments, serving as a helpful inspiration, those that argue caution comment that imitating narrowly-defined human skills via collecting data and building DL models does not contribute to tackling major issues in AI. Instead, it is argued that computer-appropriate solutions should be sought for bringing meaning/reasoning into autonomous systems, integrating causal representation/inference, and developing computationally tractable representations of uncertainty.

Extreme views are likewise moderated when contemplating how far to go with knowledge-driven learning. Few modern perspectives espouse the belief that AI research should return to the era of expert systems. While deductive reasoning and conceptual generalisations have clear benefits, the fallacy remains that rules defined by human experts are constrictive and ultimately do not capture a diversity of real-world situations. In fact, one study suggests that most of the knowledge humans use in daily life, especially that which is acquired experimentally, is tacit [226]. The study supports this claim with an example of riding a bicycle, stating that it is hard for even an expert to teach this via complete explicit instructions. For instance, countersteering is an act that everyone is forced to intuitively master in order to keep balance, yet the action remains an unintuitive concept in physics to many; casual bicycle riders do not deliberatively think that turning left first requires steering right. Accordingly, the study concludes that "we can know more than we can tell", asserting that skills are a general prerequisite to articulating knowledge. Such skills seemingly cannot be acquired by merely following rules in a textbook, instead requiring interactions with operational environments and other agents, applying search and learning, and adjusting from trial and error. It follows that the process of data-driven ML, exemplified by the success of AlphaGo in a restricted environment, is suggestive of acquiring that tacit knowledge [91]. Thus, proponents of knowledge-driven learning typically argue for it to complement modern ML/DL, not replace it.

There are unsurprisingly many views on the requirements that a hybrid AI system – this monograph projects such opinions to AutonoML – should fulfil. Some researchers state that statistics and data-driven ML should incorporate reasoning, knowledge, common sense, and human values [188]. These claims add that AI systems need to maintain a cognitive model of their operating environments, and these must consist of accessible and reusable real-world knowledge. In effect, such programs need not only to identify entities, as current state-of-the-art learning systems do, but also conduct inference and reasoning regarding the relationships between entities. Ideally, the systems should also be capable of representing and inferring human values based on accumulated common-sense knowledge. These are fine principles, but other works have indicated that identifying the fundamental building blocks of an autonomous learning system is difficult enough as it is [99, 148]; it will be challenging to compose these blocks and formulate effective techniques to make an AutonoML system adapt and evolve in response to learned knowledge.

Nonetheless, hybrid systems based on data-driven and knowledge-driven learning, acting in concert, have broad appeal. There are two fundamental aspects of human cognitive behaviour that motivate standard proposals: one, the ability to learn from experience and, two, the ability to reason with learned knowledge [299]. Additionally, strong prior knowledge and inductive biases are commonly suggested for making inferences beyond the data [289]. One study argues that humans acquire this prior knowledge by learning to learn, transferring knowledge from low-level perception to high-level reasoning, and building compositional representations that capture multi-level causal structures underlying a problem/data environment [164]. Several concepts in current ML theories/practices mirror these notions and may be helpful approaches for their integration: transfer learning, representation learning, and multitask learning. The study also advocates many of the ideas discussed in Section 4.2. In summary, it states that intelligent systems should (1) form causal models for understanding/explanation rather than limiting themselves with pattern recognition, (2) leverage theories of physics and psychology to enrich learned knowledge, and (3) collect/generalise knowledge for new contexts based on the ability to build compositional representations and learning-to-learn methods.

Despite the strong favour for this possible path within sections of the data science community, the reality remains that constructing an open-world AutonoML system is not an *immediately* feasible project. Many requirements must be satisfied. For instance, such a system must be able to react to its working environment, conduct reasoning based on prior knowledge, generalise knowledge from data, provide explanations for outcomes, and make decisions for diverse situations based on available knowledge and experience. All of this must be done in a continuous online fashion to make better decisions over time [162], as it has already been stated that it is virtually impossible to prepare for an open-ended ML application. Indeed, these autonomous systems need to regularly revise their behaviours and correct errors/biases to refine their learning processes and accommodate current contexts. Furthermore, AutonoML also needs to support effective HCI to acquire knowledge from humans; see Section 4.4. The point here is that, although automated learning systems have displayed significant progress in the past few decades, at least on a scattered in-principle basis, only those with narrow focus have been realised in practice, possessing limited functionalities and purposes, e.g. image identification, label classification, prediction, or NLP [33]. Given that the approaches behind such narrow forms of ML depend on the availability of big data and the computational power of modern computers [186], it is an open question as to whether the inclusion of high-level cognition will be more expensive, e.g. in terms of a conceptual 'mental model', or more efficient, e.g. in terms of extrapolating knowledge via symbolic logic.

What is clear is that, should the state of AutoML research ever reach the stage of automating context understanding, there is plenty of literature from which to source inspiration. Indeed, the dual-process model of human thought presented in Section 4.2.1 has motivated several approaches. One study contemplating the design of a hybrid learning/reasoning system [33] argues that the description of System 1 already aligns very well with data-driven ML, formulating models from sensory data to handle perception-based activities such as reading and pattern recognition. There are, of course, a few discrepancies, with traditional ML struggling with causality and uncertainty, unlike System 1. Meanwhile, System 2 seems more connected to conventional approaches used widely in the AI community, e.g. logical reasoning and planning. Essentially, System 2 is immediately relevant when utilising explicit knowledge, symbols, and high-level concepts. Another proposal similarly presents a framework with many layers [25], aiming to integrate sub-symbolic/neural modules and symbolic/cognitive models within an autonomous system to pursue human-competitive levels of intelligence. Such a framework can theoretically cope with the challenges of an open-ended world, including the need for real-time bidirectional learning, adaptation to changing environments, and handling the emergence of new entities that were not covered by training, e.g. objects, events, and tasks. The authors of the proposal additionally claim that an autonomous learning system needs to operate with various levels of representation corresponding to different layers.

Accordingly, a hierarchical AutonoML architecture could include a neural layer learning on the data at its base, an anchoring layer learning elementary symbolic representations of objects, a reactive layer dealing with critical situations, a DL layer learning on more abstract levels, one symbolic layer conducting reasoning and planning, and an even higher-level symbolic layer managing the essential elements of an ontology. This multi-layer approach is common in the literature, often organising various forms of complementary data/knowledge processing. In fact, one study argues that knowledge could itself be considered a multi-layered phenomenon appearing at various abstraction levels, with interactions between those levels also contributing to its manifestation [26]. In this state, knowledge can be impacted by interactions with working environments, and it can additionally be linked with actions, perception, and thinking. Hence, the study contends that a future paradigm for the architecture of smart autonomous systems should involve the accumulation and storage of knowledge in a way that captures system-environment interactions, transfers insights in action patterns to symbolic levels, reasons on abstraction levels, and adapts/repairs initial representations for new contexts. Whatever the case, the many methods and perspectives of hybridised learning, combining data-driven DL with symbolic reasoning, are reviewed in greater depth elsewhere [99].

Ultimately, while we do not back any particular forecast for AutonoML, especially given that there may be alternate routes to open-world learning based on unfolding complexity and emergent mechanisms [70, 148], the path described in Section 4.2 appears plausible. Certainly, modern ML approaches are insufficient to support trustworthy inference based on limited observations or in the face of newly encountered concepts/contexts. So, if an effective bridge is developed between sub-symbolic data and high-level symbolic representations, knowledge-based methods could potentially elevate the adaptability of AutonoML. This outcome is especially the case if a system is equipped with a conceptual meta-knowledge base comprising symbolically coded expertise in various domains; such a construct could help contextualise novel encounters or, via logic-based inference, flesh out a dataset [25]. As a result, the combination of data-based sub-symbolic processing and complicated abstract reasoning could potentially compensate for the weaknesses of each other, operating in a continuous cycle of learning, processing, reasoning, updating, and adapting. One may imagine that this will be sufficient for AutonoML systems to understand – or autonomously pursue an understanding of – whatever problem context a stakeholder presents. So, with the installation of this new capability, should one expect the nature of HCI to change?

4.4 Roles and Modes in Relation to Autonomous Open-world Systems

The distinction between proficient AutonoML in an open-ended environment and proficient AutonoML in a constrained setting simply comes down to whether an associated system can autonomously develop an understanding of problem context without human intervention. Accordingly, the impact of full technical automation on the roles and modes of human interactions, described in Section 3.2, remains mostly applicable for the upgraded technology. However, with the automation of context understanding, the nature of HCI will likely evolve in a couple of ways. First, where human involvement has shifted focus or intensity due to lessening labour demands, extending the scope of automation should continue these trends. Second, where novel machinery is integrated into existing systems to enable automated context understanding, e.g. cognition mechanisms, new roles/modes of interactions will arise to support these operations. We now describe the expected impacts of open-world AutonoML on HCI more broadly.

Humans launch ML applications in a more general manner. Given that open-world learning requires automated context understanding, it stands to reason that the initial phase of the ML workflow will be most affected by this technological upgrade. Thus, when AutonoML systems are equipped with the capacity to generalise and transfer higher-level knowledge, possibly with conceptual reasoning mechanisms, humans will no longer need to be as narrow and precise with definitions of an ML problem. For instance, one can envision a stakeholder asking an AutonoML system through a UI to set up a project by supplying the following terms: prediction, bushfire, and USA. If the application is 'from scratch', a cognitive layer might source terms that are associated online, if not causally related, with 'bushfire'. These may form headers for a template dataset that the system provides back to the stakeholder to fill out. Perhaps the dataset is immediately populated with publicly available data that the AutonoML system suggests is relevant for training. Even better, if a knowledge base of past ML applications is available to the service, possibly consisting of an Australia-specific model, solution search may be warm-started with a decent ML pipeline. This notion of porting to similar settings is considered integral to any framework that would be considered a smart adaptive system [94].

Of course, we still consider problem formulation a human responsibility, even if it can now be done in loose general terms. However, at its most advanced, AutonoML may have the capability to designate and pursue sub-problems independently. For example, consider a human setting up carefully constrained prescriptive problems for stock investments, property purchases, and bank savings. With sufficient capability for open-world learning, it is conceivable instead that a user simply provides a current multi-faceted portfolio and tells the AutonoML system, "maximise money in one year". The service may then work out on its own what sub-objectives to pursue based on the contents of the portfolio.

The nature of HCI with AutonoML systems continues shifting from direct instruction to collaboration. This evolution is based on the fact that, to the degree that AutonoML can be trusted, automated context understanding pushes the limits of its independent performance. So, while the progression of technical automation from Section 2 to Section 3 reflects graduation from micro-management to a more moderate form of direction, stakeholders should be no more than laissez-faire managers for a proficient open-world learner. Human intervention may even harm performance. However, many believe that, even in the case of system autonomy, humans and machines may still need to coordinate when performing joint tasks [81]. This assertion is because humans and AutonoML systems have complementary strengths [316]. Humans have strengths in leadership, teamwork, creativity, and social skills. Programs are capable in speed, scalability, and quantitative computation. So, in practice, the relationship between human and machine may even be seen as one of collaborative equals, at least for attacking an ML problem. In fact, it has been proposed that humans and autonomous systems should form a hybrid decision-making collective by reaching consensus or compromise, with both entities sharing moral values and ethical principles [110].

Naturally, there is no guarantee that principles will automatically be shared in real-world contexts [242], so it is an open question of how to embed these into a framework. Actively learning human values via experience is one possibility; in such a case, human involvement in the data collection phase is considered essential, especially for ethically complex problems [190]. Additionally, an AutonoML system could potentially produce a set of ML solutions, rather than any single optimum, on which humans would preference-vote. The idea here is that the system indirectly learns human values from the choices that are made and possibly applies them in the future. Whatever the approach, it is clear that, if cooperative interactions are to be pursued as part of HCI for AutonoML, then humans must have some way of ensuring principles are aligned. Biases and discrimination in ML solution outcomes must be corrected. Alternatively, while engineering an AutonoML system, stakeholders may preemptively identify which tasks and conditions are safe for independent operation and which require human interventions.

So, assuming alignment is possible, what would HCI look like in a collaborative relationship? One work states that the critical human roles in a collective process would include defining problem context, exploring data/knowledge used by a system, understanding the outcomes of an ML application, providing a correct interpretation of the results, and intervening when necessary via feedback [190]. Such feedback can help AutonoML systems extend their knowledge base and enhance decision-making efficiency and effectiveness. To envisage this procedure even more concretely, consider the design process of chairs using a computer-aided drafting system called Dreamcatcher[5] [237]. This framework can automatically create multiple industrial designs according to the specific criteria provided by a designer, e.g. materials, required function, and cost. Some viable solutions may even surprise the designer, emphasising the power of a computational algorithm. However, the designer still has collaborative influence, using an interactive UI to provide feedback on user likes and dislikes. Dreamcatcher then adapts its solutions in real-time to these comments. Ultimately, the designer has the final call, deciding whether to approve an end product, but the system handles the labour of the drafting process. In this way, stakeholders can avoid tedious tasks to focus more on uniquely human-centric facets, such as professional judgement and aesthetic sensibilities [316].

Novel roles arise to support collaborative interactions. After all, while democratisation is an aim of the AutoML endeavour, Section 2.1 indicated that there will always be varying degrees of system proximity among stakeholders, especially when large organisations run ML applications. Accordingly, some humans may have to work along the interface between a team and an AutonoML system. For instance, one proposal names two new roles that may arise [316]: explainers and sustainers. Explainers require a mix of technical skills and domain knowledge, serving to translate the behaviours of smart autonomous systems to other users. Their activities can be crucial in evidence-based fields, such as medicine and law, where non-experts need to understand how ML solution outputs came about, e.g. pharmaceutical or legal recommendations. Importantly, the prospective role supplements the explainability mechanisms discussed in Section 2.4.3, assuming they are available. For example, explanatory mechanisms may tell explainers why a self-driving vehicle took actions resulting in an accident, but the explainers then contextualise this information and communicate it to insurers and law enforcement agencies. As for sustainers, their role is intended to prevent any harmful outcomes of an autonomous system. This agenda may involve reviewing risk analyses compiled by explainers, as well as checking/validating data, outcomes and mechanical behaviours to ensure compliance with fairness metrics and ethical norms. In some instances, sustainers may also assess whether collected data is used responsibly to, for example, maintain user privacy.

Humans monitor and validate AutonoML systems frequently and in real-time. Most opinions on advanced AI believe that, even with the capability for independent action, human supervision is inescapable [81]. Computational systems are not biologically human, so there is always a concern that acquired knowledge and enacted decisions may conflict with human rules and values. The potential for autonomous systems to induce negative consequences through unpredictable actions [28, 277] becomes even more significant in open-world settings. Whereas stakeholders can have some degree of expectation for behavioural risks in a well-constrained problem, all bets are off when the constraints vanish. There have already been hints of these dangers for modern applications, with

[5] https://www.autodesk.com/research/projects/project-dreamcatcher

the Holocaust-denying Microsoft Tay bot acting as a cautionary tale for open-ended learning on, in this case, Twitter. Thus, once an AutonoML system extends from extrapolating data to deducing high-level concepts and formulating action plans for complex problems, even more opportunities arise for unintended behaviour. Certain voices in the data science community additionally state that, not only should autonomous systems be subject to human oversight, trustworthiness requires auditing from a body that is independent of an application development team [266].

Emergency procedures are established to override undesirable AutonoML behaviour. This outcome is a natural extension of the aforementioned need for oversight. Autonomous systems interacting with complex environments are unlikely to act optimally at all times. There are many examples where excessive autonomy resulted in disaster, e.g. Patriot missiles shooting down two friendly planes [28] or design flaws leading to Boeing 737 MAX crashes [210]. Many catastrophic failures actually occur because of operations within an open-ended environment, where the system fails to understand a new encounter or a nuance. As a result, researchers have considered how best to involve humans in activating emergency protocols, e.g. with a "big red button" [216]. Most of this discussion occurs in the broader literature on AI, so it is difficult to predict how relevant all the considerations are. For instance, some research is concerned about autonomous systems learning to correlate shutdown with performance reductions in pursuing an objective, particularly if these programs operate persistently on long-running applications. Within frameworks that are engineered in complex or emergent fashion, perhaps the system might then counter a shutdown by disabling the emergency button or obscuring processes/outcomes, i.e. lying. Accordingly, related research considers how to manage HCI in such an adversarial setting, safeguarding emergency protocols. However, other efforts also contemplate how to minimise disruption from such protocols. For instance, one study suggests that the "big red button" could seamlessly switch an autonomous system into an illusory environment, insulating the real world from adverse impacts [239]. Indeed, for an open-world AutonoML system with well-developed ML pipelines and substantial meta-knowledge bases, a complete reboot can be costly, so any procedure that avoids this is valuable.

Humans are forced to engineer more rigorous tests for AutonoML. Certainly, if an emergency button serves as a reactionary countermeasure to open-world AutonoML going off the rails, then a powerful testing suite serves as a proactive approach to avoid disaster. Furthermore, unlike emergency protocols, the alternative of continual self-evaluation and error diagnosis [11] does not need a human to trigger it. This fact potentially makes a system-testing process a more effective and responsive safeguard, independent of human reflexes. Thus, both measures for disaster avoidance are likely to be employed in practice. That said, a prescribed testing suite is only as good as the adverse events it can predict. Admittedly, the virtually limitless instances of potential error in open-world ML may be compressible into a finite set of error types. High-level reasoning may not just automate context understanding; it might make it easier to comprehend error. Nevertheless, the point remains that great effort needs to be directed into making open-world AutonoML robust, and some of that human labour may be available once no longer invested into carefully defining ML problems. Hopefully, the resources saved by automated context understanding outweigh those required to maintain it.

Interactions are designed to instill human/social values into AutonoML systems. This facet expands on the requirements that collaborative HCI requires. After all, with AutonoML systems being granted greater agency on account of their presumed superior performance at conducting open-world ML, a key issue is how best to ensure that automated behaviours/actions are for the sake of humans [190], e.g. ethically compliant. Many commentaries in the literature have debated this angle, warning that, to be helpful to humanity, autonomous systems must grapple with potentially conflicting factors, e.g. in terms of legal requirements, ethics, and safety [301]. Some additionally stress that socioethical values vary for stakeholders in different multicultural contexts [68]; this makes it even more complicated to support the diversity of requirements and explain outcomes/actions to all. The takeaway here is that, while it is not currently clear how to embed human values into an AutonoML system, HCI will need to support such processes. This requirement becomes even more relevant when considering that AI applications are moving into the sphere of interpersonal relationships, e.g. working to adopt complex/subtle human characteristics such as sympathy. One example is an AI assistant developed by the Koko start-up [227], which analyses the emotions of a user in real-time and then provides advice appropriate for their current sentiment. In essence, once certain stakeholders begin to interact with ML solutions embedded in open-world AutonoML systems on an emotional level, psychological damage becomes a real risk that must be managed with intelligent approaches.

Society begins to debate what responsibility and trust mean concerning AutonoML. The projected evolution of the field means that such concepts are no longer as clear-cut as in Section 3.2. Indeed, while equipping AutonoML systems with automated context understanding has a functional purpose, enabling a way to effectively/efficiently solve general ML problems with minimal human effort, the development of high-level cognition inevitably raises profound philosophical and jurisprudential issues. In purely functional terms, an open-world AutonoML framework can be compared to a human problem solver that is (1) extremely proficient at applying inductive reasoning to observations and (2) has no individual agency, i.e. they do not come up with their own core problems to solve. Then again, perhaps even this assertion is controversial? Ultimately, we do not take sides in such broader debates, but it is clear that the notion of responsibility becomes critical to the topic of autonomous systems [67]. Admittedly, much discussion in this area relates more to modern systems, i.e. non-autonomous tools, and the risks of irresponsible usage when left to non-experts. Some argue then that experts should take more responsibility for the development/deployment of smart autonomous systems, ensuring consistency with fundamental human principles and values [68]; these works urge that responsibility be foremost in mind rather than an afterthought when designing such frameworks. However, when decision-making arises purely algorithmically, potentially via human-adjacent cognitive processes, society will have to debate where to attribute legal/moral culpability for adverse outcomes. Does the fault lie with the theoretician that designed the algorithm? The developer that implemented it? The user that applied it? The machine? Nobody, i.e. an “act of God”?

Closely related to this issue is the matter of trust. As Section 2.4.1 indicates, there have already been many proposals on how to define this concept. For instance, one proposal claims that trust in automation is a confluence of several factors [81], i.e. system-related, human-related, and situational. System factors include validity, reliability, robustness, interpretability, predictability, and the number of recent failures. Human factors capture the ability of a user to understand outcomes/processes, their willingness to trust, and other personal characteristics; these include culture, gender, and age. Situational factors cover time/resource constraints, required effort, workload, and availability. Even so, once open-world AutonoML enters the public domain, it will not be sufficient to simply quantify trust. Society will also have to decide the threshold for trust, per ML application, beyond which, if things go wrong, stakeholders are prepared to ask the hard questions on responsibility.

Regulatory and governance mechanisms are established for AutonoML. This outcome is predicted to be a social consequence of the open-world ML technology entering the mainstream. Indeed, it stands to reason that, given all the previously discussed risks of AutonoML operating in open-ended environments, legislation will eventually cover such services. One study contemplates this topic, discussing governance and regulatory mechanisms for autonomous systems [234]. It argues that the learning algorithms in these systems have to be transparent, fair, accountable, and concordant with values shared by all stakeholders. The study proposes a framework that combines a human-in-the-loop paradigm with automated mechanisms, seeking to identify a consensus set of values for all stakeholders and then ensure that the system complies with these established values. Of course, where human intervention is potentially limited, explanatory mechanisms become the primary tool for regulatory oversight. Such processes are also helpful for sufficiently supporting humans in deciding whether to trust and use an autonomous system [81].

In conclusion, it may seem that such a focus in this section on the negative aspects of open-world AutonoML may suggest that the technology should be avoided. However, counterintuitively, this simply highlights the prospective power of automation, in that human resources are freed up from so many processes that they are, by and large, foreseeably redirected to mitigating risks. This eventuality ideally ensures that open-world AutonoML is as robust and safe to use as possible, with most of the HCI involved serving as a conduit to deliver beneficial ML solutions/outcomes to stakeholders.

4.5 Optimising Interactions with Autonomous Open-world Systems

In operational terms, it seems likely that most stakeholders interacting with open-world AutonoML, for the functionality of solving ML problems, will do so in a collaborative manner. This eventuality would be a shift from HCI practices with modern AutoML, as discussed in Section 2, so it is worth diving deeper into how such collaborative interactions may be optimised. First, it is worth acknowledging that biological brains and computer programs, to date, have different strengths and weaknesses. While machines excel at swift computation, humans are better at tasks requiring strong cognitive and social skills, also being capable of quick adaptation to a wide range of unforeseen situations and events [158]. Therefore, to optimise collaborative performance when solving a set of ML problems, it may be worth acknowledging individual strengths and using them to devise effective protocols/frameworks for distributing sub-tasks between stakeholders and the autonomous system. Finding the right balance for this is immediately challenging because it can be difficult to determine the appropriate amount of autonomy that establishes a good 'teammate' for a human stakeholder [1]. Too little autonomy goes against the core point of the automation endeavour, burdening humans with system management and control. Too much autonomy risks the rise of issues related to trust, safety, and explainability.

Contemplating further on how human-computer collaboration may be designed, one study delves further into the distinct qualities that each entity possesses [158]. It notes that AI systems are much better than humans when selecting and handling vast amounts of data quickly, correctly, reliably, and consistently. The nature of electronic circuitry also ensures that computer programs do not fatigue or suffer from emotional volatility and stress. Such systems are also arguably better at memorising insights and recalling skills. The study thus advises that tasks suiting these capabilities should be delegated to an AI system, despite noting that it is still crucial for humans to somewhat understand and master these processes so that sufficient action can be taken when an autonomous system errs. On the other hand, compared with standard ML/AutoML, humans are more responsive to unpredictable events, creating innovative solutions for open-ended problems that are ill-defined. These abilities are arguably based on human qualities, primarily high-level cognition, although there is a possibility that psychosocial skills contribute too. So, by delegating sub-tasks appropriately, the study mentioned above suggests a hybrid decision-making process could leverage the power of the human mind where beneficial while also addressing cognitive constraints and decision biases. Granted, a core idea behind automating context understanding and moving towards open-world AutonoML is to gradually shift the high-level cognitive load from human to machine. However, as with the full technical automation projected for closed-world AutonoML, we neither claim that mechanised operations will be perfect nor that they should never be complemented by humans, especially where stakeholders have access to knowledge that is not immediately available to the system. Thus, compromise designs that balance automation with meaningful levels of human control may arguably form the safest and most effective frameworks for autonomous AI [265]. Compromise designs do not need to be inflexible either, with mechanisms generally functioning automatically whenever they are understandable, predictable, and accountable. The ideal is that a UI simply allows human override when contexts produce issues, e.g. high levels of uncertainty.

Many more opinions exist on the topic of human-computer collaboration and its requirements. During one panel discussion, researchers argued that this higher form of interaction requires mutual goal understanding, predefined task co-management, and shared progress tracking [310]. They suggested that collaborative HCI should pursue an approach towards what they termed Computer-Supported Cooperative Work (CSCW). The CSCW perspective attempts to systematise key facets of cooperative work, i.e. awareness, articulation, and appropriation. High awareness indicates that both stakeholder and AutonoML system are cognisant of the operations that the other entity partakes in. Good articulation indicates that work can be split into logical units and distributed between stakeholders and the AutonoML system, subsequently integrated. Effective appropriation indicates that a stakeholder can adapt and apply an AutonoML system to their specific situation.

Awareness in particular, as in the exchange of information between cooperating parties, has been deemed particularly important to collaborative pursuits. One study asserts that, beyond following a common protocol, humans and an autonomous system must have a shared comprehension concerning the objectives of a problem and the aspects that are relevant to them [157]. In the terminology defined within this monograph, there must be adequate HCI support for ensuring that problem formulation is sufficiently comprehensive and its requirements are propagated into an AutonoML system, especially given that this initial sub-phase of an ML workflow still relies on humans. That said, the study continues to suggest that the comprehension of goals and relevant processes to conquer these goals can abstractly be considered a mental model, serving as the basis for both humans and systems to plan goal-driven approaches for arbitrary situations, reacting capably to dynamic changes in working environments. Consequently, an ontology-based method is proposed to maintain situational awareness, represent/infer knowledge about operating contexts, and communicate individual perceptions between humans and the system. Indeed, a shared level of situational awareness is commonly thought to be a fundamental requirement for successful coordination between collaborating parties [81].

Now, once again, we restate that an ideal open-world AutonoML system should be proficient at solving general ML problems independently, so, while joint operations for the pursuit of *learning* should be supported, they may not be required in all cases. Thus, most modern discussion about HCI concerning advanced AI keeps returning to matters of *oversight*. Herein lies a conundrum: the more automation a system has and the more robust/reliable it becomes, the less aware human stakeholders are of its situation and the more challenging it is for them to take manual control when needed. In response to this, one research effort proposes a human-autonomy system oversight (HASO) model [81]. Schematised in Fig. 9, HASO describes important system-design characteristics that potentially influence human performance when working with autonomous and semi-autonomous systems. The framework consists of features relating to system robustness/reliability, operator interfaces, and three paradigms for interaction. In general, situational awareness and workload are vital to determining the quality of oversight and interaction outcomes, dictating where humans allocate their attention within an autonomous system. Situational awareness can be heavily impacted by UI design and the manner in which HCI is performed. Of course, increasing system reliability and robustness can increase the trust level of operators, thus somewhat relaxing the need for stakeholder attention.

Based on existing studies of UI characteristics and automation paradigms described in the HASO model [81], guidelines can be proposed for the design of trustworthy autonomous systems,

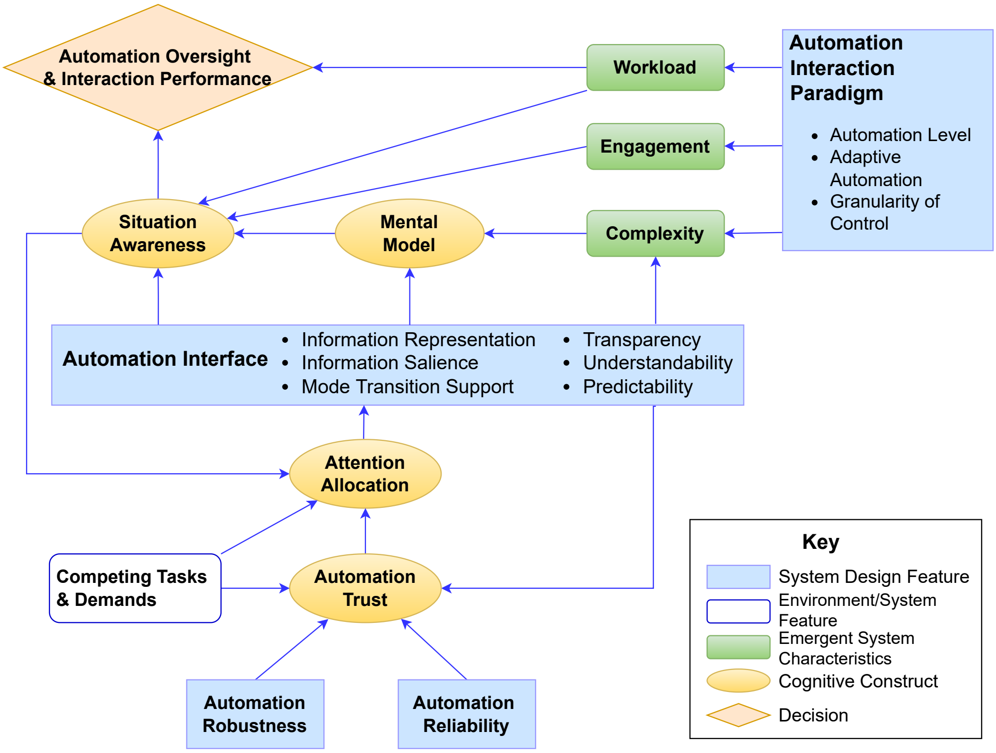


Fig. 9. A human-autonomy system oversight (HASO) model [81]. The paradigm presents major system design characteristics impacting human cognitive processes during engagement with an autonomous system, e.g. inspection, intervention, and general interaction.

supporting an optionally high level of human oversight and situational awareness. This guidance advocates for three key principles, i.e. supporting human perception, minimising system complexity, and improving situational awareness for operators. The proposal also cautions that increasing the degree of autonomy for a system must be accompanied by the development of UIs that provide a high level of transparency, understandability, and predictability regarding its behaviour. Simultaneously, these UIs must also provide operators with critical features to comprehend pivotal states and mode transitions occurring within an autonomous system. Admittedly, this discussion may seem abstract and unnecessary overengineering for modern AutoML services, which primarily focus on the model development phase of an ML workflow. However, as described in Section 3, AutonoML systems are already expected to become quite complex when automation extends along the other technical phases of the workflow [148]. Mechanisms are forecast to become wired up in an even more complicated manner once higher-level cognition enters the picture. Accordingly, HASO and the like can become useful distillations of HCI requirements to pursue when complexity threatens to further obscure operating processes.

Ultimately, the pursuit of open-world AutonoML will require updated perspectives on supporting operation and oversight with solid HCI strategies and mechanisms. Traditional thinking in this space tends to be one-dimensional, in that the price of pursuing automation is a necessary reduction in human controls [219, 262]. However, modern viewpoints argue that this perspective is incorrect, especially with the surge in concerns around reliability, safety, and trust. An example of one such counterpoint proposes a two-dimensional Human-Centered Artificial Intelligence (HCAI) framework that treats automation and human control on orthogonal axes [266], emphasising that humans should have the *option* of intervention alongside the *option* of letting the machine do all the work. According to the HCAI framework, a good design embraces the advantages of automation, including sophisticated algorithms, large-sized databases, powerful computational performance, and advanced sensors. A good design also supports human capability via smart UIs, augmenting, improving and empowering stakeholders so that they may use, oversee, master and refine autonomous systems. All this requires well-engineered services that are transparent, so that machine states and operational processes are understandable, indicating when humans should be involved and when they may sit back and relax. Admittedly, the devil is always in the details, and fully automated AutonoML systems are yet to be implemented for well-defined problems, let alone open-ended environments. Engineering issues may yet unveil unexpected challenges for how UIs connect with internal mechanisms. Nonetheless, the ongoing conversation about HCI design in the broader AI literature is almost directly applicable to ML. Hashing out principled approaches in the present can only accelerate technological progress once the world finally opens up for AutonoML.

5 CRITICAL DISCUSSION AND FUTURE DIRECTIONS

As of the early 2020s, there exists an abundance of review articles on the *mechanics* of AutoML, even though few are as broadly scoped as those that precede this monograph [70, 148]. However, deeper discussions on AutoML ergonomics, studying how humans and these systems can most effectively interact, are sparse. In fact, while recent work has surveyed how modern-day AutoML technology has organically evolved to suit human engagement in real-world applications [258], comprehensive considerations of HCI, present and prospective, are currently novel in the field. Thus, having already reviewed perspectives on HCI in detail, mainly how it is likely to evolve alongside expected advances in high-level ML automation, we present an overarching discussion of critical themes in Section 5.1. We then conclude in Section 5.2 with an examination of potential future directions for HCI research in the field of AutoML.

5.1 Critical Discussion

Although a cautious perspective is always appropriate, the mainstream emergence of AutoML theory/practice has achieved many remarkable results over the last decade. Initially propelled by the optimisation community as an approach to tackle the long-standing problem of algorithm selection, AutoML has since begun to expand to the other phases of an ML workflow, e.g. data engineering, deployment, and monitoring and maintenance. As progress continues, many researchers have developed their own views on what the 'final goal' of the endeavour may be. Some are motivated by pushing the mechanical limits of what might be achievable, e.g. whether the learning and problem-solving capabilities of an independent AutonoML system may ever rival and surpass those of a biological brain. Others appear invested in the field for the possibility of more human-centric benefits, e.g. making ML models more transparent, fair, reliable, and accessible to non-experts [319]. Unsurprisingly, these agendas need not necessarily be exclusive, even though limitations in computer resources will always put practical constraints on which system features should be prioritised. Ideally, hardware willing, AutoML will advance to a state where a framework is capable of autonomously searching, learning, reasoning and adapting within arbitrary problem/data contexts, while still supporting trustworthy collaborations with humans and other mechanised systems.

Whatever the state of AutonoML becomes in the long-term future, it seems clear that, at one level or another, humans need to remain in the loop for AI systems [39]. This sentiment is echoed by ML pioneer Michael Jordan [229], who claims that "we will need well-thought-out interactions of humans and computers to solve our most pressing problems". His view additionally suggests that ML research should not prioritise a mimicry of human thinking in the pursuit of AGI. The power of ML is in discovering hidden insights in multiple large datasets, mining obscure patterns, and recommending novel courses of action. In essence, the advocated ideal is that learning systems continue to act in applied service to humans, augmenting biological intelligence and offering new services to users in varied domains, e.g. commerce, manufacturing, health care, and transportation. Of course, in purely practical terms, autonomous systems will continue to need good HCI solutions for the foreseeable future, possibly for decades [81], so that human controllers can collaborate or, at the very least, oversee system performance and safety. Certainly, on the operational side, the involvement of technical stakeholders is expected to remain valuable simply due to their domain knowledge as well as the use of intuition/experience to cover the blind spots of an AutoML mechanism [319]. However, ensuring AutoML usage remains socially responsible is also a strong motivation for efficient and productive human-computer communications. Accordingly, the uptake of AutoML technology may be better served by proactively researching and planning strong HCI guidelines/protocols at every stage of its technical evolution, rather than waiting for UI paradigms to arise reactively as an afterthought [258].

The historical neglect of debating human roles in the context of AutoML means that many human-centric questions, both important and challenging, have mostly been overlooked. For instance, in which situations and to what extent can AutoML-supplied predictions/prescriptions be believed? Are there thresholds that demarcate when independent operation is appropriate? Whose interests are AutoML systems serving? Are the outcomes fair for everyone? At which point of an ML workflow does a discovered bias enter the problem-solving process? How can humans and AutoML systems best compensate for the limitations of each other? Is there an optimal way of translating outcome into impact and human benefit? The answers to many of these questions have broader implications, tying into what society expects for the roles of humans concerning the evolution of industry.

Notably, in recent years, certain terminology has been popularised for the notion of industrial revolution. Put simply, the following associations have been made: Industry 1.0 with steam/water power, Industry 2.0 with broad electrification, and Industry 3.0 with digital computer technology. It has been argued that society is currently undergoing a fourth revolution based on the rising importance of data networks to modern applications. Specifically, Industry 4.0 can be characterised as contributing to an interconnected world, where physical items such as sensors, devices, computers and enterprise assets are linked, communicating with each other and the Internet [271]. Commentary in the literature suggests that Industry 4.0 appears to be the perfect setting for the construction of smart autonomous systems fuelled by big data and ML [298]. This claim is because the development of large-scale machine-to-machine communication, such as with the ‘Internet of Things’, arguably supports the engineering of complex systems that are better able to work, self-monitor and diagnose problems without human intervention [201]. In effect, the focus of Industry 4.0 appears to be an increasing level of automation and digitalisation, mainly manifested via AI-driven technologies. Unfortunately, as is asserted by some sectors of the data science community, human roles and social considerations, e.g. fairness, seem to take a back seat in the Industry 4.0 paradigm, at least with respect to pure technical capability.

Just as this monograph attempts to advance the discussion around AutoML from technical operations to HCI factors, so too has there been a movement to shift the paradigm of progress into what is being called Industry 5.0. Indeed, this concept of industrial revolution appears to be a logical progression of the existing Industry 4.0 paradigm, simply bringing human, social and environmental factors back into the bigger picture [203]. In effect, the driver for industrial progress is no longer automated task proficiency, which remains important, but is instead a collection of values including human-centricity, sustainability, resilience, and ecological/social benefits [37].

A broader range of human demands and interests become core to the development process of technologies. Consequently, the adoption of technological processes is dependent on an ethical rationalisation of how human values are served by a system, rather than relying on a purely technical viewpoint [203]. For instance, many DL approaches that are considered state-of-the-art under Industry 4.0 may be less appealing under the Industry 5.0 banner due to energy usage and their impact on climate change [70]. Additionally, the notion of *co-working* becomes very important to Industry 5.0, rebounding from the pure automation focus of the previous revolution. The idea is that merging human and machine abilities will complement disparate strengths and cover individual weaknesses. This paradigm supports humans using their creativity in problem-solving, formulating new roles specific to problem context, and improving skills and knowledge in collaboration with autonomous systems. Clearly, Industry 5.0 puts a greater weight than Industry 4.0 on intelligent human-machine interfaces, the integration of human cognitive abilities into AI systems, and collaborative interactions. These characteristics are pursued in the hope that products and services will satisfy human needs, reduce environmental impacts, and use energy resources efficiently [203].

With all the above context, this review contributes to fleshing out the overarching picture of human interactions, i.e. their roles and modes, in the context of the evolving AutoML field. Section 2 anchored this perspective by first establishing the stakeholders and phases of an ML workflow that are currently involved in modern-day AutoML interactions. Given that AutoML can be assessed as having far further to go, simply in terms of technical operations, Section 3 proceeded to step forward in time and envisage full automation for every process following problem formulation and context understanding. This advance already has significant implications for how humans engage with these newly autonomous problem-solvers. However, with learning in open-ended environments as a holy grail for the field of AI in general, the subsequent Section 4 considered how AutonoML could approach such an ideal.

Open-world learning may not be entirely blue-sky, as knowledge transfer between similar and/or different domains has been investigated intensively as part of the meta-learning research agenda, e.g. reusing knowledge and search strategies from prior tasks to speed up ML pipeline construction for similar problems [4, 318]. The greatest obstacle remains identifying when and which meta-knowledge can be leveraged once incoming data and related concepts, perhaps even ML tasks, are no longer stable. It seems unavoidable then that the pursuit of context understanding must be automated. Perhaps the most prevalent opinion on this matter, especially from the traditional AI community, is that this cannot be achieved without installing AutonoML with the ability to reason, transfer and generalise knowledge on a higher level than data. Whether this is the right approach is debatable, but it is at least clear what the capabilities of an ideal open-world AutonoML system are, meaning that well-reasoned forecasts for HCI are available. There are, of course, grander implications when AutonoML systems become able to autonomously learn, self-monitor, correct errors, reason, and generalise knowledge. Some have speculated that a machine with these human-level qualities may even mark its own revolution: Industry 6.0. Perhaps a follow-up to this monograph will eventually be necessary, contemplating the roles and modes of human interactions for systems with general intelligence and self-cognition. For now, we limit speculation.

For now, present-day thought in the data science community cannot foresee a future where human involvement is completely extricated from AutonoML operations. A recent study specific to modern AutoML systems affirms the prevalence of an opinion that human developers need to provide inputs/feedback simply because a computer, in the absence of expert knowledge, cannot attain a human-level understanding of context beyond the datasets provided [311]. To many, it seems essential that technical and business stakeholders guide data cleaning/transformation phases so that AutoML algorithms can better satisfy application requirements. As a result of these views, the study advocates that an AutoML system should act as a collaborative tool that augments human capability rather than fully automating human operations.

In fairness, even were automated context understanding to be achieved, humans who formulate a problem are typically the ones who ultimately evaluate whether an ML solution satisfies their requirements. There is no standard procedure for developing trust, any more than there is a universal metric for fairness, and different cultural backgrounds will equip stakeholders with different expectations. Thus, even if an AutonoML system can solve a problem 'perfectly', human involvement is indispensable for ensuring that form of perfection accords with the definition a stakeholder chooses. Unsurprisingly, given all these motivations, several recent studies favour the development of human-guided AutoML to balance stakeholder control with automation [105, 168, 311]. Various proposals further suggest different requirements for the design of effective HCI [6, 105]. In effect, human-guided AutoML enshrines one mode of collaboration, where human users define their domain knowledge and high-level expectations of a system; the AutoML framework then performs its model search autonomously to create high-quality ML solutions for a given problem [105, 168].

Currently, it is not surprising that computer programs fuelled by sophisticated state-of-the-art ML algorithms struggle with adapting to new contexts without human innovation and rigid problem definitions. Existing AutoML mechanisms tend to automate only one aspect of an ML workflow, and often only in principle, e.g. feature engineering, model selection and HPO, deployment, or monitoring and model maintenance. Moreover, most applications still require heavy human involvement across the entire ML workflow to supervise/guide ML model construction, complement the lack of domain knowledge, verify results, and intervene when outcomes are unexpected. Admittedly, as new learning paradigms and supporting procedures are developed, so that automated high-level operations can be described as 'proficient', the need for human intervention is likely to relax. Section 3.1 mentioned AI systems capable of surpassing humans in Chess and Go without relying on expert knowledge as examples of this expected progression in AutoML. Even then, the research-and-development journey to fully autonomous open-world learning is a long one, beset by many challenges described in Section 4.1.

Accordingly, although human intelligence may not be the perfect standard for problem-solving, it can still inspire research in the way humans learn, infer, and efficiently address diverse and unpredictable situations within open-ended environments despite the absence of massive datasets. The challenge has long been known that an autonomous open-world system does not only have to collect relevant data regarding a new working environment; it also has to comprehend the *meaning* of that information and quickly model environmental dynamics to engage in a proactive decision-making process [80]. Likewise, another study argues that responding quickly to environmental changes requires capturing information from various sources, making decisions based on both obtained information and existing knowledge, and launching actions following these decisions [296]. Additionally, the system must have feedback about the impact of its actions on an environment so as to update its knowledge base and pursue better decisions in the future. This continuous cycle of learning may seem overly involved for most present-day AutoML systems that solve problems under a one-and-done paradigm, but the AutonoML frameworks of the future will certainly need to fulfil these requirements. Moreover, if collaborative operations become the optimal form of HCI for open-world AutonoML, careful design of advanced interfaces will be required to imbue a team of stakeholders and computers with a shared amount of situational awareness [81, 157].

Ultimately, the most significant motivating factor for further research into HCI may simply be what is the greatest weakness of modern AutoML offerings. Simply put, in pursuing automation, many current AutoML tools have obscured their internals, acting as black-box systems [319]. This situation immediately restricts those that wish to have flexibility in managing the ML process. The insertion of domain knowledge is likewise hindered without a clear understanding of what is happening internally [180]. Indeed, many have hinted at the dangers of an autonomous system that cannot show stakeholders what they are doing, why they are doing it, whether an action can be averted, and how a human may avert such an action [35]. Arguably, if humans are not allowed to provide inputs to an AutoML system, the bare minimum of interactivity should be the provision of outputs, i.e. explanatory mechanisms that allow stakeholders to understand system behaviours and the rationale behind decision-making processes. This support would allow users to at least determine whether it is worth accepting an ML solution, although being able to direct a learning process and integrate domain knowledge is obviously preferable. In fact, experimental observations and analyses of case studies have shown that ML model development can benefit from human involvement via the notion of 'soft information and knowledge' [285]. This concept covers additional knowledge that may be ill-defined or otherwise unavailable for pure machine-centric methods [54]; it includes theories, intuition, belief, value judgements, and so on. Regardless of details, the principle remains that people tend not to use results they cannot understand or explain. Therefore, explainability is critical to support understanding and build confidence, e.g. allowing for the diagnosis of odd behaviour. Section 2.4 indicated that these matters of trust in AI are highly topical in the early 2020s, and, without a serious effort from the AutoML community at grappling with HCI and AutoML ergonomics, uptake of related technologies may stall.

In summary, this monograph has reviewed several aspects regarding the roles and modes of human interactions with AutoML systems.

What are the types of stakeholders that may interact with AutoML systems? Many humans may directly or indirectly engage in an ML application, particularly as part of a business organisation. We categorise them as the following: technical, business, regulatory, and end-user. Technicians are typically responsible for developing, testing, deploying, monitoring, and maintaining ML models. Business stakeholders tend to translate business criteria into functional requirements for an ML solution, with some assessment of whether an application is feasible. Stakeholders in the regulatory group are in charge of independently assessing and verifying the safety and reliability of the systems, ensuring that they comply with legal, ethical and social norms. End-users are the ones that feel the impacts of an ML solution; if this is done knowingly via direct interaction, they can evaluate its usefulness, potentially provide feedback, and determine whether the solution is accepted. Due to differences in these roles, interaction modes and the flow of information between stakeholders and systems can vary. Therefore, this monograph clarified how different types of users currently interact with AutoML systems in Section 2.1.

What are the current interactions involved when developing, deploying, and maintaining ML solutions? To answer this question, we reviewed how stakeholders, primarily technical, can provide inputs into an ML application at every stage of an ML workflow. However, we considered this in the context of modern AutoML and also attempted to gauge the current balance of human and machine involvement in these activities. Discussions in Section 2.2 included how stakeholders guide the development process of ML algorithms, how they deploy and monitor ML solutions, and how such solutions are maintained and updated in response to changes in a data environment. We also reviewed the current and prospective communication modalities with which stakeholders can interact with an AutoML system or, more frequently, the ML solution it houses. These modalities, detailed in Section 2.3, covered UIs based on touch screens, voice, gestures, VR/AR, and even brain signals.

How do socioethical factors influence interactions with AutoML systems? As supported by various studies [73], data scientists only trust ML solutions provided by AutoML tools if mechanisms establish a high degree of transparency and understandability regarding system processes and outcomes. Similarly, end-users only accept ML predictions/prescriptions if the learning system can show the reasons behind these decisions [302]. For example, if an automated decision-making system in a self-driving car can explain how it identifies/avoids obstacles and provides warnings for various road situations within a simulated environment, users will have more trust in the autopilot system. Therefore, explainability is a critical factor that is likely to influence public uptake of AutoML technology. However, different forms of explanation have different impacts on human comprehension, so this monograph delved into the nuances of the topic, considering how interfaces affect explainability and how, in turn, explainability affects human interactions. Additionally, we linked trust in AutoML outcomes with other factors such as cognitive biases, fairness, and ethical norms. The importance of these facets, as well as ways to manage them with strong human-computer collaborative practices, was explored in Section 2.4.

Are there optimal design principles and requirements for AutoML UIs? Certainly, to foster effective interactions between stakeholders and AutoML systems, intelligent UIs are required [105], particularly those that allow a technician to specify ML problem settings, explore the characteristics of data, and constrain ML solution search spaces. A good UI essentially allows users to efficiently inject their domain knowledge, optionally assisting AutoML mechanisms in finding optimal ML pipelines. Such domain knowledge may be unidentifiable from merely inspecting the data, sourced instead from contexts where the data is generated, previous results in the literature, or simply conventional wisdom for applying ML models [105]. Furthermore, UIs must allow technicians to easily explore hyperparameter space themselves so that they may compare various generated solutions and potentially even reject them in favour of a manually selected ML pipeline. These are just some of the requirements to support model development; there are many other general guidelines for designing a UI that supports practical operations during deployment, continuous learning/adaptation of an ML solution, and other significant interactions. Together, these principles and their proposals were reviewed in Section 2.5. Of course, once AutoML is capable of problem-solving in open-ended environments, adapting and operating with heightened autonomy, the nature of HCI potentially evolves into a new form of collaboration. Thus, Section 4.5 returned to the question of UI design, detailing many proposals in the literature covering oversight paradigms, task division methods, and shared situational awareness.

How do the roles of humans change as AutoML technology evolves? The original purpose of the AutoML endeavour was to aid data science workers, and ideally non-experts, in finding optimal ML models for arbitrary ML problems with a minimal amount of human intervention. Even after a decade of concerted efforts, the field remains nascent, despite its great promise. However, it is expected that advances in technical problem-solving paradigms will eventually bring automation to virtually every phase of an ML workflow. Advanced strategies for search, meta-learning and adaptation will produce continuously operating AutonoML systems that are proficient, not necessarily perfect, at finding good ML solutions to constrained ML problems. In time, AutonoML research is then likely to open up the problem contexts in which a system operates, although the capacity for efficient adaptation to novel tasks and concepts may require revolutionary progress. Throughout this technological evolution, the roles and modes of human interactions are also projected to change. In current practice, described by Section 2.2, humans are usually involved directly in the development and deployment of ML solutions, selecting suitable training data and features, establishing configuration ranges for hyperparameters, assessing resulting pipelines, and monitoring the operation of solutions when deployed. Once full technical automation is enabled by strategic advances, discussed in Section 3.1, the direct involvement of stakeholders is no longer required. Instead, humans are more likely to focus on the innovative phases of problem formulation and context understanding, setting up constraints and working environments for AutoML systems, as well as supervising, auditing, and understanding the outcomes of AutoML systems. Once context understanding or the pursuit thereof is itself automated, the key roles of humans evolve even further; one responsibility is to maintain the novel mechanisms that handle knowledge generalisation and transfer, e.g. by fine-tuning a ‘mental model’ of the real world. Oversight of autonomous operations is another task, which is especially important in an unconstrained environment. However, ultimately, an open-world AutonoML system may start to be seen as a teammate, hence HCI being reframed as a conduit for collaborative problem-solving. Humans may become responsible for finding the most productive way to cooperate with an AutonoML entity, assessing socioethical factors in system behaviour, monitoring/validating outcomes in real-time, and seizing control in the case of erroneous operations. In short, we predict that human roles will likely prioritise tasks requiring high-level cognitive abilities, leaving AutonoML for the most part to do its job and proficiently solve ML problems.

5.2 Potential Research Directions

It is a common opinion that one of the most challenging issues of the coming years will be how to leverage the full power of autonomous machines while simultaneously supporting seamless coordination with humans, generating decisions that align with sophisticated values and needs [179]. The topic of HCI in AutoML is presently in its infancy, only recently emerging as the technological rise of AutoML motivates questions about how humans fit in. This transition in thinking, from a purely technical focus to one that ponders human-centric solutions, mirrors gravitation in the broader data science literature towards the concept of Industry 5.0. Therefore, given the seemingly inevitable attention that HCI will need to be paid, we now identify potential research directions for this topic.

- **Generating human-centred design principles for the selection of optimal ML solutions.** Crucially, while ML theory has been researched for many decades, it can be argued that ML technologies have only existed in the mainstream for a couple of those decades, and genuine industrial use has only taken off within the last few years. Consequently, there has been a significant historical focus on maximising the technical quality of an ML approximation, whereas satisfying human needs and desires is a relatively novel notion. The two aims do not necessarily have to compete, but, as Industry 5.0 attempts to recognise, technical performance is a narrow view of good performance. It is thus likely that human-centred design (HCD) principles [190] will become increasingly important in the coming years. The main activities of HCD for AI systems include the specification and comprehension of contexts and requirements, the latter being held by business stakeholders and end-users. Managing these processes well requires developing a solid understanding of who the relevant stakeholders are, what they care about, how they are expected to interact with an AutoML system, and what the working environments and constraints for the framework are. Hashing out ergonomic HCD principles in the near future should make the next generation of AutoML services much more amenable to collaborative interaction, thus enhancing functionality, operational efficiency, and UX.
- **Promoting transparency and explainability in AutoML systems.** This monograph stressed these socioethical concepts several times as being of surging importance in the current AI climate. Alongside related concepts such as reproducibility and predictability, these factors support stakeholders supervising the actions and behaviours of an AutoML system, giving them the ability to intervene when errors and other adverse outcomes occur. Certainly, humans are less likely to use and trust AutoML if they do not feel in control or are unable to explain system processes [266]. Thus, in lieu of loftier aims such as open-world learning, moving towards transparency with explanatory UIs and white-box system design may be an important and achievable short-term goal within the field of AutoML.

- **Integrating human values in terms of fairness and ethics into AutoML systems.** Once stakeholders can understand an autonomous system, the next step is to ensure its processes and outcomes adhere to human requirements. However, it is currently unclear how to do that because current definitions, rules and guidelines related to concepts such as fairness are generic and abstract [190]. It is not clear how to define every manifestation of bias and discrimination within the context of AI in a simple fashion [44]. Unsurprisingly, it is even more challenging to devise realistic solutions that address the fallacies. In contrast, there is a high bar of expectation for ethical and fair AI systems, which need to possess good normative models that easily integrate their behaviours into human normative institutions [179]. Trustworthy AI must not only explain how an outcome was achieved but also provide normative justification for how and why these results are consistent with human values. In short, much more research is needed to make ML and AutoML 'performant' in the broadest socioethical sense of the word.
- **Establishing concepts of operation for HCI in AutoML.** Concepts of operation cover the determination of tasks, the responsibilities of stakeholders, the degree of interrelation between operations that humans and AutoML systems separately partake in, and how interactions between humans and AutoML systems should be governed [81]. Much research into HCI for automated systems relies on a concept of operation that has stakeholders play the roles of supervisors, i.e. responsible for guiding and overseeing system performance. However, when machines become able to work more autonomously, a popular alternative for such a concept of operation is with humans and the system acting as partners or teammates, i.e. cooperating and coordinating on the tasks required by a problem [288]. Therefore, working out how to support collaborative interactions with UX design and other protocols is a future path for AutoML research. Already, it is known that well-designed UIs must provide humans with essential supporting functions that reduce workload, increase performance, and guarantee both safety and reliability [266]. Many more requirements have been proposed for concepts of operation that facilitate human-system cooperation [156]. These include establishing joint goals, communicating/understanding the roles of each other, having adequate mental models of team members, being predictable to each other, sharing and comprehending the status and intentions of other team members, being able to adapt and negotiate goals in new situations, and replanning collaborative problem-solving strategies if needed. Shared situational awareness likewise remains an important principle [81], and developing smart interfaces with sound UX design principles that accommodate this is appealing [157]. Essentially, it is up to the AutoML field to decide the ideal way that stakeholders should interact with automated services in the coming years. If the benefits of collaborative interactions are to be embraced, researchers and developers will need to determine appropriate concepts of operation for the future.
- **Building common protocols and design principles for open-world ML.** This monograph has forecasted, with supporting evidence, two major phases of progress to expect in the future of AutoML. Although innovation and ingenuity are required for any significant advances, fully automating most of an ML workflow – the first phase of progress – will unlikely require any substantially unconventional thought. As exemplified by AI systems dominating Chess and Go, continually improving search strategies may imbue automated mechanisms with sufficient proficiency for solving well-defined and constrained ML problems. On the other hand, the initial stage of an ML workflow, i.e. problem formulation and context understanding, still relies heavily on human creativity and versatility. If – the second phase of progress – AutonoML is ever to grapple capably with open-ended environments, developing a standard and comprehensive ontology for how humans and

machines are expected to operate in this space may be necessary. For instance, researchers and developers may need to contemplate the varied ways in which an ML problem and objectives can be defined, decide what knowledge base should support an automated process of context understanding, establish whether this procedure should be guided or restricted, and generally determine how to make open-ended environments manageable. Also, so that this endeavour remains feasible and practicable, it is worth contemplating how best to incrementally open up the constraints of ML problems, ensuring theory and practice can handle those tasks before advancing even further. For instance, a gradual relaxation may proceed as follows: (1) ensure AutonoML can adapt to changes in data distribution, (2) deal with changes in concept such as new features/labels without discarding already learned knowledge, (3) efficiently attack novel ML problems related by concept to previous encounters by transferring models and knowledge as appropriate, and so on.

- **Embedding situational and context awareness into AutoML.** The abilities to perceive situations and understand contexts are factors that imbue humans with the versatility to address diverse problems in complex and dynamic environments. Thus, to progress AutoML beyond restricted AutonoML into open-world learning, mechanisms will first need to be engineered to identify concepts/contexts that do not immediately fit with prior knowledge accumulations. Such a mechanism for situational awareness arguably needs to satisfy several characteristics [79]. First of all, an autonomous system must construct and maintain a model of itself and its operational environment, at least for relevant factors of interest. It must then be able to facilitate dynamic information flow and integrate low-level data into this model. Next, the situational awareness model must be able to actively learn from new insights captured by the system. Additionally, the model needs an effective well-structured representation of existing situations for quick recall, in real-time, whenever a similar situation is encountered. These principles are well-reasoned, but, given that a major focus of ML research is pushing the technical capability of deep neural networks, many proposals for high-level ML frameworks have not yet been translated into practical application. There is thus much scope for future research impact by merely verifying if proposed principles supporting prerequisite phenomena to open-world learning, such as situational awareness, are robust, complete, and useful.
- **Fusing data-driven search methods with higher-level reasoning.** Although advanced strategies for search and learning can bring success to the Alpha Zero and MuZero systems, modern-day ML is constrained to problems and data environments where contexts and relevant rules are well defined. In contrast, humans can transfer knowledge and insight among tasks, adapt to changes in a dynamic environment, reason causally and with abstraction, and logically generalise concepts rather than simply recognise patterns and detect associations [32]. Therefore, advancing the capabilities of AutonoML beyond situational awareness may require finding ways of integrating knowledge-driven learning and reasoning approaches with data-driven search algorithms. As the most speculative future research direction here, it is unclear whether this is the best path towards open-world learning. However, it is *a* path with plenty of expert backing and warrants investigation at some point; any successes here will definitively push the versatility and general applicability of AutonoML.

6 CONCLUSIONS

This monograph reviewed the roles and modes of human interactions with AutoML systems. It detailed current practices with modern-day packages and services, then extrapolated to an expected future where frameworks can proficiently run ML applications autonomously. This forecasted advance was broken into two sequential phases. First, AutoML reaches a stage of full technical automation at which, now capable of continuous independent operation, AutonoML can deftly handle constrained and well-defined problems. Second, AutonoML attains automated context understanding, at which point associated systems become capable of open-world learning.

To support the discussion of HCI and its evolution alongside this projected technical progression, different types of stakeholders were identified, as were their expected interactions at various stages of an ML workflow. These stages covered problem formulation, context understanding, data engineering, model development, deployment, monitoring, and maintenance. Additionally, the way users engage with an ML solution housed by an AutoML system was also considered part of stakeholder interactions. With this conceptual framework established, existing HCI literature was then reviewed in the field of AutoML and beyond. Accordingly, research opinions on UI requirements and approaches were discussed. This monograph also highlighted the topical issue of trust in AI, recontextualised for AutoML, detailing numerous factors that influence stakeholder engagement, such as explainability and human cognitive biases. Various perspectives on strengthening HCI were likewise surveyed, examining the problem of fairness and ethical compliance.

Having discussed the present state of human interactions in AutoML, this monograph began its forecast of HCI in earnest, identifying changes to expect as ML problem-solvers become steadily more autonomous. By and large, humans are likely to shift further and further into hands-off supervisorial roles, with less need to be involved in advanced search and learning processes. The freed labour may instead be redistributed to more creative and innovative aspects of ML applications, such as formulating problems better, enriching the understanding of their contexts, and interpreting ML solution outcomes. However, with more freedom and independence comes more risk and responsibility. Prevalent opinions maintain that mechanisms for HCI will need to support effective human oversight; this becomes even more important as problem constraints relax.

The end-point of AutoML evolution, at least within the scope of this review, is open-world learning. This monograph considered one possible pathway for the technology to attain this capability, reviewing research and perspectives on hybridising data-driven ML with both knowledge-driven learning and high-level reasoning. The consequence of elevating AutonoML with these cognitive capabilities is that systems will likely be seen more as partners or teammates of human operators than mere tools. Subsequently, opinions on how to optimise these collaborative interactions were surveyed. However, doing so in the best manner remains but one of many open questions in the field. Acknowledging this, the monograph ended by identifying and listing several promising research directions.

Ultimately, this review concludes that, while considerations of HCI for AutoML have begun in earnest, there is much further to go. It remains a frontier topic in an emergent field, nascent but both essential and promising.

REFERENCES

[1] Evan Ackerman. 2021. How the U.S. Army is turning robots into team players. *IEEE Spectrum* (2021). https://spectrum.ieee.org/ai-army-robots Accessed on 25 September 2021).

[2] Marilena Agathokleous and Nicolas Tsapatsoulis. 2013. Voting advice applications: missing value estimation using matrix factorization and collaborative filtering. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*. Springer, 20–29.

[3] Abbas Raza Ali, Marcin Budka, and Bogdan Gabrys. 2019. Towards meta-learning of deep architectures for efficient domain adaptation. In *PRICAI 2019: Trends in Artificial Intelligence*. Springer International Publishing, Cham, 66–79.

[4] Abbas Raza Ali, Marcin Budka, and Bogdan Gabrys. 2020. A Review of Meta-level Learning in the Context of Multi-component, Multi-level Evolving Prediction Systems. *CoRR* abs/2007.10818 (2020).

[5] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. 2014. Power to the people: The role of humans in interactive machine learning. *AI Magazine* 35, 4 (2014), 105–120.

[6] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N Bennett, Kori Inkpen, et al. 2019. Guidelines for human-AI interaction. In *Proceedings of the 2019 chi conference on human factors in computing systems*. 1–13.

[7] Edward Apeh, Bogdan Gabrys, and Amanda Schierz. 2014. Customer profile classification: To adapt classifiers or to relabel customer profiles? *Neurocomputing* 132 (2014), 3–13.

[8] Theo Araujo, Natali Helberger, Sanne Kruikemeier, and Claes H De Vreese. 2020. In AI we trust? Perceptions about automated decision-making by artificial intelligence. *AI & SOCIETY* 35, 3 (2020), 611–623.

[9] Ohad Arazi. 2020. AI Won't Replace Radiologists, But It Will Change Their Work. Here's How. *World Economic Forum* (2020). https://www.weforum.org/agenda/2020/10/how-ai-will-change-how-radiologists-work/

[10] Martin Arjovsky, Soumith Chintala, and Leon Bottou. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*. PMLR, 214–223.

[11] Thomas Arnold and Matthias Scheutz. 2018. The "big red button" is too late: an alternative model for the ethical evaluation of AI systems. *Ethics and Information Technology* 20, 1 (2018), 59–69.

[12] Alejandro Barredo Arrieta, Natalia Diaz-Rodriguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, et al. 2020. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion* 58 (2020), 82–115.

[13] Azure. 2021. Interpretability: Model explainability in automated ML. https://docs.microsoft.com/en-us/azure/machine-learning/how-to-machine-learning-interpretability-automl

[14] Rashid Bakirov, Damien Fay, and Bogdan Gabrys. 2021. Automated adaptation strategies for stream learning. *Machine Learning* (2021), 1–34.

[15] Rashid Bakirov, Bogdan Gabrys, and Damien Fay. 2017. Multiple adaptive mechanisms for data-driven soft sensors. *Computers & Chemical Engineering* 96 (2017), 42–54.

[16] Gianluca Baldassarre, Vieri Giuliano Santucci, Emilio Cartoni, and Daniele Caligiore. 2017. The architecture challenge: Future artificial intelligence systems will require sophisticated architectures, and knowledge of the brain might guide their construction. *Behavioral and Brain Sciences* 40 (2017).

[17] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. 2019. ObjectNet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In *Advances in Neural Information Processing Systems*, Vol. 32.

[18] Andrzej Bargiela and Witold Pedrycz. 2016. Granular computing. In *Handbook on computational intelligence: Fuzzy Logic, Systems, Artificial Neural Networks, and Learning Systems*. World Scientific, 43–66.

[19] Dawi Karomati Baroroh, Chih-Hsing Chu, and Lihui Wang. 2020. Systematic literature review on augmented reality in smart manufacturing: collaboration between human and computational intelligence. *Journal of Manufacturing Systems* (2020).

[20] Denis Baylor, Eric Breck, Heng-Tze Cheng, Noah Fiedel, Chuan Yu Foo, Zakaria Haque, Salem Haykal, Mustafa Ispir, Vihan Jain, Levent Koc, et al. 2017. Tfx: A tensorflow-based production-scale machine learning platform. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1387–1395.

[21] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, et al. 2019. AI Fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM Journal of Research and Development* 63, 4/5 (2019), 4–1.

[22] Ola Benderius, Christian Berger, and Victor Malmsten Lundgren. 2017. The best rated human–machine interface design for autonomous vehicles in the 2016 grand cooperative driving challenge. *IEEE Transactions on intelligent transportation systems* 19, 4 (2017), 1302–1307.

[23] Yoshua Bengio, Tristan Deleu, Nasim Rahaman, Rosemary Ke, Sebastien Lachapelle, Olexa Bilaniuk, Anirudh Goyal, and Christopher Pal. 2019. A meta-transfer objective for learning to disentangle causal mechanisms. In *Proceedings of the 7th International Conference on Learning Representations (ICLR 2019)*.

[24] Michael R Berthold, Nicolas Cebron, Fabian Dill, Thomas R Gabriel, Tobias Kotter, Thorsten Meinl, Peter Ohl, Kilian Thiel, and Bernd Wiswedel. 2009. KNIME-the Konstanz information miner: version 2.0 and beyond. *AcM SIGKDD Explorations Newsletter* 11, 1 (2009), 26–31.

[25] Tarek R Besold and Kai-Uwe Kuhnberger. 2015. Towards integrated neural–symbolic systems for human-level ai: Two research programs helping to bridge the gaps. *Biologically Inspired Cognitive Architectures* 14 (2015), 97–110.

[26] Tarek R Besold, Kai-Uwe Kuhnberger, Artur dÁvila Garcez, Alessandro Saffiotti, Martin H Fischer, and Alan Bundy. 2015. Anchoring knowledge in interaction: Towards a harmonic subsymbolic/symbolic framework and architecture of computational cognition. In *International Conference on Artificial General Intelligence*. Springer, 35–45.

[27] Felix Biessmann, David Salinas, Sebastian Schelter, Philipp Schmidt, and Dustin Lange. 2018. Deep Learning for Missing Value Imputation in Tables with Non-Numerical Data. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 2017–2025.

[28] Jack L Blackhurst, Jennifer S Gresham, and Morley O Stone. 2011. The autonomy paradox. *Armed Forces Journal* (2011), 20–24.

[29] Ken R Blawatt. 2016. Appendix A: List of Cognitive Biases. In *Marconomics*. Emerald Group Publishing Limited, 325–336.

[30] Razvan Gabriel Boboc, Florin Girbacia, and Eugen Valentin Butila. 2020. The application of augmented reality in the automotive industry: A systematic literature review. *Applied Sciences* 10, 12 (2020), 4259.

[31] Margaret Boden, Joanna Bryson, Darwin Caldwell, Kerstin Dautenhahn, Lilian Edwards, Sarah Kember, Paul Newman, Vivienne Parry, Geoff Pegman, Tom Rodden, et al. 2017. Principles of robotics: regulating robots in the real world. *Connection Science* 29, 2 (2017), 124–129.

[32] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, and et al. 2021. On the Opportunities and Risks of Foundation Models. *CoRR* abs/2108.07258 (2021). https://arxiv.org/abs/2108.07258

[33] Grady Booch, Francesco Fabiano, Lior Horesh, Kiran Kate, Jon Lenchner, Nick Linck, Andrea Loreggia, Keerthiram Murugesan, Nicholas Mattei, Francesca Rossi, and Biplav Srivastava. 2020. Thinking fast and slow in ai. *arXiv preprint arXiv:2010.06002* (2020).

[34] Matthew Botvinick, David GT Barrett, Peter Battaglia, Nando de Freitas, Darshan Kumaran, Joel Z Leibo, Timothy Lillicrap, Joseph Modayil, Shakir Mohamed, Neil C Rabinowitz, et al. 2017. Building machines that learn and think for themselves. *Behavioral and Brain Sciences* 40 (2017).

[35] Jeffrey M Bradshaw, Robert R Hoffman, David D Woods, and Matthew Johnson. 2013. The seven deadly myths of autonomous systems. *IEEE Intelligent Systems* 28, 3 (2013), 54–61.

[36] Daniel A Braun, Carsten Mehring, and Daniel M Wolpert. 2010. Structure learning in action. *Behavioural brain research* 206, 2 (2010), 157–165.

[37] M Breque, L De Nul, and A Petridis. 2021. Industry 5.0: towards a sustainable, human-centric and resilient European industry. *European Commission, Directorate-General for Research and Innovation* (2021).

[38] Rodney Brooks. 2019. A Better Lesson. https://rodneybrooks.com/a-better-lesson/

[39] Rodney Brooks. 2020. An Analogy For The State Of AI. https://rodneybrooks.com/an-analogy-for-the-state-of-ai/

[40] T. Brown, B. Mann, Nick Ryder, Melanie Subbiah, J. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, G. Kruger, T. Henighan, R. Child, Aditya Ramesh, D. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, E. Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, J. Clark, Christopher Berner, Sam McCandlish, A. Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS)*.

[41] Marcin Budka and Bogdan Gabrys. 2010. Ridge regression ensemble for toxicity prediction. *Procedia Computer Science* 1, 1 (2010), 193–201.

[42] Marcin Budka, Bogdan Gabrys, and Elisa Ravagnan. 2010. Robust predictive modelling of water pollution using biomarker data. *Water research* 44, 10 (2010), 3294–3308.

[43] Ruth MJ Byrne. 2019. Counterfactuals in Explainable Artificial Intelligence (XAI): Evidence from Human Reasoning.. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*. 6276–6282.

[44] T. Calders, E. Ntoutsi, M. Pechenizkiy, B. Rosenhahn, and S. Ruggieri. 2021. Call for Papers: Special issue on Bias and Fairness in AI. *Data Mining and Knowledge Discovery* (2021). https://www.springer.com/journal/10618/updates/19143254

[45] Murray Campbell, A Joseph Hoane Jr, and Feng-hsiung Hsu. 2002. Deep Blue. *Artificial Intelligence* 134, 1-2 (2002), 57–83.

[46] Yuanzhi Cao, Xun Qian, Tianyi Wang, Rachel Lee, Ke Huo, and Karthik Ramani. 2020. An Exploratory Study of Augmented Reality Presence for Tutoring Machine Tasks. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.

[47] Yuanzhi Cao, Tianyi Wang, Xun Qian, Pawan S Rao, Manav Wadhawan, Ke Huo, and Karthik Ramani. 2019. GhostAR: A time-space editor for embodied authoring of human-robot collaborative task with augmented reality. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 521–534.

[48] C Carreto, D Gêgo, and L Figueiredo. 2018. An eye-gaze tracking system for teleoperation of a mobile robot. *Journal of Information Systems Engineering & Management* 3, 2 (2018), 16.

[49] Dylan Cashman, Shah Rukh Humayoun, Florian Heimerl, Kendall Park, Subhajit Das, John Thompson, Bahador Saket, Abigail Mosca, John Stasko, Alex Endert, et al. 2019. A User-based Visual Analytics Workflow for Exploratory Model Analysis. *Computer Graphics Forum* 38, 3 (2019), 185–199.

[50] Simon Caton and Christian Haas. 2020. Fairness in machine learning: A survey. *arXiv preprint arXiv:2010.04053* (2020).

[51] Bilge Celik and Joaquin Vanschoren. 2021. Adaptation strategies for automated machine learning on evolving data. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[52] Girish Chandrashekar and Ferat Sahin. 2014. A survey on feature selection methods. *Computers & Electrical Engineering* 40, 1 (2014), 16–28.

[53] Pete Chapman, Julian Clinton, Randy Kerber, Thomas Khabaza, Thomas Reinartz, Colin Shearer, and Rudiger Wirth. 2000. *CRISP-DM 1.0: Step-by-step data mining guide.* SPSS.

[54] Min Chen and Amos Golan. 2015. What may visualization processes optimize? *IEEE transactions on visualization and computer graphics* 22, 12 (2015), 2619–2632.

[55] Zhiyuan Chen and Bing Liu. 2018. Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 12, 3 (2018), 1–207.

[56] Lu Cheng, Kush R Varshney, and Huan Liu. 2021. Socially Responsible AI Algorithms: Issues, Purposes, and Challenges. *arXiv preprint arXiv:2101.02032* (2021).

[57] Alexandra Chouldechova. 2017. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data* 5, 2 (2017), 153–163.

[58] Google Cloud. 2021. Explaining predictions. https://cloud.google.com/automl-tables/docs/explain

[59] Jacob W Crandall and Michael A Goodrich. 2002. Characterizing efficiency of human robot interaction: A case study of shared-control teleoperation. In *IEEE/RSJ international conference on intelligent robots and systems*, Vol. 2. IEEE, 1290–1295.

[60] Andrew Crotty, Alex Galakatos, Emanuel Zgraggen, Carsten Binnig, and Tim Kraska. 2015. Vizdom: interactive analytics through pen and touch. *Proceedings of the VLDB Endowment* 8, 12 (2015), 2024–2027.

[61] Wenlin Dai and Marc G Genton. 2018. Multivariate functional data visualization and outlier detection. *Journal of Computational and Graphical Statistics* 27, 4 (2018), 923–934.

[62] Robert Dale. 2021. GPT-3: What's it good for? *Natural Language Engineering* 27, 1 (2021), 113–118.

[63] Brian d'Alessandro, Cathy O'Neil, and Tom LaGatta. 2017. Conscientious classification: A data scientist's guide to discrimination-aware classification. *Big data* 5, 2 (2017), 120–134.

[64] DataRobot. 2021. Automated Machine Learning Platform. https://www.datarobot.com/platform/automated-machine-learning/ Accessed: 2021-09-24.

[65] Tilman Deuschel and Ted Scully. 2016. On the Importance of Spatial Perception for the Design of Adaptive User Interfaces. In *Proceedings of the 10th International Conference on Self-Adaptive and Self-Organizing Systems (SASO).* IEEE, 70–79.

[66] William Dieterich, Christina Mendoza, and Tim Brennan. 2016. COMPAS risk scales: Demonstrating accuracy equity and predictive parity. *Northpointe Inc* (2016).

[67] Virginia Dignum. 2017. Responsible Autonomy. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17.* 4698–4704.

[68] Virginia Dignum. 2018. Ethics in artificial intelligence: introduction to the special issue. *Ethics and Information Technology* 20, 1 (2018), 1–3.

[69] Pedro Domingos. 2012. A Few Useful Things to Know about Machine Learning. *Commun. ACM* 55, 10 (2012), 78–87.

[70] Xuanyi Dong, David Jacob Kedziora, Katarzyna Musial, and Bogdan Gabrys. 2021. Automated Deep Learning: Neural Architecture Search Is Not the End. *arXiv preprint arXiv:2112.09245* (2021).

[71] Hubert Dreyfus, Stuart E Dreyfus, and Tom Athanasiou. 2000. *Mind over machine.* Simon and Schuster.

[72] Iddo Drori, Yamuna Krishnamurthy, Remi Rampin, Raoni Lourenço, Jorge One, Kyunghyun Cho, Claudio Silva, and Juliana Freire. 2018. AlphaD3M: Machine learning pipeline synthesis. In *AutoML Workshop at ICML.*

[73] Jaimie Drozdal, Justin Weisz, Dakuo Wang, Gaurav Dass, Bingsheng Yao, Changruo Zhao, Michael Muller, Lin Ju, and Hui Su. 2020. Trust in AutoML: exploring information needs for establishing trust in automated machine learning systems. In *Proceedings of the 25th International Conference on Intelligent User Interfaces.* 297–307.

[74] Juan Manuel Duran and Karin Rolanda Jongsma. 2021. Who is afraid of black box algorithms? On the epistemological and ethical basis of trust in medical AI. *Journal of Medical Ethics* 47, 5 (2021), 329–335.

[75] Upol Ehsan, Brent Harrison, Larry Chan, and Mark O Riedl. 2018. Rationalization: A neural machine translation approach to generating natural language explanations. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society.* 81–87.

[76] Upol Ehsan, Q Vera Liao, Michael Muller, Mark O Riedl, and Justin D Weisz. 2021. Expanding explainability: Towards social transparency in AI systems. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems.* 1–19.

[77] Upol Ehsan, Pradyumna Tambwekar, Larry Chan, Brent Harrison, and Mark O Riedl. 2019. Automated rationale generation: a technique for explainable AI and its effects on human perceptions. In *Proceedings of the 24th International Conference on Intelligent User Interfaces.* 263–274.

[78] Christos Emmanouilidis, Petros Pistofidis, Luka Bertoncelj, Vassilis Katsouros, Apostolos Fournaris, Christos Koulamas, and Cristobal Ruiz-Carcel. 2019. Enabling the human in the loop: Linked data and knowledge in industrial cyber-physical systems. *Annual reviews in control* 47 (2019), 249–265.

[79] MR Endsley. 2011. Bringing cognitive engineering to the information fusion problem: Creating systems that understand situations. In *Plenary Presentation to the 14th International Conference on Information Fusion*.

[80] Mica R Endsley. 1995. Toward a theory of situation awareness in dynamic systems. *Human Factors* 37, 1 (1995), 32–64.

[81] Mica R Endsley. 2017. From here to autonomy: lessons learned from human–automation research. *Human factors* 59, 1 (2017), 5–27.

[82] Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander Smola. 2020. Autogluon-tabular: Robust and accurate automl for structured data. *arXiv preprint arXiv:2003.06505* (2020).

[83] Dikai Fang, Huahu Xu, Xiaoxian Yang, and Minjie Bian. 2020. An augmented reality-based method for remote collaborative real-time assistance: from a system perspective. *Mobile Networks and Applications* 25, 2 (2020), 412–425.

[84] William Fedus, Barret Zoph, and Noam Shazeer. 2021. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. *CoRR* abs/2101.03961 (2021). https://arxiv.org/abs/2101.03961

[85] Geli Fei, Shuai Wang, and Bing Liu. 2016. Learning cumulatively to become more knowledgeable. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1565–1574.

[86] J-D Fekete. 2004. The infovis toolkit. In *IEEE Symposium on Information Visualization*. 167–174.

[87] Matthias Feurer, Katharina Eggensperger, Stefan Falkner, Marius Lindauer, and Frank Hutter. 2021. Auto-Sklearn 2.0: The Next Generation. *CoRR* abs/2007.04074 (2021). https://arxiv.org/abs/2007.04074

[88] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. 2015. Efficient and Robust Automated Machine Learning. In *Advances in Neural Information Processing Systems*, Vol. 28.

[89] Matthias Feurer, Jost Springenberg, and Frank Hutter. 2015. Initializing bayesian hyperparameter optimization via meta-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 29. 1128–1135.

[90] Lukas Fischer, Lisa Ehrlinger, Verena Geist, Rudolf Ramler, Florian Sobiezky, Werner Zellinger, David Brunner, Mohit Kumar, and Bernhard Moser. 2021. AI System Engineering—Key Challenges and Lessons Learned. *Machine Learning and Knowledge Extraction* 3, 1 (2021), 56–83.

[91] Ragnar Fjelland. 2020. Why general artificial intelligence will not be realized. *Humanities and Social Sciences Communications* 7, 1 (2020), 1–9.

[92] Johannes Furnkranz. 2005. From local to global patterns: Evaluation issues in rule learning algorithms. In *Local pattern detection*. Springer, 20–38.

[93] Johannes Furnkranz, Tomas Kliegr, and Heiko Paulheim. 2020. On cognitive preferences and the plausibility of rule-based models. *Machine Learning* 109, 4 (2020), 853–898.

[94] Bogdan Gabrys. 2005. Do Smart Adaptive Systems Exist?—Introduction. In *Do Smart Adaptive Systems Exist?* Springer, 1–17.

[95] Bogdan Gabrys and Andrzej Bargiela. 2000. General fuzzy min-max neural network for clustering and classification. *IEEE transactions on neural networks* 11, 3 (2000), 769–783.

[96] Krzysztof Z Gajos, Katherine Everitt, Desney S Tan, Mary Czerwinski, and Daniel S Weld. 2008. Predictability and accuracy in adaptive user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1271–1274.

[97] Charles R Gallistel and Adam Philip King. 2011. *Memory and the computational brain: Why cognitive science will transform neuroscience*. Vol. 6. John Wiley & Sons.

[98] Joao Gama, Indre Zliobaite, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. 2014. A survey on concept drift adaptation. *ACM computing survey* 46, 4 (2014), 1–37.

[99] Artur dÁvila Garcez and Luis C Lamb. 2020. Neurosymbolic AI: The 3rd Wave. *arXiv preprint arXiv:2012.05876* (2020).

[100] Unai Garciarena, Roberto Santana, and Alexander Mendiburu. 2018. Analysis of the complexity of the automatic pipeline generation problem. In *Proccedings of the IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 1–8.

[101] Jason Gauci, Nathalie Cauchi, Kevin Theuma, David Zammit-Mangion, and Alan Muscat. 2015. Design and evaluation of a touch screen concept for pilot interaction with avionic systems. In *2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)*. IEEE, 3C2–1.

[102] Verena Geist, Michael Moser, Josef Pichler, Rodolfo Santos, and Volkmar Wieser. 2021. Leveraging machine learning for software redocumentation—A comprehensive comparison of methods in practice. *Software: Practice and Experience* 51, 4 (2021), 798–823.

[103] Efstathios D Gennatas, Jerome H Friedman, Lyle H Ungar, Romain Pirracchio, Eric Eaton, Lara G Reichmann, Yannet Interian, José Marcio Luna, Charles B Simone, Andrew Auerbach, et al. 2020. Expert-augmented machine learning. *Proceedings of the National Academy of Sciences* 117, 9 (2020), 4571–4577.

[104] Lise Getoor. 2019. Responsible Data Science. In *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 1–1.

[105] Yolanda Gil, James Honaker, Shikhar Gupta, Yibo Ma, Vito DÓrazio, Daniel Garijo, Shruti Gadewar, Qifan Yang, and Neda Jahanshad. 2019. Towards human-guided machine learning. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*. 614–624.

[106] Daniel Golovin, Benjamin Solnik, Subhodeep Moitra, Greg Kochanski, John Karro, and D Sculley. 2017. Google vizier: A service for black-box optimization. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 1487–1495.

[107] Dominic Gorecky, Mathias Schmitt, Matthias Loskyll, and Detlef Zuhlke. 2014. Human-machine-interaction in the industry 4.0 era. In *2014 12th IEEE international conference on industrial informatics (INDIN)*. IEEE, 289–294.

[108] Anirudh Goyal, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Scholkopf. 2019. Recurrent independent mechanisms. *arXiv preprint arXiv:1909.10893* (2019).

[109] Richard D Greenblatt, Donald E Eastlake III, and Stephen D Crocker. 1967. The Greenblatt chess program. In *Proceedings of the November 14-16, 1967, fall joint computer conference*. 801–810.

[110] Joshua Greene, Francesca Rossi, John Tasioulas, Kristen Brent Venable, and Brian Williams. 2016. Embedding ethical principles in collective decision support systems. In *Proceedings of the Thirtieth Conference on Artificial Intelligence (AAAI)*. 4147–4151.

[111] Giorgio Gronchi and Fabio Giovannelli. 2018. Dual Process Theory of Thought and Default Mode Network: A Possible Neural Foundation of Fast Thinking. *Frontiers in Psychology* 9 (2018), 1237.

[112] Serge Guillaume. 2001. Designing fuzzy inference systems from data: An interpretability-oriented review. *IEEE Transactions on fuzzy systems* 9, 3 (2001), 426–443.

[113] Xiaoxiao Guo, Satinder Singh, Honglak Lee, Richard L Lewis, and Xiaoshi Wang. 2014. Deep learning for real-time Atari game play using offline Monte-Carlo tree search planning. In *Advances in neural information processing systems*. 3338–3346.

[114] Patrik Gustavsson, Anna Syberfeldt, Rodney Brewster, and Lihui Wang. 2017. Human-robot collaboration demonstrator combining speech recognition and haptic control. *Procedia CIRP* 63 (2017), 396–401.

[115] Nicolin Hainc, Christian Federau, Bram Stieltjes, Maria Blatow, Andrea Bink, and Christoph Stippich. 2017. The Bright, Artificial Intelligence-Augmented Future of Neuroimaging Reading. *Frontiers in Neurology* 8 (2017), 489.

[116] Sara Hajian, Francesco Bonchi, and Carlos Castillo. 2016. Algorithmic bias: From discrimination discovery to fairness-aware data mining. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2125–2126.

[117] Yuval Noah Harari. 2014. *Sapiens: A brief history of humankind.* Random House.

[118] Michaela Hardt, Xiaoguang Chen, Xiaoyi Cheng, Michele Donini, Jason Gelman, Satish Gollaprolu, John He, Pedro Larroy, Xinyu Liu, Nick McCarthy, et al. 2021. Amazon SageMaker Clarify: Machine Learning Bias Detection and Explainability in the Cloud. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

[119] David Harel, Assaf Marron, and Joseph Sifakis. 2020. Autonomics: In search of a foundation for next-generation autonomous systems. *Proceedings of the National Academy of Sciences* 117, 30 (2020), 17491–17498.

[120] Christopher Harris. 2020. Mitigating Cognitive Biases in Machine Learning Algorithms for Decision Making. In *Proceedings of the Web Conference*. 775–781.

[121] Hugh Harvey. 2020. Why AI will not replace radiologists. *Towards Data Science* (2020). https://towardsdatascience.com/why-ai-will-not-replace-radiologists-c7736f2c7d80 Accessed on 25 September 2022).

[122] Andrew J. Hawkins. 2018. Inside Waymo's Strategy to Grow the Best Brains for Self-Driving Cars. https://www.theverge.com/2018/5/9/17307156/google-waymo-driverless-cars-deep-learning-neural-net-interview

[123] Jeffrey Heer. 2019. Agency plus automation: Designing artificial intelligence into interactive systems. *Proceedings of the National Academy of Sciences* 116, 6 (2019), 1844–1850.

[124] Frank Heyen, Tanja Munz, Michael Neumann, Daniel Ortega, Ngoc Thang Vu, Daniel Weiskopf, and Michael Sedlmair. 2020. ClaVis: An Interactive Visual Comparison System for Classifiers. In *Proceedings of the International Conference on Advanced Visual Interfaces*. 1–9.

[125] Markus Hofmann and Ralf Klinkenberg. 2016. *RapidMiner: Data mining use cases and business analytics applications.* CRC Press.

[126] Andreas Holzinger, Andre Carrington, and Heimo Muller. 2020. Measuring the quality of explanations: the system causability scale (SCS). *KI-Kunstliche Intelligenz* (2020), 1–6.

[127] Andreas Holzinger, Georg Langs, Helmut Denk, Kurt Zatloukal, and Heimo Muller. 2019. Causability and explainability of artificial intelligence in medicine. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 9, 4 (2019), e1312.

[128] Kristina Hook. 2000. Steps to take before intelligent user interfaces become real. *Interacting with computers* 12, 4 (2000), 409–426.

[129] Eric Horvitz. 1999. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. 159–166.

[130] Zhongshi Hu, Xin Xin, Wanting Xu, Yuwei Sun, Zhenyu Jiang, Xiangyu Wang, Yishan Liu, Siyao Lu, and Min Zhao. 2019. A Literature Review of the Research on Interaction Mode of Self-driving Cars. In *International Conference on Human-Computer Interaction*. Springer, 29–40.

[131] Elle Hunt. 2016. Tay, Microsoft's AI chatbot, gets a crash course in racism from Twitter. *The Guardian* 24, 3 (2016).

[132] John D Hunter. 2007. Matplotlib: A 2D graphics environment. *Computing in science & engineering* 9, 03 (2007), 90–95.

[133] Vinod Iyengar. 2019. New Innovations in Driverless AI. https://www.h2o.ai/blog/new-innovations-in-driverless-ai/

[134] Alon Jacovi, Ana Marasovic, Tim Miller, and Yoav Goldberg. 2021. Formalizing Trust in Artificial Intelligence: Prerequisites, Causes and Goals of Human Trust in AI. In *Proceedings of the Fourth Annual ACM FAccT Conference*.

[135] Anthony Jameson. 2007. Adaptive interfaces and agents. In *The human-computer interaction handbook*. CRC Press, 459–484.

[136] Michael I Jordan. 2019. Artificial intelligence—the revolution hasn't happened yet. *Harvard Data Science Review* 1, 1 (2019).

[137] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Zidek, Anna Potapenko, et al. 2021. Highly accurate protein structure prediction with AlphaFold. *Nature* 596, 7873 (2021), 583–589.

[138] John Just and Sambuddha Ghosal. 2019. Deep Generative Models Strike Back! Improving Understanding and Evaluation in Light of Unmet Expectations for OoD Data. *arXiv preprint arXiv:1911.04699* (2019).

[139] Petr Kadlec and Bogdan Gabrys. 2009. Architecture for development of adaptive on-line prediction models. *Memetic Computing* 1, 4 (2009), 241–269.

[140] Petr Kadlec and Bogdan Gabrys. 2009. Evolving on-line prediction model dealing with industrial data sets. In *2009 IEEE Workshop on Evolving and Self-Developing Intelligent Systems*. IEEE. https://doi.org/10.1109/esdis.2009.4938995

[141] Petr Kadlec and Bogdan Gabrys. 2009. Soft Sensor Based on Adaptive Local Learning. In *Advances in Neuro-Information Processing*. Springer Berlin Heidelberg, 1172–1179. https://doi.org/10.1007/978-3-642-02490-0_142

[142] Petr Kadlec and Bogdan Gabrys. 2009. Soft sensors: where are we and what are the current and future challenges? *IFAC Proceedings Volumes* 42, 19 (2009), 572–577. https://doi.org/10.3182/20090921-3-tr-3005.00098

[143] Petr Kadlec and Bogdan Gabrys. 2010. Adaptive on-line prediction soft sensing without historical data. In *Proceedings of the 2010 international joint conference on neural networks (IJCNN)*. IEEE, 1–8.

[144] Petr Kadlec and Bogdan Gabrys. 2011. Local learning-based adaptive soft sensor for catalyst activation prediction. *AIChE Journal* 57, 5 (2011), 1288–1301.

[145] Daniel Kahneman. 2011. *Thinking, fast and slow*. Macmillan.

[146] Faisal Kamiran, Sameen Mansha, Asim Karim, and Xiangliang Zhang. 2018. Exploiting reject option in classification for social discrimination control. *Information Sciences* 425 (2018), 18–33.

[147] Fariba Karimi, Mathieu Genois, Claudia Wagner, Philipp Singer, and Markus Strohmaier. 2018. Homophily influences ranking of minorities in social networks. *Scientific reports* 8, 1 (2018), 1–12.

[148] David Jacob Kedziora, Katarzyna Musial, and Bogdan Gabrys. 2020. AutonoML: Towards an Integrated Framework for Autonomous Machine Learning. *arXiv preprint arXiv:2012.12600* (2020).

[149] Charles Kemp, Amy Perfors, and Joshua B Tenenbaum. 2007. Learning overhypotheses with hierarchical Bayesian models. *Developmental science* 10, 3 (2007), 307–321.

[150] Thanh Tung Khuat, Fang Chen, and Bogdan Gabrys. 2021. An effective multi-resolution hierarchical granular representation based classifier using general fuzzy min-max neural network. *IEEE Transactions on Fuzzy Systems* 29, 2 (2021), 427–441.

[151] Thanh Tung Khuat, Dymitr Ruta, and Bogdan Gabrys. 2021. Hyperbox-based machine learning algorithms: a comprehensive survey. *Soft Computing* 25, 2 (2021), 1325–1363.

[152] John KC Kingston. 2016. Artificial intelligence and legal liability. In *Proceedings of the International conference on innovative techniques and applications of artificial intelligence*. Springer, 269–279.

[153] Rob Kitchin. 2017. Thinking critically about and researching algorithms. *Information, Communication & Society* 20, 1 (2017), 14–29.

[154] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. 2017. Inherent Trade-Offs in the Fair Determination of Risk Scores. In *Proceedings of the 8th Innovations in Theoretical Computer Science Conference*, Vol. 67. 43:1–43:23.

[155] Tomas Kliegr, Stepan Bahnik, and Johannes Furnkranz. 2018. A review of possible effects of cognitive biases on interpretation of rule-based machine learning models. *arXiv preprint arXiv:1804.02969* (2018).

[156] Glen Klien, David D Woods, Jeffrey M Bradshaw, Robert R Hoffman, and Paul J Feltovich. 2004. Ten challenges for making automation a" team player" in joint human-agent activity. *IEEE Intelligent Systems* 19, 6 (2004), 91–95.

[157] Mieczyslaw M Kokar and Mica R Endsley. 2012. Situation awareness and cognitive modeling. *IEEE Intelligent Systems* 27, 3 (2012), 91–96.

[158] J Korteling, G Van de Boer-Visschedijk, R Blankendaal, Rudy Boonekamp, and Aletta Eikelboom. 2021. Human-versus Artificial Intelligence. *Frontiers in Artificial Intelligence* 4 (2021).

[159] Tim Kraska. 2018. Northstar: an interactive data science system. *Proceedings of the VLDB Endowment* 11, 12 (2018), 2150–2164.

[160] Todd Kulesza, Margaret Burnett, Weng-Keen Wong, and Simone Stumpf. 2015. Principles of explanatory debugging to personalize interactive machine learning. In *Proceedings of the 20th international conference on intelligent user interfaces*. 126–137.

[161] Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. 2017. Counterfactual Fairness. In *Advances in Neural Information Processing Systems*, Vol. 30.

[162] John Laird and Shiwali Mohan. 2018. Learning fast and slow: Levels of learning in general autonomous intelligent agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32. 7983–7987.

[163] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. 2015. Human-level concept learning through probabilistic program induction. *Science* 350, 6266 (2015), 1332–1338.

[164] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. 2017. Building machines that learn and think like people. *Behavioral and brain sciences* 40 (2017).

[165] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. 2017. Ingredients of intelligence: From classic debates to an engineering roadmap. *Behavioral and Brain Sciences* 40 (2017).

[166] H. Chad Lane. 2012. *Cognitive Models of Learning*. Springer, 608–610.

[167] Mathias Lechner, Ramin Hasani, Alexander Amini, Thomas A Henzinger, Daniela Rus, and Radu Grosu. 2020. Neural circuit policies enabling auditable autonomy. *Nature Machine Intelligence* 2, 10 (2020), 642–652.

[168] Doris Jung-Lin Lee, Stephen Macke, Doris Xin, Angela Lee, Silu Huang, and Aditya Parameswaran. 2019. A Human-in-the-loop Perspective on AutoML: Milestones and the Road Ahead. *IEEE Data Engineering Bulletin* 42, 2 (2019), 59–70.

[169] Jaedong Lee, Changhyeon Lee, and Gerard Jounghyun Kim. 2017. Vouch: multimodal touch-and-voice input for smart watches under difficult operating conditions. *Journal on Multimodal User Interfaces* 11, 3 (2017), 289–299.

[170] Min Kyung Lee. 2018. Understanding perception of algorithmic decisions: Fairness, trust, and emotion in response to algorithmic management. *Big Data & Society* 5, 1 (2018), 1–16.

[171] Min Kyung Lee, Daniel Kusbit, Anson Kahng, Ji Tae Kim, Xinran Yuan, Allissa Chan, Daniel See, Ritesh Noothigattu, Siheon Lee, Alexandros Psomas, and Ariel D. Procaccia. 2019. WeBuildAI: Participatory Framework for Algorithmic Governance. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW (2019), 1–35.

[172] Christiane Lemke, Marcin Budka, and Bogdan Gabrys. 2015. Metalearning: a survey of trends and technologies. *Artificial Intelligence Review* 44, 1 (2015), 117–130. https://doi.org/10.1007/s10462-013-9406-y

[173] Christiane Lemke and Bogdan Gabrys. 2010. Meta-learning for time series forecasting and forecast combination. *Neurocomputing* 73, 10-12 (2010), 2006–2016. https://doi.org/10.1016/j.neucom.2009.09.020

[174] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P. Trevino, Jiliang Tang, and Huan Liu. 2017. Feature Selection: A Data Perspective. *Comput. Surveys* 50, 6 (2017), 1–45.

[175] Chin-Teng Lin and Tien-Thong Nguyen Do. 2021. Direct-sense brain–computer interfaces and wearable computers. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 51, 1 (2021), 298–312.

[176] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. 2021. Explainable ai: A review of machine learning interpretability methods. *Entropy* 23, 1 (2021), 18.

[177] Richard Lippmann, William Campbell, and Joseph Campbell. 2016. An Overview of the DARPA Data Driven Discovery of Models (D3M) Program. In *NIPS Workshop on Artificial Intelligence for Data Science*.

[178] Peter Lipton. 1990. Contrastive explanation. *Royal Institute of Philosophy Supplements* 27 (1990), 247–266.

[179] Michael L. Littman, Ifeoma Ajunwa, Guy Berger, Craig Boutilier, Morgan Currie, Finale Doshi-Velez, Gillian Hadfield, Michael C. Horowitz, Charles Isbell, Hiroaki Kitano, Karen Levy, Terah Lyons, Melanie Mitchell, Julie Shah, Steven Sloman, Shannon Vallor, and Toby Walsh. 2021. *Gathering Strength, Gathering Storms: The One Hundred Year Study on Artificial Intelligence (AI100) 2021 Study Panel Report*. Technical Report. Stanford University, Stanford, CA. 1–82 pages. http://ai100.stanford.edu/2021-report

[180] Shixia Liu, Xiting Wang, Mengchen Liu, and Jun Zhu. 2017. Towards better analysis of machine learning models: A visual analytics perspective. *Visual Informatics* 1, 1 (2017), 48–56.

[181] Yue Liu, Wenjie Tian, and Shuang Li. 2021. Meta-data Augmentation Based Search Strategy Through Generative Adversarial Network for AutoML Model Selection.. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*. Springer, 312–324.

[182] Luca Longo. 2016. Argumentation for knowledge representation, conflict resolution, defeasible inference and its integration with machine learning. In *Machine Learning for Health Informatics*. Springer, 183–208.

[183] Yuqian Lu, Juvenal Sastre Adrados, Saahil Shivneel Chand, and Lihui Wang. 2021. Humans Are Not Machines—Anthropocentric Human–Machine Symbiosis for Ultra-Flexible Smart Manufacturing. *Engineering* 7, 6

(2021), 734–737.

[184] Saifuddin Mahmud, Xiangxu Lin, and Jong-Hoon Kim. 2020. Interface for Human Machine Interaction for assistant devices: a review. In *Proccedings of the 10th Annual Computing and Communication Workshop and Conference*. IEEE, 768–773.

[185] Yaoli Mao, Dakuo Wang, Michael Muller, Kush R Varshney, Ioana Baldini, Casey Dugan, and Aleksandra Mojsilovic. 2019. How data scientists work together with domain experts in scientific collaborations: To find the right answer or to ask the right question? *Proceedings of the ACM on Human-Computer Interaction* 3 (2019), 1–23.

[186] Gary Marcus. 2020. The next decade in AI: four steps towards robust artificial intelligence. *arXiv preprint arXiv:2002.06177* (2020).

[187] Gary Marcus and Ernest Davis. 2019. *Rebooting AI: Building artificial intelligence we can trust.* Vintage.

[188] Gary Marcus and Ernest Davis. 2021. Has AI found a new Foundation? *The Gradient* (2021). https://thegradient.pub/has-ai-found-a-new-foundation

[189] Gary F Marcus. 2019. *The algebraic mind: Integrating connectionism and cognitive science.* MIT press.

[190] George Margetis, Stavroula Ntoa, Margherita Antona, and Constantine Stephanidis. 2021. Human-Centered Design of Artificial Intelligence. *Handbook of Human Factors and Ergonomics* (2021), 1085–1106.

[191] John McCarthy. 2007. From here to human-level AI. *Artificial Intelligence* 171, 18 (2007), 1174–1182.

[192] Michael McGough. 2018. How bad is sacramento's air, exactly? Google results appear at odds with reality, some say. *Sacramento Bee* (2018). https://www.sacbee.com/news/state/california/fires/article216227775.html

[193] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A Survey on Bias and Fairness in Machine Learning. 54, 6 (2021), 1—-35.

[194] Manuel Mendoza, Miguel Mendoza, Eloy Mendoza, and Eduardo Gonzalez. 2015. Augmented reality as a tool of training for data collection on torque auditing. *Procedia Computer Science* 75 (2015), 5–11.

[195] Michael Mertens and Herman J Damveld. 2012. An avionics touch screen-based control display concept. In *Head-and Helmet-Mounted Displays XVII; and Display Technologies and Applications for Defense, Security, and Avionics VI*, Vol. 8383. International Society for Optics and Photonics, 83830L.

[196] Bertalan Mesko. 2019. The real era of the art of medicine begins with artificial intelligence. *Journal of medical Internet research* 21, 11 (2019), e16295.

[197] Tomas Mikolov, Armand Joulin, and Marco Baroni. 2016. A roadmap towards machine intelligence. In *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 29–61.

[198] James D Miller, Roman Yampolskiy, Olle Haggstrom, and Stuart Armstrong. 2020. Chess as a Testing Grounds for the Oracle Approach to AI Safety. *arXiv preprint arXiv:2010.02911* (2020).

[199] Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence* 267 (2019), 1–38.

[200] Christoph Molnar. 2020. *Interpretable machine learning*. Lulu. com.

[201] Mike Moore. 2018. What is Industry 4.0? Everything you need to know. *World of Tech* (2018).

[202] Katharina Muelling, Arun Venkatraman, Jean-Sebastien Valois, John E Downey, Jeffrey Weiss, Shervin Javdani, Martial Hebert, Andrew B Schwartz, Jennifer L Collinger, and J Andrew Bagnell. 2017. Autonomy infused teleoperation with application to brain computer interface controlled manipulation. *Autonomous Robots* 41, 6 (2017), 1401–1422.

[203] J Müller. 2020. Enabling technologies for industry 5.0, results of a workshop with europe's technology leaders. *Directorate-General for Research and Innovation* (2020).

[204] Jared S Murray. 2018. Multiple imputation: a review of practical and theoretical findings. *Statist. Sci.* 33, 2 (2018), 142–159.

[205] Luigi Nardi, David Koeplinger, and Kunle Olukotun. 2019. Practical Design Space Exploration. In *Proceedings of the 27th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*. 347–358. https://doi.org/10.1109/MASCOTS.2019.00045

[206] Allen Newell. 1980. Physical symbol systems. *Cognitive science* 4, 2 (1980), 135–183.

[207] Allen Newell, Herbert Alexander Simon, et al. 1972. *Human problem solving*. Vol. 104. Prentice-hall Englewood Cliffs, NJ.

[208] Tien-Dung Nguyen, David Jacob Kedziora, Katarzyna Musial, and Bogdan Gabrys. 2021. Exploring Opportunistic Meta-knowledge to Reduce Search Spaces for Automated Machine Learning. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*. 1–8. https://doi.org/10.1109/IJCNN52387.2021.9533431

[209] Tien-Dung Nguyen, Katarzyna Musial, and Bogdan Gabrys. 2021. AutoWeka4MCPS-AVATAR: Accelerating automated machine learning pipeline composition and optimisation. *Expert Systems with Applications* 185 (2021), 115643. https://doi.org/10.1016/j.eswa.2021.115643

[210] Jack Nicas, Natalie Kitroeff, David Gelles, and James Glanz. 2019. Boeing built deadly assumptions into 737 Max, blind to a late design change. *The New York Times* (2019). https://www.nytimes.com/2019/06/01/business/boeing-737-max-crash.html

[211] Chris North. 2006. Toward measuring visualization insight. *IEEE computer graphics and applications* 26, 3 (2006), 6–9.

[212] Eirini Ntoutsi, Pavlos Fafalios, Ujwal Gadiraju, Vasileios Iosifidis, Wolfgang Nejdl, Maria-Esther Vidal, Salvatore Ruggieri, Franco Turini, Symeon Papadopoulos, Emmanouil Krasanakis, Ioannis Kompatsiaris, Katharina Kinder-Kurlanda, Claudia Wagner, Fariba Karimi, Miriam Fernandez, Harith Alani, Bettina Berendt, Tina Kruegel, Christian Heinze, Klaus Broelemann, Gjergji Kasneci, Thanassis Tiropanis, and Steffen Staab. 2020. Bias in data-driven artificial intelligence systems—An introductory survey. *WIREs Data Mining and Knowledge Discovery* 10, 3 (2020), e1356.

[213] Haydemar Nunez, Cecilio Angulo, and Andreu Catala. 2002. Rule extraction from support vector machines.. In *Proceedings of the European Symposium on Artificial Neural Networks*. 107–112.

[214] Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. 2019. Dissecting racial bias in an algorithm used to manage the health of populations. *Science* 366, 6464 (2019), 447–453.

[215] Jorge Piazentin Ono, Sonia Castelo, Roque Lopez, Enrico Bertini, Juliana Freire, and Claudio Silva. 2021. Pipeline-Profiler: A Visual Analytics Tool for the Exploration of AutoML Pipelines. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2021), 390–400.

[216] Laurent Orseau and Stuart Armstrong. 2016. Safely Interruptible Agents. (2016), 557–566.

[217] Vasile Palade, Daniel-Ciprian Neagu, and Ron J Patton. 2001. Interpretation of trained neural networks by rule extraction. In *International Conference on Computational Intelligence*. Springer, 152–161.

[218] Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22, 10 (2009), 1345–1359.

[219] Raja Parasuraman, Thomas B Sheridan, and Christopher D Wickens. 2000. A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 30, 3 (2000), 286–297.

[220] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. 2019. Continual lifelong learning with neural networks: A review. *Neural Networks* 113 (2019), 54–71.

[221] Heungseok Park, Jinwoong Kim, Minkyu Kim, Ji-Hoon Kim, Jaegul Choo, Jung-Woo Ha, and Nako Sung. 2019. VisualHyperTuner: Visual analytics for user-driven hyperparameter tuning of deep neural networks. In *Demo at SysML Conference*.

[222] Judea Pearl. 2019. The seven tools of causal inference, with reflections on machine learning. *Commun. ACM* 62, 3 (2019), 54–60.

[223] Judea Pearl and Dana Mackenzie. 2018. *The book of why: the new science of cause and effect*. Basic books.

[224] Irit Peled and Orit Zaslavsky. 1997. Counter-Examples That (Only) Prove and Counter-Examples That (Also) Explain. *FOCUS on Learning Problems in mathematics* 19, 3 (1997), 49–61.

[225] Adam Perer and Ben Shneiderman. 2008. Integrating statistics and visualization: case studies of gaining clarity during exploratory data analysis. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. 265–274.

[226] Michael Polanyi. 2012. *Personal knowledge*. Routledge.

[227] Benjamin Powers. 2018. Once more with feeling: Teaching empathy to machines. *The Wall Street Journal* (2018). https://www.wsj.com/articles/once-more-with-feeling-teaching-empathy-to-machines-11544713141

[228] Anne Preisz. 2019. Fast and slow thinking; and the problem of conflating clinical reasoning and ethical deliberation in acute decision-making. *Journal of paediatrics and child health* 55, 6 (2019), 621–624.

[229] Kathy Pretz. 2021. Stop Calling Everything AI, Machine-Learning Pioneer Says. https://spectrum.ieee.org/stop-calling-everything-ai-machinelearning-pioneer-says

[230] Caroline Proctor and Woo-Kyoung Ahn. 2007. The effect of causal knowledge on judgments of the likelihood of unknown features. *Psychonomic Bulletin & Review* 14, 4 (2007), 635–639.

[231] Camilo Perez Quintero, Romeo Tatsambon Fomena, Azad Shademan, Oscar Ramirez, and Martin Jagersand. 2014. Interactive teleoperation interface for semi-autonomous control of robot arms. In *Proceedings of the Canadian Conference on Computer and Robot Vision*. IEEE, 357–363.

[232] Emilee Rader, Kelley Cotter, and Janghee Cho. 2018. Explanations as mechanisms for supporting algorithmic transparency. In *Proceedings of the 2018 CHI conference on human factors in computing systems*. 1–13.

[233] Jagdish Lal Raheja, Radhey Shyam, Umesh Kumar, and P Bhanu Prasad. 2010. Real-time robotic hand control using hand gestures. In *Proceedings of the Second International Conference on Machine Learning and Computing*. IEEE, 12–16.

[234] Iyad Rahwan. 2018. Society-in-the-loop: programming the algorithmic social contract. *Ethics and Information Technology* 20, 1 (2018), 5–14.

[235] Kaivalya Rawal and Himabindu Lakkaraju. 2020. Beyond Individualized Recourse: Interpretable and Interactive Summaries of Actionable Recourses. *Advances in Neural Information Processing Systems* 33 (2020).

[236] Bob Rehder. 2003. A causal-model theory of conceptual representation and categorization. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 29, 6 (2003), 1141.

[237] M Rhodes. 2016. So. algorithms are designing chairs now. *Wired* (2016). https://www.wired.com/2016/10/elbo-chair-autodesk-algorithm/

[238] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 1135–1144.

[239] M. Riedl. 2016. Big red button. https://markriedl.github.io/big-red-button/

[240] Tânia Rocha, Diana Carvalho, Maximino Bessa, Sofia Reis, and Luís Magalhães. 2017. Usability evaluation of navigation tasks by people with intellectual disabilities: a Google and SAPO comparative study regarding different interaction modalities. *Universal Access in the Information Society* 16, 3 (2017), 581–592.

[241] Francesca Rossi and Andrea Loreggia. 2019. Preferences and Ethical Priorities: Thinking Fast and Slow in AI. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. 3–4.

[242] Francesca Rossi and Nicholas Mattei. 2019. Building ethically bounded AI. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 9785–9789.

[243] Wilfred Rouwhorst, Ronald Verhoeven, Marieke Suijkerbuijk, Tanja Bos, Anneloes Maij, Mick Vermaat, and Roy Arents. 2017. Use of touch screen display applications for aircraft flight control. In *2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*. IEEE, 1–10.

[244] Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* 1, 5 (2019), 206–215. https://doi.org/10.1038/s42256-019-0048-x

[245] Charlotte Ruhl. 2021. Cognitive bias examples. *Simply Psychology* (2021). www.simplypsychology.org/cognitive-bias.html

[246] Dymitr Ruta and Bogdan Gabrys. 2005. Classifier selection for majority voting. *Information Fusion* 6, 1 (2005), 63–81. https://doi.org/10.1016/j.inffus.2004.04.008

[247] Christoph Sager, Christian Janiesch, and Patrick Zschech. 2021. A survey of image labelling for computer vision applications. *Journal of Business Analytics* (2021), 1–20.

[248] Manuel Martin Salvador, Marcin Budka, and Bogdan Gabrys. 2016. Adapting multicomponent predictive systems using hybrid adaptation strategies with auto-weka in process industry. In *Workshop on Automatic Machine Learning*. PMLR, 48–57.

[249] Manuel Martin Salvador, Marcin Budka, and Bogdan Gabrys. 2016. Towards automatic composition of multicomponent predictive systems. In *International conference on hybrid artificial intelligence systems*. Springer, 27–39.

[250] Manuel Martin Salvador, Marcin Budka, and Bogdan Gabrys. 2017. Modelling Multi-Component Predictive Systems as Petri Nets. In *Proceedings of the 15th Annual Industrial Simulation Conference*. 17–23.

[251] Manuel Martin Salvador, Marcin Budka, and Bogdan Gabrys. 2018. Automatic composition and optimization of multicomponent predictive systems with an extended auto-WEKA. *IEEE Transactions on Automation Science and Engineering* 16, 2 (2018), 946–959.

[252] Wojciech Samek, Gregoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Muller. 2019. *Explainable AI: interpreting, explaining and visualizing deep learning*. Vol. 11700. Springer Nature.

[253] Gustavo Saposnik, Donald Redelmeier, Christian C Ruff, and Philippe N Tobler. 2016. Cognitive biases associated with medical decisions: a systematic review. *BMC medical informatics and decision making* 16, 1 (2016), 1–14.

[254] Nripsuta Ani Saxena, Karen Huang, Evan DeFilippis, Goran Radanovic, David C. Parkes, and Yang Liu. 2019. How Do Fairness Definitions Fare? Examining Public Attitudes Towards Algorithmic Definitions of Fairness. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. 99–106.

[255] Sebastian Schelter, Felix Biessmann, Dustin Lange, Tammo Rukat, Phillipp Schmidt, Stephan Seufert, Pierre Brunelle, and Andrey Taptunov. 2019. Unit testing data with deequ. In *Proceedings of the 2019 International Conference on Management of Data*. 1993–1996.

[256] Philipp Schmidt and Felix Biessmann. 2019. Quantifying interpretability and trust in machine learning systems. *arXiv preprint arXiv:1901.08558* (2019).

[257] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. 2020. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature* 588, 7839 (2020), 604–609.

[258] Alexander Scriven, David Jacob Kedziora, Katarzyna Musial, and Bogdan Gabrys. 2021. The Technological Emergence of AutoML: A Survey of Performant Software and Applications in the Context of Industry. *Preprint* (2021).

[259] Floarea Serban, Joaquin Vanschoren, Jörg-Uwe Kietz, and Abraham Bernstein. 2013. A survey of intelligent assistants for data analysis. *Comput. Surveys* 45, 3 (2013), 1–35.

[260] Rita Sevastjanova, Fabian Beck, Basil Ell, Cagatay Turkay, Rafael Henkin, Miriam Butt, Daniel A Keim, and Mennatallah El-Assady. 2018. Going beyond visualization: Verbalization as complementary medium to explain machine learning models. In *Workshop on Visualization for AI Explainability at IEEE VIS*.

[261] Zeyuan Shang, Emanuel Zgraggen, Benedetto Buratti, Ferdinand Kossmann, Philipp Eichmann, Yeounoh Chung, Carsten Binnig, Eli Upfal, and Tim Kraska. 2019. Democratizing data science through interactive curation of ML pipelines. In *Proceedings of the 2019 International Conference on Management of Data*. 1171–1188.

[262] Thomas B Sheridan and William L Verplank. 1978. *Human and computer control of undersea teleoperators.* Technical Report. Massachusetts Inst of Tech Cambridge Man-Machine Systems Lab.

[263] Donghee Shin. 2020. How do users interact with algorithm recommender systems? The interaction of users, algorithms, and performance. *Computers in Human Behavior* 109 (2020), 106344.

[264] Donghee Shin. 2021. The effects of explainability and causability on perception, trust, and acceptance: Implications for explainable AI. *International Journal of Human-Computer Studies* 146 (2021), 102551.

[265] Ben Shneiderman. 2020. Design lessons from AI's two grand goals: Human emulation and useful applications. *IEEE Transactions on Technology and Society* 1, 2 (2020), 73–82.

[266] Ben Shneiderman. 2020. Human-centered artificial intelligence: Reliable, safe & trustworthy. *International Journal of Human–Computer Interaction* 36, 6 (2020), 495–504.

[267] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature* 529, 7587 (2016), 484–489.

[268] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. 2018. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science* 362, 6419 (2018), 1140–1144.

[269] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. 2017. Mastering the game of go without human knowledge. *Nature* 550, 7676 (2017), 354–359.

[270] Monika Simmler and Ruth Frischknecht. 2021. A taxonomy of human-machine collaboration: capturing automation and technical autonomy. *AI & SOCIETY* 36, 1 (2021), 239–250.

[271] Konstantinos Sipsas, Kosmas Alexopoulos, Vangelis Xanthakis, and George Chryssolouris. 2016. Collaborative maintenance in flow-line manufacturing environments: An Industry 4.0 approach. *Procedia Cirp* 55 (2016), 236–241.

[272] Steven Sloman. 2014. Two systems of reasoning: An update. In *Dual-process theories of the social mind*, J. W. Sherman, B. Gawronski, and Y. Trope (Eds.). The Guilford Press, Chapter 5, 69–79.

[273] Anthia Solaki, Francesco Berto, and Sonja Smets. 2019. The logic of fast and slow thinking. *Erkenntnis* (2019), 1–30.

[274] Thilo Spinner, Udo Schlegel, Hanna Schäfer, and Mennatallah El-Assady. 2020. explAIner: A visual analytics framework for interactive and explainable machine learning. *IEEE transactions on visualization and computer graphics* 26, 1 (2020), 1064–1074.

[275] Frederic Stahl, Bogdan Gabrys, Mohamed Medhat Gaber, and Monika Berendsen. 2013. An overview of interactive visual data mining techniques for knowledge discovery. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 3, 4 (2013), 239–256.

[276] Scott Stephenson. 2020. Deepgram Pioneers Novel Training Approach Setting New Standard for AI Companies. https://deepgram.com/blog/deepgram-pioneers-novel-training-approach-setting-new-standard-for-ai-companies-2

[277] Barry Strauch. 2017. Ironies of automation: Still unresolved after all these years. *IEEE Transactions on Human-Machine Systems* 48, 5 (2017), 419–433.

[278] Eliza Strickland. 2021. The Turbulent Past and Uncertain Future of Artificial Intelligence. *IEEE Spectrum* (2021). https://spectrum.ieee.org/history-of-ai Accessed on 25 September 2021).

[279] Hendrik Strobelt, Sebastian Gehrmann, Hanspeter Pfister, and Alexander M Rush. 2017. Lstmvis: A tool for visual analysis of hidden state dynamics in recurrent neural networks. *IEEE transactions on visualization and computer graphics* 24, 1 (2017), 667–676.

[280] Stefan Studer, Thanh Binh Bui, Christian Drescher, Alexander Hanuschkin, Ludwig Winkler, Steven Peters, and Klaus-Robert Müller. 2021. Towards CRISP-ML (Q): a machine learning process model with quality assurance methodology. *Machine Learning and Knowledge Extraction* 3, 2 (2021), 392–413.

[281] Shiliang Sun. 2013. A survey of multi-view machine learning. *Neural computing and applications* 23, 7 (2013), 2031–2038.

[282] Shiliang Sun, Honglei Shi, and Yuanbin Wu. 2015. A survey of multi-source domain adaptation. *Information Fusion* 24 (2015), 84–92.

[283] Richard Sutton. 2019. The bitter lesson. http://www.incompleteideas.net/IncIdeas/BitterLesson.html

[284] Thomas Swearingen, Will Drevo, Bennett Cyphers, Alfredo Cuesta-Infante, Arun Ross, and Kalyan Veeramachaneni. 2017. ATM: A distributed, collaborative, scalable system for automated machine learning. In *Proceedings of the IEEE International Conference on Big Data (Big Data)*. IEEE, 151–162.

[285] Gary KL Tam, Vivek Kothari, and Min Chen. 2016. An analysis of machine- and human-analytics in classification. *IEEE transactions on visualization and computer graphics* 23, 1 (2016), 71–80.

[286] Tadahiro Taniguchi, Emre Ugur, Matej Hoffmann, Lorenzo Jamone, Takayuki Nagai, Benjamin Rosman, Toshihiko Matsuka, Naoto Iwahashi, Erhan Oztop, Justus Piater, et al. 2018. Symbol emergence in cognitive developmental systems: a survey. *IEEE transactions on Cognitive and Developmental Systems* 11, 4 (2018), 494–516.

[287] Dusan Tatic and Bojan Tesic. 2017. The application of augmented reality technologies for the improvement of occupational safety in an industrial environment. *Computers in Industry* 85 (2017), 1–10.

[288] Robert M Taylor and John Reising. 1995. The Human-Electronic Crew: Can We Trust The Team?

[289] Joshua B Tenenbaum, Charles Kemp, Thomas L Griffiths, and Noah D Goodman. 2011. How to grow a mind: Statistics, structure, and abstraction. *science* 331, 6022 (2011), 1279–1285.

[290] Neil C. Thompson, Kristjan Greenewald, Keeheon Lee, and Gabriel F. Manso. 2021. Deep Learning's Diminishing Returns. *IEEE Spectrum* (2021). https://spectrum.ieee.org/deep-learning-computational-cost Accessed on 25 September 2021).

[291] Chris Thornton, Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. 2013. Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 847–855.

[292] Richard Tomlinson. 2020. Introducing MLOps Champion/Challenger Models. https://www.datarobot.com/blog/introducing-mlops-champion-challenger-models/

[293] Son N Tran and Artur S dÁvila Garcez. 2016. Deep logic networks: Inserting and extracting knowledge from deep belief networks. *IEEE transactions on neural networks and learning systems* 29, 2 (2016), 246–258.

[294] Pedro Tsividis, Joshua B Tenenbaum, and Laura Schulz. 2015. Constraints on hypothesis selection in causal learning. In *Proceedings of the 37th Annual Cognitive Science Society*. 2434–439.

[295] Amos Tversky and Daniel Kahneman. 1974. Judgment under Uncertainty: Heuristics and Biases: Biases in judgments reveal some heuristics of thinking under uncertainty. *Science* 185, 4157 (1974), 1124–1131.

[296] Brian E Ulicny, Jakub J Moskal, Mieczyslaw M Kokar, Keith Abe, and John Kei Smith. 2014. Inference and ontologies. In *Cyber Defense and Situational Awareness*. Springer, 167–199.

[297] Ehsan Upol. 2021. Towards Human-Centered Explainable AI: the journey so far. https://thegradient.pub/human-centered-explainable-ai/. *The Gradient* (2021).

[298] Saurabh Vaidya, Prashant Ambad, and Santosh Bhosle. 2018. Industry 4.0–a glimpse. *Procedia Manufacturing* 20 (2018), 233–238.

[299] Leslie G. Valiant. 2003. Three Problems in Computer Science. *J. ACM* 50, 1 (2003), 96–99.

[300] Jordi Vallverdu. 2020. Approximate and Situated Causality in Deep Learning. *Philosophies* 5, 1 (2020), 2:1–2:12.

[301] Peter Vamplew, Richard Dazeley, Cameron Foale, Sally Firmin, and Jane Mummery. 2018. Human-aligned artificial intelligence is a multiobjective problem. *Ethics and Information Technology* 20, 1 (2018), 27–40.

[302] Jasper Van der Waa, Elisabeth Nieuwburg, Anita Cremers, and Mark Neerincx. 2021. Evaluating XAI: A comparison of rule-based and example-based explanations. *Artificial Intelligence* 291 (2021), 103404.

[303] Kurt VanLehn. 1999. Rule-learning events in the acquisition of a complex skill: An evaluation of CASCADE. *The Journal of the Learning Sciences* 8, 1 (1999), 71–125.

[304] Joaquin Vanschoren. 2011. Meta-Learning Architectures: Collecting, Organizing and Exploiting Meta-Knowledge. In *Studies in Computational Intelligence*. Springer Berlin Heidelberg, 117–155.

[305] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, Vol. 30.

[306] Neil Vaughan, Bodgan Gabrys, and Venketesh N Dubey. 2016. An overview of self-adaptive technologies within virtual reality training. *Computer Science Review* 22 (2016), 65–87.

[307] Sahil Verma and Julia Rubin. 2018. Fairness definitions explained. In *IEEE/ACM international workshop on software fairness (fairware)*. IEEE, 1–7.

[308] Giulia Vilone and Luca Longo. 2020. Explainable artificial intelligence: a systematic review. *arXiv preprint arXiv:2006.00093* (2020).

[309] Marcel Walch, Lorenz Jaksche, Philipp Hock, Martin Baumann, and Michael Weber. 2017. Touch screen maneuver approval mechanisms for highly automated vehicles: A first evaluation. In *Proceedings of the 9th International Conference on Automotive User Interfaces and Interactive Vehicular Applications Adjunct*. 206–211.

[310] Dakuo Wang, Elizabeth Churchill, Pattie Maes, Xiangmin Fan, Ben Shneiderman, Yuanchun Shi, and Qianying Wang. 2020. From Human-Human Collaboration to Human-AI Collaboration: Designing AI Systems That Can Work Together with People. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–6.

[311] Dakuo Wang, Justin D. Weisz, Michael Muller, Parikshit Ram, Werner Geyer, Casey Dugan, Yla Tausczik, Horst Samulowitz, and Alexander Gray. 2019. Human-AI Collaboration in Data Science: Exploring Data Scientists' Perceptions of Automated AI. *Proceedings of the ACM on Human-Computer Interaction* 3 (2019), 1–24.

[312] Danding Wang, Qian Yang, Ashraf Abdul, and Brian Y Lim. 2019. Designing theory-driven user-centric explainable AI. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–15.

[313] Qianwen Wang, Yao Ming, Zhihua Jin, Qiaomu Shen, Dongyu Liu, Micah J. Smith, Kalyan Veeramachaneni, and Huamin Qu. 2019. ATMSeer: Increasing Transparency and Controllability in Automated Machine Learning. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.

[314] Nutthanan Wanluk, Sarinporn Visitsattapongse, Aniwat Juhong, and C Pintavirooj. 2016. Smart wheelchair based on eye tracking. In *Proceedings of the 9th Biomedical Engineering International Conference*. IEEE, 1–4.

[315] Daniel Karl I Weidele, Justin D Weisz, Erick Oduor, Michael Muller, Josh Andres, Alexander Gray, and Dakuo Wang. 2020. AutoAIViz: opening the blackbox of automated artificial intelligence with conditional parallel coordinates. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*. 308–312.

[316] H James Wilson and Paul R Daugherty. 2018. Collaborative intelligence: humans and AI are joining forces. *Harvard Business Review* 96, 4 (2018), 114–123.

[317] Anja Wolker and Thomas E Powell. 2018. Algorithms in the newsroom? News readers' perceived credibility and selection of automated journalism. *Journalism* (2018), 1–18.

[318] Catherine Wong, Neil Houlsby, Yifeng Lu, and Andrea Gesmundo. 2018. Transfer Learning with Neural AutoML. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. 8366–8375.

[319] Doris Xin, Eva Yiwei Wu, Doris Jung-Lin Lee, Niloufar Salehi, and Aditya Parameswaran. 2021. Whither AutoML? Understanding the Role of Automation in Machine Learning Workflows. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–16.

[320] Wang Xing-Yu, Jin Jing, Yu Zhang, and Wang Bei. 2013. Brain control: human-computer integration control based on brain-computer interface approach. *Acta Automatica Sinica* 39, 3 (2013), 208–221.

[321] Qian Yang, Aaron Steinfeld, Carolyn Rose, and John Zimmerman. 2020. Re-examining whether, why, and how human-AI interaction is uniquely difficult to design. In *Proceedings of the 2020 chi conference on human factors in computing systems*. 1–13.

[322] Qian Yang, Jina Suh, Nan-Chen Chen, and Gonzalo Ramos. 2018. Grounding interactive machine learning tool design in how non-experts actually build models. In *Proceedings of the 2018 Designing Interactive Systems Conference*. 573–584.

[323] Quanming Yao, Mengshuo Wang, Yuqiang Chen, Wenyuan Dai, Yu-Feng Li, Wei-Wei Tu, Qiang Yang, and Yang Yu. 2018. Taking human out of learning applications: A survey on automated machine learning. *arXiv preprint arXiv:1810.13306* (2018).

[324] Ji Soo Yi, Youn Ah-Kang, John Stasko, and Julie A Jacko. 2007. Toward a deeper understanding of the role of interaction in information visualization. *IEEE transactions on visualization and computer graphics* 13, 6 (2007), 1224–1231.

[325] Kaichao You, Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. 2019. Universal domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2720–2729.

[326] Tal Zarsky. 2016. The trouble with algorithmic decisions: An analytic road map to examine efficiency and fairness in automated and opaque decision making. *Science, Technology, & Human Values* 41, 1 (2016), 118–132.

[327] Tomas Zemcik. 2021. Failure of chatbot Tay was evil, ugliness and uselessness in its nature or do we judge it through cognitive shortcuts and biases? *AI & SOCIETY* 36, 1 (2021), 361–367.

[328] Ce Zhang and Azim Eskandarian. 2021. A Survey and Tutorial of EEG-Based Brain Monitoring for Driver State Analysis. *IEEE/CAA Journal of Automatica Sinica* 8, 7 (2021), 1222–1242.

[329] Hongming Zhang and Tianyang Yu. 2020. AlphaZero. In *Deep Reinforcement Learning*. Springer, 391–415.

[330] Renqiao Zhang, Jiajun Wu, Chengkai Zhang, William T. Freeman, and Joshua B. Tenenbaum. 2016. A Comparative Evaluation of Approximate Probabilistic Simulation and Deep Neural Networks as Accounts of Human Physical Scene Understanding. In *Proceedings of the 38th Annual Meeting of the Cognitive Science Society, Recogbizing and Representing Events*. 1781–1786.

[331] Xiawu Zheng, Yang Zhang, Sirui Hong, Huixia Li, Lang Tang, Youcheng Xiong, Jin Zhou, Yan Wang, Xiaoshuai Sun, Pengfei Zhu, et al. 2021. Evolving Fully Automated Machine Learning via Life-Long Knowledge Anchors. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[332] Jichen Zhu, Antonios Liapis, Sebastian Risi, Rafael Bidarra, and G Michael Youngblood. 2018. Explainable AI for designers: A human-centered perspective on mixed-initiative co-creation. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)*. 1–8.

[333] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2020. A comprehensive survey on transfer learning. *Proc. IEEE* 109, 1 (2020), 43–76.

[334] Indre Zliobaite, Albert Bifet, Mohamed Gaber, Bogdan Gabrys, Joao Gama, Leandro Minku, and Katarzyna Musial. 2012. Next challenges for adaptive learning systems. *ACM SIGKDD Explorations Newsletter* 14, 1 (2012), 48–55.

[335] Jean-Daniel Zucker. 2003. A grounded theory of abstraction in artificial intelligence. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences* 358, 1435 (2003), 1293–1309.