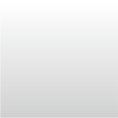
Knowledge and Information Systems (2023) 65:5097–5149 <https://doi.org/10.1007/s10115-023-01935-1>

[](http://crossmark.crossref.org/dialog/?doi=10.1007/s10115-023-01935-1&domain=pdf)

**REVIEW**

**Eight years of AutoML: categorisation, review and trends**

**Rafael Barbudo1,2 · Sebastián Ventura1,2 · José Raúl Romero1,2**

Received: 16 September 2022 / Revised: 25 April 2023 / Accepted: 10 July 2023 /

Published online: 8 August 2023 © The Author(s) 2023

**Abstract**

Knowledge extraction through machine learning techniques has been successfully applied in

a large number of applicationdomains. However, apartfromtherequiredtechnicalknowledge and background in the application domain, it usually involves a number of time-consuming and repetitive steps. Automated machine learning (AutoML) emerged in 2014 as an attempt to mitigate these issues, making machine learning methods more practicable to both data sci- entistsanddomain experts. AutoML is abroad area encompassing a widerange ofapproaches aimed at addressing a diversity of tasks over the different phases of the knowledge discovery process being automated with speciﬁc techniques. To provide a big picture of the whole area, we have conducted a systematic literature review based on a proposed taxonomy that permits categorising 447 primary studies selected from a search of 31,048 papers. This review per- forms an extensive and rigorous analysis of the AutoML ﬁeld, scrutinising how the primary studies have addressed the dimensions of the taxonomy, and identifying any gaps that remain unexplored as well as potential future trends. The analysis of these studies has yielded some intriguing ﬁndings. For instance, we have observed a signiﬁcant growth in the number of publications since 2018. Additionally, it is noteworthy that the algorithm selection problem has gradually been superseded by the challenge of workﬂow composition, which automates more than one phase of the knowledge discovery process simultaneously. Of all the tasks in AutoML, the growth of neural architecture search is particularly noticeable.

**Keywords** AutoML · Algorithm selection · Hyper-parameter optimisation · Neural architecture search · Workﬂow composition · Systematic literature review

B José Raúl Romero jrromero@uco.es

Rafael Barbudo

rbarbudo@uco.es

Sebastián Ventura sventura@uco.es

1 Department of Computer Science and Numerical Analysis, University of Córdoba, 14071 Córdoba, Spain

2 Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), Córdoba, Spain

**1 Introduction**

Despite the huge amount of data available in corporations, there are still a large number of

them, specially small- and medium-sized organisations, that do not handle and take advan- tage of such potential information. Deﬁnitely, its analysis could bring valuable insights into their working activity [[9](#bookmark2)]. Extracting useful and novel knowledge for decision-making from raw data is a complex process that involves different phases [[12](#bookmark3)] and requires strong back- ground in multiple ﬁelds, such as pattern mining, databases, statistics, data visualisation or machine learning. In general, data scientists are responsible for identifying relevant data sources, automating the collection processes, preprocessing and analysing the acquired data, and communicating their ﬁndings to the stakeholders, so that they can propose solutions and strategies to business challenges [[27](#bookmark4)]. The role of these data professionals is crucial in reaching quality decisions, requiring from their experience, intuition and the application of their know-how.

Alongtheknowledgediscoveryprocess, differentphases are accomplished, some ofwhich are human-dependent like interpreting the mined patterns. In contrast, others are made up of tasks that are repetitive and time-consuming. Also, it comes into play that the area is progressing fast and the algorithms and methods used for cleaning data or learning from them are becoming more and more sophisticated, and also more and more numerous [[21](#bookmark5)]. In fact, some methods may be better suited than others to the data characteristics and formulation of a particular organisation [[24](#bookmark6)]. This means that automated methods can make it more practicable for the data professional to explore the alternatives that best ﬁt the problem at hand, as well as alleviating those tasks that are more time-consuming. An illustrative example of this situation is model building, where the large number of available machine learning (ML) algorithms makes it difﬁcult to take the right decisions for the construction of the most appropriate models [[1](#bookmark7), [128](#bookmark8), [344](#bookmark9)]. Here, apart from selecting the right algorithms, their performance could heavily rely on the selected values for their hyper-parameters [[4](#bookmark10), [229](#bookmark11), [364](#bookmark12)]. In this context, the term automated machine learning (AutoML) [[20](#bookmark13)] was ﬁrst coined in 2014 to encompass those approaches that are committed to precisely automate these challenging tasks, allowing data scientists to cover a wider spectrum of alternatives, as well as shifting the focus to those phases requiring from their know-how and intuition and, ultimately, bringing the knowledge discovery process closer to domain experts [[77](#bookmark14), [129](#bookmark15)]. Since then, AutoML is a rapidly expanding ﬁeld. Indeed, some AutoML proposals have already shown that they can outperform humans for certain tasks, such as the design of neural network architectures [[8](#bookmark16), [483](#bookmark17)].

AutoML approaches can be analysed from multiple perspectives. Different concrete oper- ations or tasks like algorithm selection or hyper-parameter optimisation can be performed to automate the knowledge discovery process. Also, a wide range of techniques can be applied to complete a task. It is common in this ﬁeld to ﬁnd studies not explicitly categorised as AutoML, even though they are speciﬁcally oriented towards addressing AutoML problems like the design of neural network architectures, the selection of the best preprocessing or ML algorithms [[1](#bookmark7), [6](#bookmark18)] and their hyper-parameter values [[33](#bookmark19), [105](#bookmark20)], as well as the automatic composition of workﬂows for knowledge discovery [[129](#bookmark15), [285](#bookmark21), [307](#bookmark22)]. This might blur the scope, applicability and categorisation of these methods. Because of the growth of AutoML in recent years, other works have explored this ﬁeld across speciﬁc domains of knowledge. Nevertheless, we contend that a comprehensive global review, accompanied by a taxonomy of the ﬁeld, would provide a holistic overview and situational analysis of the ongoing efforts. This would help identify emerging trends, assess the extent of coverage of different phases of

the process, and highlight the dominant techniques being employed in each case. Therefore, this literature review focuses on the breadth of the knowledge area, prior to the requisite deepening of speciﬁc categories of tasks, phases or techniques.

More speciﬁcally, Luo [[26](#bookmark23)] conducted a review of the various approaches that deal with the selection of the most appropriate ML algorithms and the optimisation of their hyper-parameters. The study mainly focused on the supervised setting and the biomedical environment. With respect to the hyper-parameter optimisation, Hutter et al. [[19](#bookmark24)] examined approaches that use Bayesian optimisation and provided a concise overview of traditional techniques. Furthermore, Elsken et al. [[10](#bookmark25)] reviewed the design of neural network architec- tures.Intermsofautomatic workﬂow optimisation, Serbanetal. [[31](#bookmark26)]evaluatedtheIntelligent Discovery Assistants, which were described as systems that aid users in analysing their data. This comprehensive deﬁnition encompasses a vast number of diverse proposals, some of which are related to AutoML. Speciﬁcally, the authors assessed a limited number of approaches using AI planning to generate knowledge discovery workﬂows automatically. Recently, He et al. [[16](#bookmark27)] published a survey covering the automation of various phases of the knowledge discovery process, being specially oriented to the design of neural network architectures. Also, Hutter et al. [[20](#bookmark13)] presented a volume mainly focused on particular tasks and AutoML systems, such as Auto-sklearn, Auto-Net or Automatic Statistician.

To the best of our knowledge, this paper presents the ﬁrst systematic literature review that provides a comprehensive overview of the research ﬁeld of AutoML, without any speciﬁc emphasis on particular tasks or phases. Considering that this area embraces a broad range of proposals, we ﬁrst categorise and then analyse them according to the following perspectives: phases, tasks and techniques. Regarding the phases, we discuss to what extent AutoML has covered the knowledge discovery process, as this could help practitioners to identify the appropriate initiative to use at a given moment. As for the tasks, it is interesting to provide insights on how phases have been automated. This information will contribute to understand the level of expertise required by those implementing an AutoML method. We also study those techniques being considered in AutoML and identify possible relationships among them, as well as cross-relations with phases and tasks.

Following a rigorous methodology, we conduct a systematic literature review that starts with an exhaustive search returning 31,048 manuscripts, from which we detected and exam- ined a total of 447 primary studies between 2014 and 2021. All these studies are analysed and ﬁltered according to a precisely deﬁned protocol based on best practices [[22](#bookmark28)]. As a result, we ﬁrst categorise the main characteristics of current AutoML developments, including the terms referring to the conducted tasks, the covered phases and the most widely applied tech- niques. These terms are formulated and interrelated in form of a high-level taxonomy that provides a frame to understand how current AutoML studies are organised. The analysis of the primary studies also leads to interesting ﬁndings. Just to highlight a few, the quali- tative analysis reveals that those phases of the knowledge extraction process considered as inherently human—e.g. the interpretation of results—have been barely considered yet. Also, it has been observed that the increasing complexity of the problems addressed has led to the automation of more than one phase simultaneously. In this vein, algorithm selection is gradually replaced by the composition of knowledge discovery workﬂows. It is also inter- esting to note that interesting relationships of techniques with phases and tasks have been found. For example, evolutionary computation and reinforcement learning was widely used by early proposals for the design of neural network architectures, while more recent studies are mainly based on gradient-based methods.

The rest of the paper is organised as follows: Section[2](#bookmark29) provides some theoretical back- ground on the knowledge discovery process and the most recurrent tasks within AutoML.

The research questions and the review methodology are explained in Sect.[3.](#bookmark30) Then, in Sect.[4](#bookmark31) we introduce our AutoML taxonomy, which serves as a guideline for the analysis of the extracted primary studies. Some relevant quantitative ﬁndings are reported in Sect.[5](#bookmark32), while Sect.[6](#bookmark33) discusses other more qualitative ﬁndings from the analysis process. Then, Sect.[7](#bookmark34) performs a cross-analysis between phases, tasks and techniques. The conducted analysis has served to identify open issues and future lines of research that are discussed inSect.[8](#bookmark35). Finally, we conclude with Sect.[9.](#bookmark36)

**2 Background**

The term AutoML ﬁrst appeared in 2014 to refer to those approaches automating the machine learning process. In the ﬁrst instance,it cannot be attributed to a single author or reference, but to a complete community. More precisely, the ﬁrst occurrence was in the AutoML work- shop, which has been held jointly with the *International Conference on Machine Learning* (ICML) for the last years (2014–2021). Recently, this workshop has been transformed into the AutoML conference in 2022. To the best of our knowledge, the ﬁrst papers explicitly referring to AutoML were published in 2015 [[14](#bookmark37), [129](#bookmark15)]. Regardless of thesedates, notice that some problems addressed by AutoML have been studied for decades, such as the selection of the best ML algorithms and the tuning of theirrespective hyper-parameter values.

As mentioned above, AutoML aims at automating the different phases of the knowledge discovery process (Sect.[2.1](#bookmark38)) by performing some particular tasks (Sect.[2.2](#bookmark39)) in different ways. AutoML is not only intended to be useful by both domain experts and data scientists. Indeed, certain tasks like hyper-parameter optimisation or neural architecture search (NAS) are intended to improve the performance of the generated models, even though they still require some prior knowledge on the applicable algorithms and their hyper-parameters. In this context, Olson et al. [[307](#bookmark22)] stated that AutoML is not a replacement for data scientists nor ML practitioners, but a “data science assistant” . Similarly, Gil et al. [[13](#bookmark40)] proposed the development of interactive AutoML approaches that leverage the experience and background ofbothdomain experts anddatascientists, as they can complementand constrainthe available data and solutions to their particular needs and requirements.

**2.1 Phases of the knowledge discovery process**

Differentmethodologies,eachwithits ownphases,have beenproposedtoconducttheknowl- edge discovery process. A representative example is the Knowledge Discovery in Databases (KDD) process that was originally described by Fayyad et al. [[12](#bookmark3)]. It is an iterative and inter- active process whose cornerstone is the data mining phase, which aims at extracting patterns from large databases. This process starts by studying the application domain and any prior knowledge relevant to the process. Then, a target dataset is built by focusing on a subset of variables, or data samples, on which the discovery is to be performed. To ensure its qual- ity, a number of data cleaning and transformation methods is applied. Once the dataset has been properly preprocessed, the data mining phase is conducted and the resulting patterns are evaluated. In this context, different data mining methods are identiﬁed: classiﬁcation, regression, clustering, summarisation, dependency modelling or outliers detection, among others. Finally, the derived knowledge is documented, reported and incorporated to another system for further actions like decision-making.

Alternatively, the CRoss-Industry Standard Process for Data Mining (CRISP-DM) was deﬁned by Chapman et al. [[7](#bookmark41)]. This process, which is specially designed for industry, consists on a cycle encompassing six phases. The ﬁrst two phases, business understanding and data understanding, are closely related to the domain understanding in KDD. In a third phase, raw data is subject to a set of transformations to improve its quality, thus helping to deploy better models during the fourth phase, which is equivalent to the data mining phase deﬁned by KDD. The resulting models are thoroughly evaluated and the previous steps are revisited if needed, until the business objectives are met. Finally, the acquired knowledge is organised and presented to the customers. Closely related, the Sample, Explore, Modify, Model, Assess (SEMMA) [[2](#bookmark42)] is another process consisting in a cycle with ﬁve stages, as referred by the ﬁve terms of the process name. In short, these stages can be seen as equivalent to those deﬁned by KDD or CRISP-DM. Notice that all these methodologies are analogous with respect to

the phases they involve, the data mining phase being their cornerstone.

**2.2 Tasks in AutoML**

A preliminary analysis of the manuscripts compiled by the aforementioned surveys and reviews revealed the existence of a set of common operations or tasks conducted in the ﬁeld of AutoML, namely hyper-parameter optimisation, neural architecture search, algorithm selection and workﬂow composition. This section brieﬂy introduces these tasks.

**2.2.1 Hyper-parameter optimisation and neural architecture search**

Most ML algorithms have at least one parameter (a.k.a. hyper-parameter) controlling how they behave. Illustrative examples are the kernel of a support vector machine or the maximum depth of a decision tree. Not properly tuning such hyper-parameters can greatly hamper the performance of these algorithms. In this context, the hyper-parameter optimisation (HPO) task aims at automatically selecting the hyper-parameters values that maximise the perfor- mance of a given algorithm [[4](#bookmark10)]. As this problem has been studied for decades, a large number of techniques have been applied like grid search, random search, evolutionary algorithms, racing algorithms or Bayesian optimisation, among others. Hutter et al. [[19](#bookmark24)] reviewed the HPO problem focusing on Bayesian optimisation and giving a brief overview of the tradi- tional techniques. They also analysed some approaches focused on reducing the number of hyper-parameters and discussed how to combine them with traditional HPO approaches.

Recently, the availability of new scalable HPO approaches [[323](#bookmark43)] has supported the devel- opment of deep neural networks, which have achieved a great success in various application domains like speech [[17](#bookmark44)] and image recognition [[15](#bookmark45)]. This has led to the emergence of the ﬁeld known as neural architecture search, which is an instance of the HPO problem aiming at simultaneously optimising the architecture and hyper-parameters of the artiﬁcial neural networks [[10](#bookmark25)]. It is worth noting that these automatically designed neural networks are com- parable or even outperform hand-designed architectures [[8](#bookmark16),[483](#bookmark17)]. Elsken etal. [[10](#bookmark25)] published a survey that categorises the NAS approaches according to three dimensions: search space, search strategy and performance estimation strategy. More recently, a survey mostly covering NAS, as well as some preprocessing steps of the knowledge discovery process, has appeared in the literature [[16](#bookmark27)]. Regarding the NAS approaches, they are categorised according to their search space, optimisation technique and model evaluation method. As for the preprocessing, it considers both data preparation and feature engineering methods.

**2.2.2 Algorithmselection**

Itiswellknown thatthereis no single algorithmthatoutperformsthe othersfor allthe available problems [[36](#bookmark46)]. In the context of ML, algorithms are based on a set of assumptions about the data,i.e. their inductive bias, what makes some of them more suitable for a particular dataset. Rice [[29](#bookmark47)] formalised the algorithm selection problem (AS) with the aim of selecting the best algorithm for each situation and thus limiting the aforementioned shortcoming. It is usually treated as a ML problem where the dataset characteristics, e.g. the number of attributes and classes, are used to predict the expected algorithm performance, which could be measured by its accuracy. The aim is to predict whether an algorithm—or a set of algorithms—is suitable for a speciﬁc dataset. Recently, Luo [[26](#bookmark23)] reviewed a number of proposals dealing with the AS and/or the HPO during the data mining phase. More speciﬁcally, it is focused on supervised learning,i.e. classiﬁcation and regression. Similarly, Tripathy and Panda [[35](#bookmark48)] reviewed those proposals that exploit meta-learning techniques to carry out the selection of the best data mining algorithm. It is worth noting that the AS problem has also been considered in other application domains like the combinatorial optimisation search, which was reviewed by Kotthoff [[23](#bookmark49)].

A related task is the Combined Algorithm Selection and Hyper-parameter optimisation (CASH). It was deﬁned by Thornton et al. [[34](#bookmark50)], who applied a Bayesian optimisation algo- rithmtojointlyaddressthe AS and HPOproblems. This was achievedby treatingtheselection of the ML algorithm as an hyper-parameter itself, thus building a conditional space of hyper- parameters where the selection of a given algorithm triggers the optimisation of other certain hyper-parameters. In addition,assigningspeciﬁc valuesofcertainhyper-parameters may trig- ger the optimisation of other hyper-parameters. Previous to its formalisation, this problem was already addressed by Escalanteetal. [[11](#bookmark51)] with particle swarm optimisation.

**2.2.3 Workﬂow composition**

While the AS and the HPO provides a valuable assistance during the knowledge discov- ery process, their use is rather limited to data mining phase [[6](#bookmark18)]. However, selecting and tuning the best algorithm for each phase independently could hamper the performance and even generate invalid sequences of algorithms. There are relationships, synergies and con- straints between such algorithms that should be thoroughly examined to reach a pipeline with a good performance. Therefore, some authors have begun to propose methods that can automatically compose knowledge discovery workﬂows involving two or more phases of the process. These proposals rely on different techniques such as particle swarm optimisa- tion [[11](#bookmark51)], Bayesian optimisation [[34](#bookmark50), [129](#bookmark15)], evolutionary computation [[307](#bookmark22)] or automated planning and scheduling [[285](#bookmark21)].

As can be inferred from the applied techniques, workﬂow composition has been mainly addressed as an optimisation problem. This poses a number of new challenges related to the increase of the search space. Apart from optimising the structure of the workﬂow itself, the bestalgorithmsshouldbe selectedforeachphase as well. This is even more challengingwhen the hyper-parameters of these algorithms need to be considered. To address this problem, some proposals ﬁxthe workﬂow structure in advance to generate simpler workﬂows [[30](#bookmark52), [34](#bookmark50), [129](#bookmark15), [285](#bookmark21)] or give preference to the shorter ones during the optimisation process [[307](#bookmark22)]. Although not speciﬁcally oriented towards the workﬂow composition problem, Serban et al. [[31](#bookmark26)] reviewed a number of intelligent discovery assistants, which are systems that advise users during the data analysis process. Among them, they identiﬁed a set of approaches using automated planning and scheduling techniques to automatically compose ML workﬂows [[5](#bookmark53),

[37](#bookmark54)]. More recently, Hutter et al. [[20](#bookmark13)] analyse in detail some AutoML tools performing this task.

**3 Review methodology**

As an approachforthedevelopment ofthis study, we have followedtheguidelinesby Kitchen- ham and Charters [[22](#bookmark28)]. They provide a rigorous and structured approach to identifying and summarising all relevant research on a particular topic, ensuring that the review process is transparent and minimises potential sources of bias. These guidelines can assist us in con- ducting our study in a manner that is replicable by other researchers, which is essential for building a reliable body of knowledge in the ﬁeld of AutoML. Note that this approach was created according to the experience of domain experts in a variety of disciplines interested in evidence-based practice, and has undergone several revisions since it was ﬁrst published. In this way, they have gradually reﬁned a process that leads to high-quality SLRs by carrying out several steps that researchers should follow. In general, the review process comprises three main steps, namely *planning*, *conducting* and *reporting*. In terms of the *planning* step, it involves identifying the need for the review, which is motivated in Sect.[1](#bookmark1), and deﬁning the research questions (see Sect.[3.1](#bookmark55)) along with creating a review protocol. The *conducting* step involves identifying a wide range of primary studies pertaining to the research question and then selecting those that are most relevant (see Sect.[3.2)](#bookmark56). The quality of primary studies is evaluated, and to prevent any potential biases, the decision is solely founded on the source of the primary studies. Upon obtaining the primary studies, data extraction forms are developed to record the information accurately that researchers obtain from them (see Sect.[3.3](#bookmark57)), and later synthesised. Finally, the *reporting* step involves specifying mechanisms for dissemina- tion and formatting the main report. To ensure replicability, the review protocol, as well as the outcomes of both the search and review processes, is available as supplementary material.

**3.1 Research questions**

This paper aims at responding to the following research questions (RQs):

**RQ1:** From the primary studies, is it possible to extract a common terminology in the ﬁeld of AutoML?

In recent years, there has been a signiﬁcant increase in the number of papers on AutoML. However, many of these papers are not explicitly classiﬁed as AutoML, which could lead to a blurred understanding of their scope, applicability and categorisation. The lack of a common terminology can also lead to confusion and inconsistencies in the ﬁeld, hindering progress and collaboration. Therefore, we envisage that the establishment of a common taxonomy will assist researchers and practitioners in understanding and communicating each other’s work more precisely and consistently, ultimately advancing the ﬁeld of AutoML.

**RQ2:** In quantitative terms, how does the research in AutoML evolve?

The aim of this research question is to examine the quantitative evolution of the ﬁeld of AutoML. The rationale behind this lies in the fact that, while some AutoML problems, such as AS and HPO, have been studied for decades, the term AutoML as a comprehensive

**Table 1** The three search strings deﬁned

|  |  |  |
| --- | --- | --- |
| Search string | #1 | “machine learning” AND (“algorithm selection” OR “algorithm  recommendation” OR ((hyperparameter OR “hyper parameter”) AND (optimization OR tuning)) OR (auto\* AND (workﬂow OR pipeline)  AND (planning OR design OR generation OR composition))) |
| Search string | #2 | “neural architectur\* search” OR (auto\* AND (“neural network” OR “deep learning”) AND (design OR search OR evol\*) AND architectur\*) |
| Search string | #3 | Auto\* AND “machine learning” AND ((data AND (preprocessing OR  “preprocessing” OR integration OR clean\* OR transformation)) OR (feature AND (engineering OR selection OR extraction)) OR  postprocessing OR “post-processing” OR (knowledge AND (ﬁltering OR integration)) OR ((pattern OR model) AND (interpretation OR  explanation))) |

discipline was only introduced in 2014. By answering this question, researchers and practi- tioners will gain insight into the rate at which new studies are being published. Understanding the quantitative development of AutoML can also assist in identifying emerging trends and patterns in research, as well as potential areas for future work.

**RQ3:** Which are the phases of the knowledge discovery process covered by the different tasks performed in AutoML and what different techniques are applied?

The knowledge discovery process is a fundamental aspect of data mining and involves several key phases. By analysing the different tasks performed in AutoML, researchers and practitioners can gain a better understanding of how each phase of the knowledge discovery process is being addressed in the ﬁeld, and what techniques are being applied. This informa- tion can help categorise current and future proposals, identify strengths and weaknesses in the current approaches, and suggest potential areas for improvement or future research. In addition, a cross-category analysis can assist in identifying the relationships and synergies among phases, tasks and techniques.

**RQ4:** What are the emerging trends and open gaps for future research?

Given that AutoML is a broad and still emerging ﬁeld of research, the rationale behind this research question is to identify emerging trends and open gaps in the ﬁeld of AutoML for future research. This can help guide future research directions and ensure that research efforts are focused on areas that are most likely to lead to signiﬁcant advances in the ﬁeld. Also, this information can help guide researchers and practitioners towards important research areas and facilitate the development of more effective and efﬁcient AutoML systems. Overall, this research question can provide valuable insights into the current state of AutoML research and future directions for the ﬁeld.

**3.2 Literature search and selection strategy**

The ﬁrst step consists in the automatic literature search by querying the following digital librariesand citationdatabases:IEEEXplore, ScienceDirect, ACM Digital Library, Springer- Link,ISIWebofKnowledgeand Scopus. Table[1](#bookmark58)shows thethreeﬁnal searchstrings executed. From the output returned by a initial version of these search strings, we conducted an analysis

Merge and

Application

of inclusion

criteria

Automatic

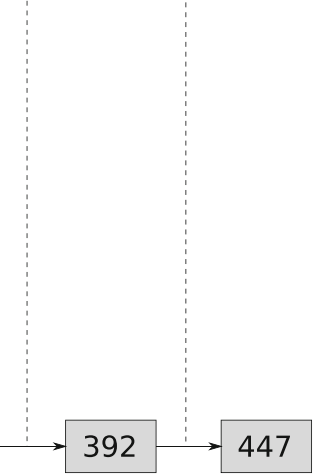
variants

identification

snow-balling procedure

application of

literature search

exclusion criteria





ssl

ss2

ss3



2327



ssl

ss2

ss3



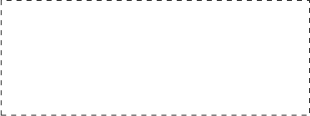
ssl

ss2

ss3



3,048



ssl

ss2

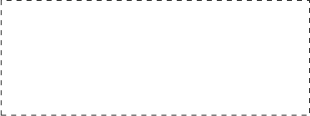
ss3



10,988

primary studies

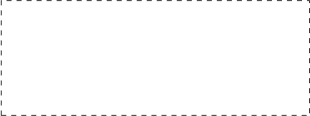
ss3 6,657



ss1 1,455

ss2 2,482

10,202



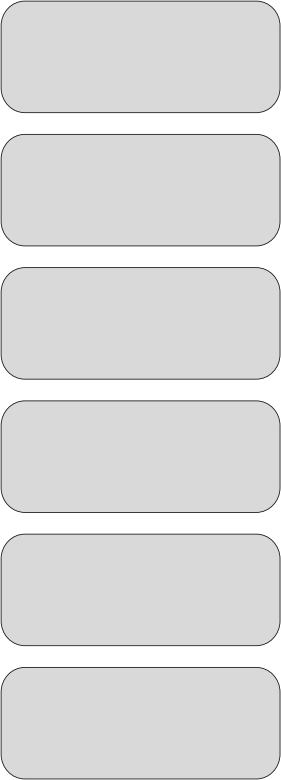
ssl

ss2

1,079

ss3 2,335

3,720



ACM Digit a Library

science Direct

springer Link

IEEE xplore

Sc opus

ISI Web of science

**Fig. 1** Quantitative results of the search and selection steps of primary studies

to deﬁne the taxonomy that depicts the main characteristics of current AutoML approaches. This initial version of the taxonomy serves to reﬁne these pilot search strings as well. For example, the strings are completed with sets of tasks and phases being used to determine a list of terms guiding the search. As a strategy, the search strings #1 and #2 seek to cover the main tasks of AutoML, while string #3 is more general, and includes those studies considering the automation of any phase composing the knowledge discovery process. In this way, we avoid limiting the search to the main AutoML tasks, thus offering a wider picture of the area. As shown in Fig.[1](#bookmark59), a total of 31,048 papers are returned by these queries before removing duplicates and applying the following exclusion criteria:

1. Papers prior to the appearance of the term AutoML,i.e. 2014, are not considered.

2. Papers that are not written in English are not considered.

3. Papers with an unclear evidence of a blinded, peer-review process are not considered.

4. Papers that are not published in conferences ranked A\* or A according to the CORE ranking system[1](#bookmark60) are not considered.

5. Papers published in journals not indexed in JCR are not considered.

Once the ﬁlter was applied, we obtained a total of 10,596 candidate papers. It is important to mention that the ﬁrst three criteria could be integrated into the search strings, depending on the speciﬁc query format of each database (see the review protocol in the supplementary material for further details). In order to make the results from the various digital libraries and citation databases consistent and to aid in the cleaning and homogenisation of the retrieved data, a set of Python scripts were developed. For this purpose, we invoked the bibtexparser and pybliometrics packages. Next, the titles and abstracts of candidate papers are double- checked to ensure that they meet the following inclusion criteria. Firstly, the automation must bedeﬁned for a phase of the knowledge discovery process and the contribution must be focused on making a real progress in the ﬁeld of AutoML, not as much on the speciﬁc

1 <https://www.core.edu.au/conference-portal> (last accessed: March 29, 2022).

1 灬

artiﬁcial intelligence methods applied. Additionally, the proposal must be independent of a speciﬁc dataset, and could be of a theoretical nature.

The application of these inclusion criteria reduces the number of selected papers to 404. Also, variants of a paper presenting similar results—or a subset of the results—without a clearly differentiated contribution are removed. Note that extended versions in journals prevail over conferences. The identiﬁcation of these variants concludes with 392 primary studies, which were complemented by a snow-balling procedure consisting of checking cross-references and detecting papers potentially omitted by the automatic literature search. More precisely, each referenced paper is treated as a potential primary study as if it had been obtained through the search strings. Then, inclusion and exclusion criteria are applied as already described. To avoid the exponential growth of primary studies, we limit the snow- balling procedure to one level. This is, it is not applied to primary studies obtained through it. As shown in Fig.[1](#bookmark59), 447 primary studies were ﬁnally selected for the data collection and review process.

**3.3 Data collection process**

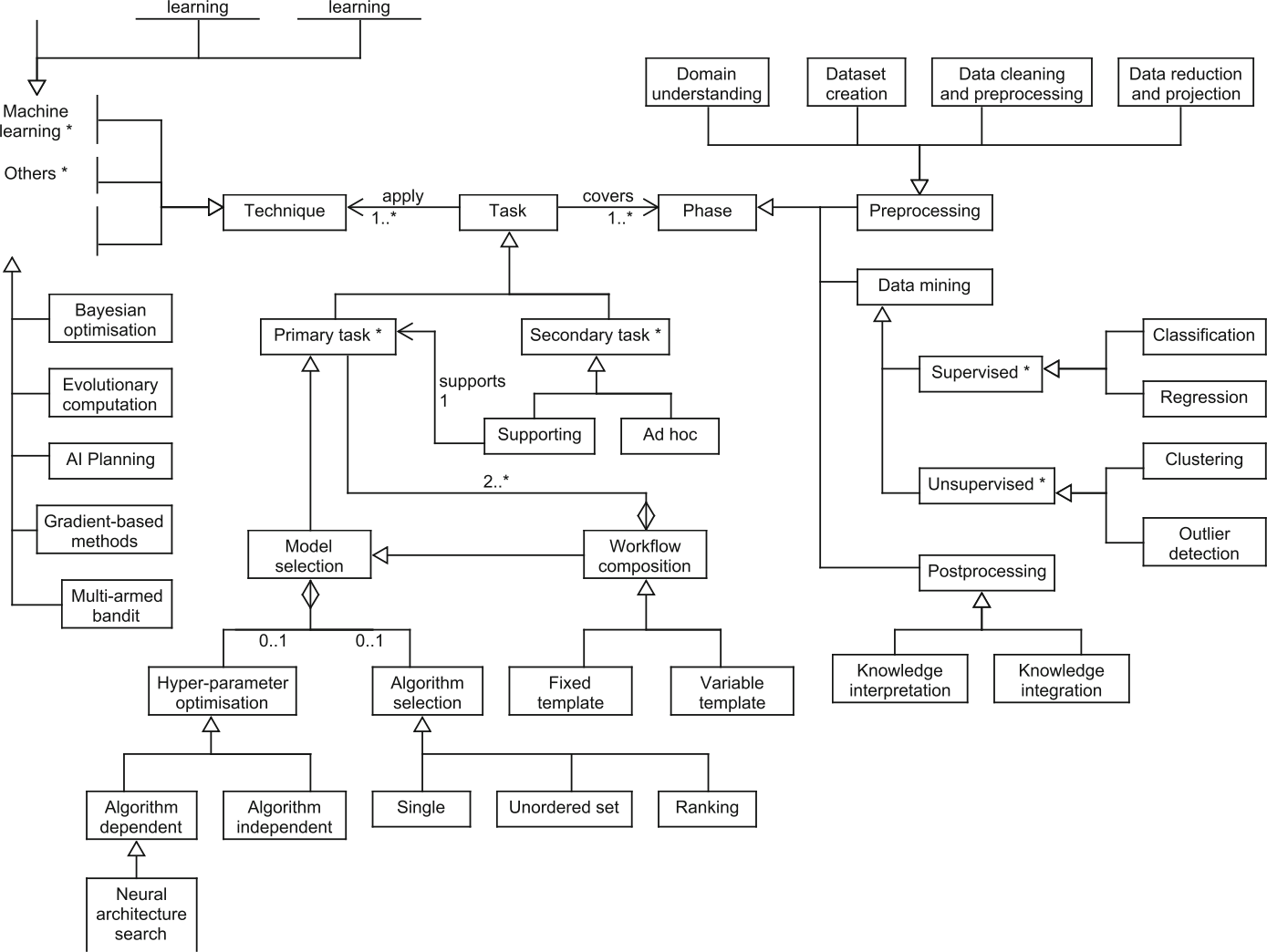
Next, we conduct the review process by thoroughly analysing the primary studies. With this aim, these papers are randomly distributed among the three authors according to the following rules: (1) every paper should be analysed by at least two authors; (2) there should be a balance among the reviewers with respect to the number of papers, their source (i.e. journal, conference) and the search string they are related to. Following these rules, all authors ﬁll a data extraction form for each paper. Then, ﬁlled forms concerning the same paper are compared to mitigate any possible error during the review process. In the event of discrepancies,the third author is contacted to reach a consensus form. It should be noted that all papers have been objectively reviewed under strict control conditions, and no other information is inferred during the extraction process. Therefore, all data being reﬂected in the extraction forms is explicitly collected from primary studies. Consequently, data may be missing for some categories in some of the primary studies analysed. Notice that the authors of the papers are not contacted in order to avoid any type of subjectivity or bias.

For each primary study, comprehensive information is collected for classiﬁcation regard- ing the phases of the knowledge discovery process being automated, the accomplished tasks and the applied techniques. In addition, information on the experimental framework and the additional material available—mostly related to replicability—is analysed too. These cate- gories are further divided into more speciﬁc properties aligned with the taxonomy developed in Sect.[4.](#bookmark31) A more detailed description on the data extraction form, including these properties and their possible values, can be found in the additional material. Together with this infor- mation, we also retrieve a brief summary of the primary study, as well as a brief overview of the future work reported by the authors. In terms of meta-information, each primary study

is annotated with the following identiﬁcation data: (a) author(s) name; (b) title; (c) year of publication; (d) type of publication,i.e. conference or journal; (e) name of publication and publisher; (f) volume, issue and pages; and (g) its DOI.

**4 A taxonomy for AutoML**

In response to RQ1,Fig.[2](#bookmark61)shows our taxonomy, which reﬂectsthe main classiﬁcation scheme obtained from the analysis of current AutoML approaches. Our taxonomy is constructed on

Eight years of AutoML: categorisation, review and trends 5107

|  |
| --- |
| Meta-learning |

|  |
| --- |
| Reinforcement |

|  |
| --- |
| Transfer |





|  |
| --- |
| search and optimisation \* |



**Fig. 2** Taxonomy for classifying terms in the ﬁeld of AutoML

the basis of the three core AutoML terms: (1) the *phases* of the knowledge discovery process being automated; (2) the *tasks* conducted to perform such automation; and (3) the *techniques* carrying out these tasks. As explained in Sect.[3](#bookmark30), a preliminary version of this taxonomy was ﬁrst drafted from the pilot literature search and then iterativelyreﬁned along the review process as primarystudieshave been examined. Theresultingtaxonomyrepresentsthecurrent state of the art of the ﬁeld. In Fig.[2](#bookmark61), those terms followed by an asterisk symbol (\*) denote extension points,i.e. those points in the classiﬁcation scheme where it can be expected to extend the terminology as the number of studies in the area increases. It is worth noting that this taxonomy is not intended to provide an exhaustive list of terms of the area, but a categorisation scheme for AutoML proposals.

As pointed out in Sect.[2.1](#bookmark38), there are different methodologies to conduct the knowledge discovery process. They all deﬁne a number of *phases* that are likely to be automated. Here, KDD, CRISP-DM and SEMMA have been considered in this taxonomy as the most general models from which to extract the different phases. Thus, we distinguish between *Prepro- cessing*, *Data mining* and *Postprocessing*. These three top categories are further divided into more speciﬁc terms. The preprocessing encompasses those phases that are performed prior to the generation of models, such as *Domain understanding*, *Dataset creation*, *Data cleaning and preprocessing* and *Data reduction and projection*. Regarding the data mining phase, both *Supervised*, i.e. *Classiﬁcation* and *Regression*, and *Unsupervised* learning,i.e. *Clustering* and *Outlier detection*, are considered. Note that only those categories found in the literature are explicitly considered in the taxonomy, without limiting the possibility that future studies may include other types of proposals, such as pattern mining under *Unsu- pervised* approaches. Finally, postprocessing includes the *Knowledge interpretation* and the

1 灬

*Knowledge integration* phases, which act on the knowledge discovered during the previous phases.

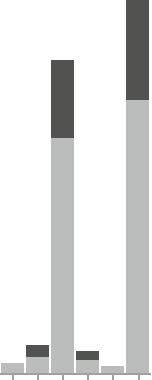
There is a set of *Task*s usually conducted in the ﬁeld of AutoML, which are referred as *Primary task*s in our taxonomy. In contrast, *Secondary task*s comprise those problems that have been studied to a lesser extent in the ﬁeld. Recurring primary tasks are *Algorithm selection* and *Hyper-parameter optimisation*, which aim at selecting the most accurate ML models,i.e. *Modelselection*. RegardingAS, we differentiatebetweenstudiesrecommending a *Single* algorithm or severalalgorithms, either an *Unorderedset* ofrecommendable algorithms or a *Ranking* that explicitly indicates the best ones. As for HPO, this category can be further decomposed into *Algorithm-dependent* and *Algorithm-independent* proposals, depending on whether they are designed to tune the hyper-parameters of a speciﬁc algorithm—or family of algorithms—or not. A representative example of the ﬁrst group is *Neural architecture search*, which optimises both the architecture and the hyper-parameters of artiﬁcial neural networks. Furthermore, we deﬁne *Workﬂow composition* as a speciﬁc model selection problem, where different algorithms, and optionally their hyper-parameters values, are selected. This task can be further decomposed according to whether the generated workﬂows follow a *Fixed template* or a *Variable template*. It is worth noting that these tasks require a set of algorithms, so they are only applicable if a catalogue of alternative algorithms that address them is already available, for example, contrary to what happens in cases such as the knowledge interpretation. Focusing on the secondary tasks, we distinguish between *Supporting* and *Ad hoc* tasks. On the one hand, supporting tasks refer to those that are closely related to primary tasks and are usually aimed at assisting them. As an example, recommending whether it is worth investing resources in tuning the hyper-parameters of a ML algorithm may accelerate the HPO process. On the other hand, ad hoc tasks are those speciﬁcally designed for the phase being automated, e.g. generating a report explaining in natural language a ML model.

Finally, primary studies have shown a wide range of *Technique*s applied to conduct AutoML tasks. Notice that applying two or more techniques simultaneously to address a single task is a common practice. Techniques can be mostly categorised into three groups: *Search and Optimisation*, *Machine Learning* and *Others*. Regarding *Search and Optimi- sation*, there are various techniques available such as *Bayesian optimisation*, *Evolutionary computation*, *AIPlanning*, *Gradient-basedmethods*, and*Multi-armedbandit* toconducttasks such as hyper-parameter optimisation. Additionally, in *Machine Learning*, it is noteworthy how ML techniques are used to conduct speciﬁc AutoML tasks. For instance, *Meta-learning* is commonly used to develop models for AS. *Transfer learning* involves reusing knowledge gained from solving a problem to tackle a related one and is often used in conjunction with other techniques. *Reinforcement learning*, which focuses on how agents should act in an environment to maximise rewards, is typically employed to design neural network architec- tures. Lastly, the *Others* category encompasses techniques that are scarcely used in primary studies and do not fall under any of the previous categories. It should be noted that, due to space limitations, the taxonomy covers speciﬁc techniques within these categories based on how frequently they appear in current AutoML studies. We expect to expand these categories as new proposals emerge.

**5 Quantitative analysis**

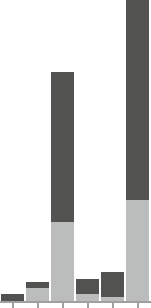
In response to RQ2, this section performs a quantitative analysis of the primary studies. Firstly, the number of contributions per year and their distribution by type of publication



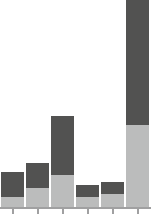
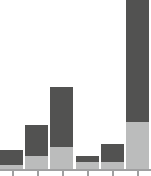
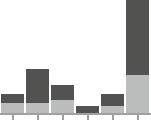
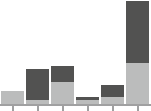
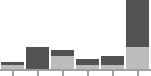
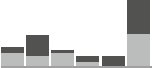


source





|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 兰 5 |  | 兰 5 |  | 兰 5 |  | 兰 5 |  | 兰 5 |  | 兰 5 |  | 兰 5 | 兰 |  |

**Fig. 3** Number of publications per year and task

(conference or journal) has been calculated. The idea is to identify current publication trends and check how AutoML evolves in different ways. Figure[3](#bookmark62) shows the number of primary studies published in conferences and journals per year since the appearance of the term AutoML. For each year, the distribution of papers focused on the different AutoML tasks (see our taxonomy in Sect.[4](#bookmark31)) is depicted,where the last column shows the total amount of papers for this year. Notice that the category HPO has been decomposed into algorithm-independent tasks, *HPO(indep)*, and algorithm-dependent tasks, *HPO(dep)*. All HPO tasks together with algorithm selection (*AS*) and workﬂow composition (*WC*) sum those papers related to the model selection primary task. Papers on secondary tasks (*ST*) are also calculated. As can be observed, the number of primary studies published between 2014 and 2017 remains fairly steady. However, since 2018 the number of papers has increased signiﬁcantly, mostly as the number of works related to *HPO(dep)* also grows. This is due to a large increase in the number ofNASproposals. Also,algorithmselectionhas traditionallyattracted more attention than workﬂow composition. This is the case until 2020, when we can speculate that there could be a change in trend due to an improvement in computational resources, what offer the possibility of executing workﬂow composition proposals that include part of the AS and HPO tasks. In the particular case of algorithm-independent proposals, *HPO(indep)*, its number has been larger than for *HPO(dep)* until 2017. As for the studies focused on secondary tasks, its number is still limited even when it involves both supporting and ad hoc tasks. Finally, notice that the number of journal papers is notoriously greater than publications in conferences in 2021. However, we speculate that it could be due to the COVID-19 situation.

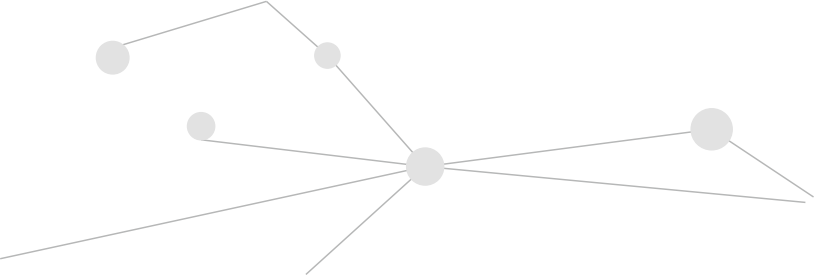
Regarding the type of publication, 55% of primary studies have been submitted to con- ferences, while 45% have been published in JCR journals. Also, observed that, following our search protocol, only top conferences (A\* and A) are considered in the analysis. We can ﬁnd 57 different conferences, including the *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (CVPR) (38 primary studies), the *International Conference on Machine Learning* (ICML) (20), the *International Joint Conference on Neural Networks* (IJCNN) (18), the *European Conference on Computer Vision* (ECCV) (16) and the *Genetic and Evo- lutionary Computation Conference* (GECCO) (15). As for the journals, the largest number of primary studies is published in *Neurocomputing* (19 papers), *IEEE Access* (17), *IEEE Transactions on Pattern Analysis and Machine Intelligence* (11) and *IEEE Transactions on Neural Networks and Learning Systems* (10). In general, no particular preference in terms of tasks or phases of the knowledge discovery process has been observed for one publication or the other. However, we observe that algorithm-dependent HPO approaches, which mainly include NAS-speciﬁc proposals, are commonly submitted to publications more focused on

5110 R. Barbudo et al.



The university f British columbia

IBM Research

university of cambridge

universitt Freiburg

Deakin university

udwi-Maximilians  universitt Minchen

Google LLC

National university of singapore

Massachusetts Institute of Technology

universidade de SO Paulo

Mellon university

carnegie

shanghajlagunversaty

Hong kong university of science and Technology

Tencent

university of cafornia , Berkealey

Harbin Institute of Technology

Beihang university

Tsinghua university

peng cheng Laboratory

Institute of Automation chinese Academy of sciences

university of chinese Academy of sciences

of wellington

victoria university

Huawei Noah ' S Ark Lab

peking university

Huawei Technologies

sydney

unbverstyofinology

**Fig. 4** Afﬁliations and collaborations

artiﬁcial neural networks or their applications like CVPR (37 primary studies), *Neurocom- puting* (15), ECCV (14) or IJCNN (11). Other conferences like GECCO have also accepted NAS-related papers (10), since they commonly apply evolutionary algorithms. Similarly, the largest number of WC studies is also found in GECCO (4) as well. As for algorithm- independent HPO approaches, ICML (13) and the *International Conference on Artiﬁcial Intelligence and Statistics* (AISTATS) (9) have attracted the majority of primary studies.

In average, each primary study in this area is co-authored by 4 researchers from 2 dis- tinctinstitutions. Altogether, over 1000 authors have been identiﬁed, most of them having participated in just one primary study. As for international collaborations, the emergence of international networking groups such as the COSEAL (COnﬁguration and SElection of [ALgorithms) group2](#bookmark64) is noteworthy. Figure[4](#bookmark63)shows the relationships among collaborating institutions. In this graph, nodes represent afﬁliations, while their size depicts the number of contributions. The edges indicate collaborations. However, due to space limitations and for the sake of readability, we do not show afﬁliations with less than 6 contributions. As can be seen, the institutions that have co-authored most primary studies are: *Google LLC* (16 papers in collaboration with others), *Carnegie Mellon University* (15), *Massachusetts Institute of Technology* (13),*Deakin University* (13),*Tsinghua University* (12) and *University of Chinese Academy of Sciences* (11). Of these, the *Beihang University* (20), *Tsinghua University* (18), *University of Chinese Academy of Sciences* (18) and *Tencent* (18) have been more proactive in collaborating with other institutions. Other afﬁliations with a large number of collabora- tions are *Massachusetts Institute of Technology* (16), *Peng Cheng Laboratory* (15), and *Hong Kong University of Science and Technology* (15).

**6 Findings of the review process**

In response to RQ3, the analysis presented next, which is founded on the proposed taxonomy in Sect.[4](#bookmark31), covers the phases automated (Sect.[6.1](#bookmark65)), the tasks conducting such automation (Sect.[6.2](#bookmark66)), and the techniques applied (Sect.[6.3)](#bookmark67). We have also examined the experiments carried out in the primary studies, as well as their available additional materials (Sect.[6.4)](#bookmark68). Notice that, due to space limitations and for the sake of readability, Tables [2](#bookmark69) , [3](#bookmark70)and [4](#bookmark71)use

2 COSEAL group: <https://www.coseal.net/> (last accessed: 24/08/2022).

1 灬

Eight years of AutoML: categorisation, review and trends 5111

**Table 2** Phases currently covered by AutoML

1 灬

|  |  |  |
| --- | --- | --- |
| Category | % | Papers |
| Preprocessing (14%) Domain understanding Dataset creation  Data cleaning and preprocessing | 11% 3% 48% | [[ 107](#bookmark72), [111](#bookmark73), [118](#bookmark74), [168](#bookmark75), [263](#bookmark76), [287](#bookmark77), [402](#bookmark78)] [[ 106](#bookmark79), [268](#bookmark80)]  [[59](#bookmark81), [61](#bookmark82), [62](#bookmark83), [69](#bookmark84), [97](#bookmark85), [121](#bookmark86), [122](#bookmark87), [129](#bookmark15), [140](#bookmark88), [158](#bookmark89), [204](#bookmark90), [207](#bookmark91), [273](#bookmark92), [278](#bookmark93), [285](#bookmark21), [294](#bookmark94), [297](#bookmark95), [307](#bookmark22), [311](#bookmark96), [331](#bookmark97), [333](#bookmark98), [343](#bookmark99), [356](#bookmark100), [361](#bookmark101), [375](#bookmark102), [408](#bookmark103), [428](#bookmark104), [445](#bookmark105), [450](#bookmark106), [473](#bookmark107)] |
| Data reduction and projection | 79% | [[50](#bookmark108), [57](#bookmark109), [59](#bookmark81), [61](#bookmark82), [69](#bookmark84), [74](#bookmark110), [91](#bookmark111), [93](#bookmark112), [97](#bookmark85), [119](#bookmark113), [121](#bookmark86), [122](#bookmark87), [129](#bookmark15), [135](#bookmark114), [148](#bookmark115), [158](#bookmark89), [200](#bookmark116), [203](#bookmark117), [204](#bookmark90), [206](#bookmark118), [207](#bookmark91), [212](#bookmark119), [264](#bookmark120), [273](#bookmark92), [285](#bookmark21), [293](#bookmark121), [295](#bookmark122), [297](#bookmark95), [307](#bookmark22), [311](#bookmark96), [316](#bookmark123), [317](#bookmark124), [331](#bookmark97), [333](#bookmark98), [341](#bookmark125), [343](#bookmark99), [356](#bookmark100), [361](#bookmark101), [375](#bookmark102), [377](#bookmark126), [390](#bookmark127), [393](#bookmark128), [408](#bookmark103), [428](#bookmark104), [445](#bookmark105), [450](#bookmark106), [461](#bookmark129), [473](#bookmark107), [482](#bookmark130), [484](#bookmark131)] |
| Data mining (93%) Classiﬁcation | 20%  71% | [[38](#bookmark132), [47](#bookmark133), [51](#bookmark134), [57](#bookmark109), [58](#bookmark135), [61](#bookmark82), [63](#bookmark136), [67](#bookmark137), [69](#bookmark84), [77](#bookmark14), [87](#bookmark138), [95](#bookmark139), [99](#bookmark140), [109](#bookmark141), [121](#bookmark86), [122](#bookmark87), [126](#bookmark142), [129](#bookmark15), [131](#bookmark143), [135](#bookmark114), [158](#bookmark89), [164](#bookmark144), [167](#bookmark145), [185](#bookmark146), [191](#bookmark147), [193](#bookmark148), [204](#bookmark90), [207](#bookmark91), [211](#bookmark149), [212](#bookmark119), [215](#bookmark150), [222](#bookmark151), [235](#bookmark152), [265](#bookmark153), [267](#bookmark154), [269](#bookmark155), [270](#bookmark156), [272](#bookmark157), [273](#bookmark92), [283](#bookmark158), [285](#bookmark21), [294](#bookmark94), [295](#bookmark122), [297](#bookmark95), [307](#bookmark22), [311](#bookmark96), [319](#bookmark159), [321](#bookmark160), [328](#bookmark161), [331](#bookmark97)[–333](#bookmark98), [337](#bookmark162), [338](#bookmark163), [340](#bookmark164), [342](#bookmark165), [343](#bookmark99), [345](#bookmark166), [346](#bookmark167), [352](#bookmark168), [356](#bookmark100), [361](#bookmark101), [371](#bookmark169), [375](#bookmark102), [377](#bookmark126), [379](#bookmark170), [392](#bookmark171), [395](#bookmark172), [397](#bookmark173), [401](#bookmark174), [404](#bookmark175), [408](#bookmark103), [411](#bookmark176), [427](#bookmark177)[–429](#bookmark178), [433](#bookmark179), [445](#bookmark105), [471](#bookmark180), [473](#bookmark107), [483](#bookmark17), [484](#bookmark131)]  † [[39](#bookmark181)[–46](#bookmark182), [48](#bookmark183), [49](#bookmark184), [52](#bookmark185)[–56](#bookmark186), [60](#bookmark187), [64](#bookmark188)[–66](#bookmark189), [73](#bookmark190), [75](#bookmark191), [78](#bookmark192)[–86](#bookmark193), [88](#bookmark194)[–94](#bookmark195), [96](#bookmark196), [98](#bookmark197), [100](#bookmark198)[–103](#bookmark199), [105](#bookmark20), [110](#bookmark200), [112](#bookmark201)[–117](#bookmark202), [120](#bookmark203), [123](#bookmark204)[–125](#bookmark205), [127](#bookmark206), [130](#bookmark207), [132](#bookmark208)[–134](#bookmark209), [136](#bookmark210)[–139](#bookmark211), [141](#bookmark212)[–145](#bookmark213), [147](#bookmark214), [149](#bookmark215)[–157](#bookmark216), [160](#bookmark217), [163](#bookmark218), [165](#bookmark219), [166](#bookmark220), [169](#bookmark221)[–172](#bookmark222), [174](#bookmark223)[–183](#bookmark224), [186](#bookmark225), [187](#bookmark226), [189](#bookmark227), [190](#bookmark228), [192](#bookmark229), [194](#bookmark230), [195](#bookmark231), [197](#bookmark232)[–200](#bookmark116), [202](#bookmark233),  [208](#bookmark234)[–210](#bookmark235), [214](#bookmark236), [216](#bookmark237)[–221](#bookmark238), [223](#bookmark239)[–231](#bookmark240), [233](#bookmark241), [234](#bookmark242), [236](#bookmark243), [238](#bookmark244)[–247](#bookmark245), [249](#bookmark246)[–252](#bookmark247), [254](#bookmark248)[–262](#bookmark249), [266](#bookmark250), [271](#bookmark251), [274](#bookmark252)[–277](#bookmark253), [279](#bookmark254)[–282](#bookmark255), [286](#bookmark256), [288](#bookmark257), [290](#bookmark258)[–292](#bookmark259), [296](#bookmark260), [300](#bookmark261)[–306](#bookmark262), [308](#bookmark263), [309](#bookmark264), [313](#bookmark265)[–315](#bookmark266), [318](#bookmark267), [320](#bookmark268), [322](#bookmark269), [323](#bookmark43), [326](#bookmark270), [329](#bookmark271), [334](#bookmark272)[–336](#bookmark273), [339](#bookmark274), [347](#bookmark275)[–351](#bookmark276), [353](#bookmark277), [355](#bookmark278), [357](#bookmark279)[–360](#bookmark280), [362](#bookmark281)[–369](#bookmark282), [372](#bookmark283)[–374](#bookmark284), [376](#bookmark285), [378](#bookmark286), [380](#bookmark287)[–385](#bookmark288), [387](#bookmark289), [388](#bookmark290), [391](#bookmark291), [394](#bookmark292), [396](#bookmark293), [399](#bookmark294), [400](#bookmark295), [403](#bookmark296), [405](#bookmark297)[–407](#bookmark298), [409](#bookmark299), [410](#bookmark300), [413](#bookmark301)[–421](#bookmark302), [423](#bookmark303)[–426](#bookmark304), [430](#bookmark305)[–432](#bookmark306), [434](#bookmark307)[–444](#bookmark308), [446](#bookmark309)[–460](#bookmark310), [462](#bookmark311)[–470](#bookmark312), [472](#bookmark313), [474](#bookmark314), [476](#bookmark315)[–482](#bookmark130)] |
| Regression | 6% 8% | [[61](#bookmark82), [69](#bookmark84), [71](#bookmark316), [76](#bookmark317), [95](#bookmark139), [173](#bookmark318), [205](#bookmark319), [207](#bookmark91), [211](#bookmark149)[–213](#bookmark320), [215](#bookmark150), [232](#bookmark321), [248](#bookmark322), [273](#bookmark92), [294](#bookmark94), [297](#bookmark95), [311](#bookmark96), [344](#bookmark9), [361](#bookmark101), [370](#bookmark323), [412](#bookmark324), [473](#bookmark107), [475](#bookmark325)]  † [[70](#bookmark326), [92](#bookmark327), [104](#bookmark328), [108](#bookmark329), [146](#bookmark330), [159](#bookmark331), [161](#bookmark332), [162](#bookmark333), [165](#bookmark219), [184](#bookmark334), [188](#bookmark335), [194](#bookmark230), [199](#bookmark336), [201](#bookmark337), [214](#bookmark236), [243](#bookmark338), [247](#bookmark245), [253](#bookmark339), [284](#bookmark340), [289](#bookmark341), [299](#bookmark342), [301](#bookmark343), [304](#bookmark344), [310](#bookmark345), [312](#bookmark346), [330](#bookmark347), [354](#bookmark348), [367](#bookmark349), [386](#bookmark350), [418](#bookmark351), [438](#bookmark352), [450](#bookmark106)] |
| Clustering | 2%  < 1% | [[71](#bookmark316), [128](#bookmark8), [324](#bookmark353), [325](#bookmark354), [327](#bookmark355), [398](#bookmark356), [484](#bookmark131)] † [[453](#bookmark357), [457](#bookmark358)] |
| Outlier detection  Postprocessing (1%)  Knowledge interpretation Knowledge integration | < 1%  67% 33% | [[196](#bookmark359), [237](#bookmark360)]  [[248](#bookmark322), [389](#bookmark361)] [[72](#bookmark362)] |

1 灬

5 112 R Barb udo et a l.

**Table 3** Tasks currently accomplished by AutoML

|  |  |
| --- | --- |
| Category % Papers | |
| *P. tasks*  *S. tasks* | *Algorithm selection (9%)*  Single algorithm 57% [[57](#bookmark109), [58](#bookmark135), [99](#bookmark140), [140](#bookmark88), [148](#bookmark115), [162](#bookmark333), [164](#bookmark144), [176](#bookmark363), [179](#bookmark364), [196](#bookmark359), [213](#bookmark320), [248](#bookmark322), [263](#bookmark76), [267](#bookmark154), [287](#bookmark77), [316](#bookmark123), [317](#bookmark124), [342](#bookmark165), [345](#bookmark166), [352](#bookmark168), [395](#bookmark172), [398](#bookmark356), [406](#bookmark365)] Set of algorithms 20% [[ 118](#bookmark74), [140](#bookmark88), [168](#bookmark75), [205](#bookmark319), [338](#bookmark163), [344](#bookmark9), [411](#bookmark176)]  Ranking of algorithms 30% [[38](#bookmark132), [47](#bookmark133), [59](#bookmark81), [95](#bookmark139), [107](#bookmark72), [128](#bookmark8), [263](#bookmark76), [287](#bookmark77), [324](#bookmark353), [325](#bookmark354), [340](#bookmark164), [397](#bookmark173)]  *Hyper-parameter optimisation (77%)*  Alg. independent 25% [[42](#bookmark366), [44](#bookmark367), [56](#bookmark186), [60](#bookmark187), [73](#bookmark190), [75](#bookmark191), [83](#bookmark368), [103](#bookmark199), [108](#bookmark329), [123](#bookmark204), [133](#bookmark369), [134](#bookmark209), [141](#bookmark212), [143](#bookmark370), [144](#bookmark371), [146](#bookmark330), [151](#bookmark372), [159](#bookmark331)[–162](#bookmark333), [165](#bookmark219), [176](#bookmark363), [178](#bookmark373), [179](#bookmark364), [182](#bookmark374), [184](#bookmark334), [185](#bookmark146), [194](#bookmark230), [195](#bookmark231), [197](#bookmark232), [198](#bookmark375), [202](#bookmark233), [208](#bookmark234)[–210](#bookmark235), [223](#bookmark239), [225](#bookmark376), [229](#bookmark11), [231](#bookmark240), [247](#bookmark245), [248](#bookmark322), [255](#bookmark377), [260](#bookmark378), [267](#bookmark154), [274](#bookmark252), [276](#bookmark379), [296](#bookmark260), [299](#bookmark342), [300](#bookmark261), [304](#bookmark344), [306](#bookmark262), [320](#bookmark268), [322](#bookmark269), [334](#bookmark272), [338](#bookmark163), [339](#bookmark274), [342](#bookmark165), [345](#bookmark166), [350](#bookmark380), [351](#bookmark276), [353](#bookmark277)[–355](#bookmark278), [363](#bookmark381), [364](#bookmark12), [367](#bookmark349), [373](#bookmark382), [385](#bookmark288), [406](#bookmark365), [407](#bookmark298), [414](#bookmark383), [418](#bookmark351)[–420](#bookmark384), [431](#bookmark385), [432](#bookmark306), [434](#bookmark307), [436](#bookmark386), [437](#bookmark387), [450](#bookmark106), [455](#bookmark388), [457](#bookmark358), [464](#bookmark389), [470](#bookmark312)]  Alg. dependent 5% [[40](#bookmark390), [63](#bookmark136), [76](#bookmark317), [167](#bookmark145), [189](#bookmark227), [251](#bookmark391), [266](#bookmark250), [275](#bookmark392), [283](#bookmark158), [289](#bookmark341), [310](#bookmark345), [327](#bookmark355), [386](#bookmark350), [393](#bookmark128), [398](#bookmark356), [401](#bookmark174), [475](#bookmark325)]  Neural architecture search 70% [[39](#bookmark181), [41](#bookmark393), [43](#bookmark394), [45](#bookmark395), [46](#bookmark182), [48](#bookmark183), [49](#bookmark184), [51](#bookmark134)[–55](#bookmark396), [64](#bookmark188)[–66](#bookmark189), [70](#bookmark326), [74](#bookmark110), [78](#bookmark192)[–82](#bookmark397), [84](#bookmark398)[–94](#bookmark195), [96](#bookmark196), [98](#bookmark197), [100](#bookmark198)[–102](#bookmark399), [104](#bookmark328), [105](#bookmark20), [110](#bookmark200), [112](#bookmark201)[–117](#bookmark202), [120](#bookmark203), [124](#bookmark400)[–127](#bookmark206), [130](#bookmark207)[–132](#bookmark208), [136](#bookmark210)[–139](#bookmark211), [142](#bookmark401), [145](#bookmark213), [147](#bookmark214), [149](#bookmark215), [150](#bookmark402), [152](#bookmark403)[–157](#bookmark216), [163](#bookmark218), [166](#bookmark220), [169](#bookmark221), [170](#bookmark404), [172](#bookmark222), [173](#bookmark318), [175](#bookmark405), [177](#bookmark406), [181](#bookmark407), [183](#bookmark224), [186](#bookmark225)[–188](#bookmark335), [190](#bookmark228)[–193](#bookmark148), [199](#bookmark336)[–201](#bookmark337), [214](#bookmark236), [216](#bookmark237)[–221](#bookmark238), [224](#bookmark408), [226](#bookmark409)[–228](#bookmark410), [230](#bookmark411), [232](#bookmark321)[–246](#bookmark412), [249](#bookmark246), [250](#bookmark413), [252](#bookmark247)[–254](#bookmark248), [256](#bookmark414)[–259](#bookmark415), [261](#bookmark416), [262](#bookmark249), [265](#bookmark153), [271](#bookmark251), [272](#bookmark157), [277](#bookmark253), [279](#bookmark254)[–282](#bookmark255), [284](#bookmark340), [286](#bookmark256), [288](#bookmark257),  [290](#bookmark258)[–292](#bookmark259), [302](#bookmark417), [303](#bookmark418), [305](#bookmark419), [308](#bookmark263), [309](#bookmark264), [312](#bookmark346)[–315](#bookmark266), [318](#bookmark267), [319](#bookmark159), [321](#bookmark160), [323](#bookmark43), [328](#bookmark161), [330](#bookmark347), [332](#bookmark420), [336](#bookmark273), [337](#bookmark162), [348](#bookmark421), [349](#bookmark422), [357](#bookmark279)[–360](#bookmark280), [362](#bookmark281), [365](#bookmark423), [366](#bookmark424),  [368](#bookmark425)[–372](#bookmark283), [374](#bookmark284), [377](#bookmark126)[–382](#bookmark426), [384](#bookmark427), [387](#bookmark289), [388](#bookmark290), [391](#bookmark291), [394](#bookmark292), [396](#bookmark293), [399](#bookmark294), [400](#bookmark295), [403](#bookmark296)[–405](#bookmark297), [409](#bookmark299), [410](#bookmark300), [412](#bookmark324), [413](#bookmark301), [415](#bookmark428), [417](#bookmark429), [421](#bookmark302), [423](#bookmark303)[–427](#bookmark177), [429](#bookmark178), [430](#bookmark305), [435](#bookmark430), [438](#bookmark352)[–444](#bookmark308), [446](#bookmark309)[–449](#bookmark431), [451](#bookmark432), [452](#bookmark433), [454](#bookmark434), [456](#bookmark435), [458](#bookmark436)[–460](#bookmark310), [462](#bookmark311), [463](#bookmark437), [465](#bookmark438)[–469](#bookmark439), [471](#bookmark180), [472](#bookmark313), [474](#bookmark314), [476](#bookmark315)[–483](#bookmark17)]  *Workﬂow composition (8%)*  Fixed template 37% [[71](#bookmark316), [129](#bookmark15), [135](#bookmark114), [212](#bookmark119), [222](#bookmark151), [294](#bookmark94), [297](#bookmark95), [343](#bookmark99), [361](#bookmark101), [375](#bookmark102), [408](#bookmark103), [445](#bookmark105), [473](#bookmark107)]  Variable template 63% [[57](#bookmark109), [61](#bookmark82), [69](#bookmark84), [77](#bookmark14), [97](#bookmark85), [121](#bookmark86), [122](#bookmark87), [158](#bookmark89), [204](#bookmark90), [207](#bookmark91), [211](#bookmark149), [215](#bookmark150), [273](#bookmark92), [285](#bookmark21), [295](#bookmark122), [307](#bookmark22), [311](#bookmark96), [331](#bookmark97), [333](#bookmark98), [356](#bookmark100), [433](#bookmark179), [484](#bookmark131)]  *Supporting tasks (7%)*  Supporting AS 23% [[58](#bookmark135), [109](#bookmark141), [164](#bookmark144), [278](#bookmark93), [326](#bookmark270), [346](#bookmark167), [416](#bookmark440)]  Supporting HPO 63% [[67](#bookmark137), [171](#bookmark441), [174](#bookmark223), [180](#bookmark442), [210](#bookmark235), [260](#bookmark378), [269](#bookmark155), [270](#bookmark156), [301](#bookmark343), [329](#bookmark271), [335](#bookmark443), [339](#bookmark274), [347](#bookmark275), [376](#bookmark285), [383](#bookmark444), [391](#bookmark291), [392](#bookmark171), [416](#bookmark440), [453](#bookmark357)] Supporting AWC 17% [[68](#bookmark445), [207](#bookmark91), [298](#bookmark446), [422](#bookmark447), [428](#bookmark104)]  *Ad hoc tasks (4%)*  [50, 62, 72, 106, 111, 119, 203, 206, 248, 264, 268, 293, 341, 389, 390, 402, 461] |

**Table 4** Techniques applied by AutoML primary studies

1 灬

Eight years of AutoML: categorisation, review and trends 5113

|  |  |
| --- | --- |
| Category % | Papers |
| *Search and optimisation (SO) (74%)*  Evolutionary algorithms 35% | [[40](#bookmark390), [41](#bookmark393), [44](#bookmark367), [49](#bookmark184), [51](#bookmark134), [57](#bookmark109), [60](#bookmark187), [63](#bookmark136), [74](#bookmark110), [75](#bookmark191), [77](#bookmark14), [78](#bookmark192), [81](#bookmark448), [82](#bookmark397), [86](#bookmark193), [88](#bookmark194), [90](#bookmark449), [92](#bookmark327), [96](#bookmark196), [100](#bookmark198), [102](#bookmark399), [104](#bookmark328), [105](#bookmark20), [119](#bookmark113), [121](#bookmark86), [122](#bookmark87), [125](#bookmark205), [127](#bookmark206), [144](#bookmark371), [153](#bookmark450), [156](#bookmark451), [169](#bookmark221), [172](#bookmark222), [186](#bookmark225), [193](#bookmark148), [201](#bookmark337), [210](#bookmark235), [211](#bookmark149), [214](#bookmark236), [221](#bookmark238), [230](#bookmark411), [234](#bookmark242), [240](#bookmark452), [243](#bookmark338)[–246](#bookmark412), [249](#bookmark246), [250](#bookmark413), [252](#bookmark247), [256](#bookmark414)[–259](#bookmark415), [262](#bookmark249), [265](#bookmark153), [271](#bookmark251), [272](#bookmark157), [279](#bookmark254), [281](#bookmark453), [284](#bookmark340), [286](#bookmark256), [288](#bookmark257), [292](#bookmark259), [307](#bookmark22), [309](#bookmark264), [311](#bookmark96), [315](#bookmark266), [319](#bookmark159), [328](#bookmark161), [332](#bookmark420), [336](#bookmark273), [337](#bookmark162), [342](#bookmark165), [343](#bookmark99), [345](#bookmark166), [348](#bookmark421), [351](#bookmark276), [358](#bookmark454), [360](#bookmark280), [366](#bookmark424), [370](#bookmark323)[–372](#bookmark283), [378](#bookmark286), [379](#bookmark170), [381](#bookmark455), [382](#bookmark426), [384](#bookmark427), [390](#bookmark127), [391](#bookmark291), [400](#bookmark295), [401](#bookmark174), [406](#bookmark365), [409](#bookmark299), [410](#bookmark300), [421](#bookmark302), [426](#bookmark304), [429](#bookmark178), [439](#bookmark456), [440](#bookmark457), [443](#bookmark458), [444](#bookmark308), [451](#bookmark432), [454](#bookmark434), [458](#bookmark436), [460](#bookmark310), [462](#bookmark311), [465](#bookmark438), [468](#bookmark459), [473](#bookmark107), [475](#bookmark325), [478](#bookmark460), [479](#bookmark461), [484](#bookmark131)] |
| Sequential model-based opt 31% | [[42](#bookmark366), [44](#bookmark367), [45](#bookmark395), [48](#bookmark183), [52](#bookmark185), [56](#bookmark186), [60](#bookmark187), [61](#bookmark82), [70](#bookmark326), [71](#bookmark316), [73](#bookmark190), [96](#bookmark196), [103](#bookmark199), [108](#bookmark329), [110](#bookmark200), [115](#bookmark462), [116](#bookmark463), [123](#bookmark204), [129](#bookmark15), [135](#bookmark114), [141](#bookmark212), [143](#bookmark370), [144](#bookmark371), [146](#bookmark330), [151](#bookmark372), [159](#bookmark331)[–162](#bookmark333), [165](#bookmark219), [176](#bookmark363), [178](#bookmark373), [182](#bookmark374), [192](#bookmark229), [194](#bookmark230), [195](#bookmark231), [197](#bookmark232)[–199](#bookmark336), [202](#bookmark233), [204](#bookmark90), [208](#bookmark234)[–210](#bookmark235), [212](#bookmark119), [215](#bookmark150), [222](#bookmark151)[–225](#bookmark376), [231](#bookmark240), [239](#bookmark464), [242](#bookmark465), [247](#bookmark245), [255](#bookmark377), [267](#bookmark154), [273](#bookmark92), [274](#bookmark252), [276](#bookmark379), [294](#bookmark94), [296](#bookmark260), [297](#bookmark95), [299](#bookmark342), [300](#bookmark261), [304](#bookmark344), [306](#bookmark262), [321](#bookmark160), [322](#bookmark269), [331](#bookmark97), [333](#bookmark98), [334](#bookmark272), [336](#bookmark273), [350](#bookmark380), [354](#bookmark348)[–356](#bookmark100), [361](#bookmark101)[–364](#bookmark12), [367](#bookmark349), [369](#bookmark282), [373](#bookmark382), [375](#bookmark102), [376](#bookmark285), [385](#bookmark288), [398](#bookmark356), [406](#bookmark365), [414](#bookmark383), [415](#bookmark428), [418](#bookmark351)[–420](#bookmark384), [423](#bookmark303), [431](#bookmark385)[–434](#bookmark307), [436](#bookmark386), [455](#bookmark388), [457](#bookmark358), [470](#bookmark312), [482](#bookmark130)] |
| Gradient methods 18% | [[43](#bookmark394), [64](#bookmark188), [80](#bookmark466), [85](#bookmark467), [87](#bookmark138), [94](#bookmark195), [113](#bookmark468), [117](#bookmark202), [120](#bookmark203), [134](#bookmark209), [137](#bookmark469), [149](#bookmark215), [152](#bookmark403), [155](#bookmark470), [163](#bookmark218), [166](#bookmark220), [167](#bookmark145), [173](#bookmark318), [175](#bookmark405), [177](#bookmark406), [188](#bookmark335), [191](#bookmark147), [220](#bookmark471), [241](#bookmark472), [246](#bookmark412), [248](#bookmark322), [261](#bookmark416),  [266](#bookmark250), [290](#bookmark258), [291](#bookmark473), [305](#bookmark419), [310](#bookmark345), [312](#bookmark346)[–314](#bookmark474), [320](#bookmark268), [365](#bookmark423), [380](#bookmark287), [382](#bookmark426), [386](#bookmark350)[–388](#bookmark290), [396](#bookmark293), [403](#bookmark296), [413](#bookmark301), [417](#bookmark429), [427](#bookmark177), [435](#bookmark430), [441](#bookmark475), [445](#bookmark105)[–447](#bookmark476), [449](#bookmark431), [459](#bookmark477), [463](#bookmark437), [466](#bookmark478), [469](#bookmark439), [472](#bookmark313), [480](#bookmark479)] |
| Multi-armed bandit 7% Swarm intelligence 6% Random search 4% Greedy search 3% AI planning 2% Other SO techniques 9% | [[44](#bookmark367), [61](#bookmark82), [79](#bookmark480), [80](#bookmark466), [96](#bookmark196), [123](#bookmark204), [124](#bookmark400), [144](#bookmark371), [176](#bookmark363), [179](#bookmark364), [184](#bookmark334), [189](#bookmark227), [229](#bookmark11), [231](#bookmark240), [247](#bookmark245), [311](#bookmark96), [339](#bookmark274), [346](#bookmark167), [352](#bookmark168), [356](#bookmark100), [361](#bookmark101), [373](#bookmark382), [482](#bookmark130)] [[76](#bookmark317), [93](#bookmark112), [126](#bookmark142), [130](#bookmark207), [131](#bookmark143), [142](#bookmark401), [183](#bookmark224), [186](#bookmark225), [192](#bookmark229), [200](#bookmark116), [218](#bookmark481), [219](#bookmark482), [235](#bookmark152), [251](#bookmark391), [283](#bookmark158), [284](#bookmark340), [289](#bookmark341), [359](#bookmark483), [377](#bookmark126), [404](#bookmark175), [405](#bookmark297)]  [[45](#bookmark395), [54](#bookmark484), [61](#bookmark82), [96](#bookmark196), [133](#bookmark369), [144](#bookmark371), [176](#bookmark363), [179](#bookmark364), [190](#bookmark228), [228](#bookmark410), [311](#bookmark96), [424](#bookmark485), [463](#bookmark437)]  [[69](#bookmark84), [96](#bookmark196), [106](#bookmark79), [170](#bookmark404), [181](#bookmark407), [226](#bookmark409), [248](#bookmark322), [264](#bookmark120), [374](#bookmark284), [395](#bookmark172), [463](#bookmark437)] [[62](#bookmark83), [204](#bookmark90), [207](#bookmark91), [285](#bookmark21), [295](#bookmark122)]  [[45](#bookmark395), [46](#bookmark182), [61](#bookmark82), [98](#bookmark197), [118](#bookmark74), [119](#bookmark113), [144](#bookmark371), [150](#bookmark402), [151](#bookmark372), [154](#bookmark486), [164](#bookmark144), [185](#bookmark146), [210](#bookmark235), [216](#bookmark237), [227](#bookmark487), [254](#bookmark248), [333](#bookmark98), [338](#bookmark163), [353](#bookmark277), [398](#bookmark356), [408](#bookmark103), [415](#bookmark428), [423](#bookmark303), [438](#bookmark352), [448](#bookmark488), [453](#bookmark357), [462](#bookmark311), [464](#bookmark389)] |
| *Machine learning (ML) (41%)*  Transfer learning 48% | [[39](#bookmark181), [43](#bookmark394), [52](#bookmark185), [54](#bookmark484), [55](#bookmark396), [78](#bookmark192), [80](#bookmark466), [82](#bookmark397), [84](#bookmark398), [85](#bookmark467), [88](#bookmark194), [90](#bookmark449), [91](#bookmark111), [94](#bookmark195), [101](#bookmark489), [112](#bookmark201)[–114](#bookmark490), [130](#bookmark207), [137](#bookmark469), [138](#bookmark491), [144](#bookmark371), [152](#bookmark403), [153](#bookmark450), [163](#bookmark218), [165](#bookmark219), [166](#bookmark220), [175](#bookmark405), [177](#bookmark406), [180](#bookmark442),  [188](#bookmark335), [217](#bookmark492), [220](#bookmark471), [224](#bookmark408), [228](#bookmark410), [233](#bookmark241), [236](#bookmark243), [238](#bookmark244), [241](#bookmark472), [243](#bookmark338), [246](#bookmark412), [247](#bookmark245), [258](#bookmark493), [282](#bookmark255), [302](#bookmark417), [303](#bookmark418), [312](#bookmark346), [314](#bookmark474), [318](#bookmark267), [321](#bookmark160), [323](#bookmark43), [332](#bookmark420), [349](#bookmark422), [360](#bookmark280), [365](#bookmark423), [368](#bookmark425), [374](#bookmark284), [380](#bookmark287), [385](#bookmark288), [387](#bookmark289), [388](#bookmark290), [399](#bookmark294), [403](#bookmark296), [409](#bookmark299), [413](#bookmark301), [415](#bookmark428), [417](#bookmark429), [421](#bookmark302), [426](#bookmark304), [427](#bookmark177), [430](#bookmark305), [435](#bookmark430), [441](#bookmark475), [446](#bookmark309), [448](#bookmark488), [449](#bookmark431), [451](#bookmark432), [455](#bookmark388), [456](#bookmark435), [459](#bookmark477), [462](#bookmark311), [463](#bookmark437), [466](#bookmark478), [468](#bookmark459), [469](#bookmark439), [472](#bookmark313), [483](#bookmark17)] |
| Meta-learning 24% | [[38](#bookmark132), [58](#bookmark135), [59](#bookmark81), [67](#bookmark137), [84](#bookmark398), [95](#bookmark139), [99](#bookmark140), [120](#bookmark203), [128](#bookmark8), [129](#bookmark15), [134](#bookmark209), [140](#bookmark88), [164](#bookmark144), [168](#bookmark75), [180](#bookmark442), [196](#bookmark359), [203](#bookmark117), [205](#bookmark319), [211](#bookmark149), [213](#bookmark320), [247](#bookmark245), [267](#bookmark154), [269](#bookmark155), [270](#bookmark156), [283](#bookmark158), [287](#bookmark77), [293](#bookmark121), [312](#bookmark346), [316](#bookmark123), [317](#bookmark124), [324](#bookmark353), [325](#bookmark354), [329](#bookmark271), [338](#bookmark163), [340](#bookmark164), [344](#bookmark9), [347](#bookmark275), [356](#bookmark100), [392](#bookmark171), [397](#bookmark173), [406](#bookmark365), [411](#bookmark176), [442](#bookmark494), [445](#bookmark105)] |

1 灬

**Table 4** continued

5 114 R Barb udo et a l.

|  |
| --- |
| Category % Papers |
| Reinforcement learning 24% [[39](#bookmark181), [53](#bookmark495), [55](#bookmark396), [65](#bookmark496), [66](#bookmark189), [83](#bookmark368), [84](#bookmark398), [89](#bookmark497), [91](#bookmark111), [114](#bookmark490), [124](#bookmark400), [138](#bookmark491), [139](#bookmark211), [145](#bookmark213), [157](#bookmark216), [158](#bookmark89), [173](#bookmark318), [187](#bookmark226), [206](#bookmark118), [217](#bookmark492), [236](#bookmark243)[–238](#bookmark244), [253](#bookmark339), [254](#bookmark248), [277](#bookmark253), [280](#bookmark498), [282](#bookmark255), [302](#bookmark417),  [303](#bookmark418), [314](#bookmark474), [318](#bookmark267), [323](#bookmark43), [375](#bookmark102), [396](#bookmark293), [399](#bookmark294), [437](#bookmark387), [465](#bookmark438), [471](#bookmark180), [474](#bookmark314), [476](#bookmark315), [477](#bookmark499), [483](#bookmark17)]  Ensemble learning 3% [[51](#bookmark134), [129](#bookmark15), [131](#bookmark143), [205](#bookmark319), [342](#bookmark165), [433](#bookmark179), [482](#bookmark130)]  Other ML techniques 16% [[40](#bookmark390), [50](#bookmark108), [69](#bookmark84), [89](#bookmark497), [109](#bookmark141), [111](#bookmark73), [115](#bookmark462), [132](#bookmark208), [144](#bookmark371), [148](#bookmark115), [190](#bookmark228), [261](#bookmark416), [263](#bookmark76), [275](#bookmark392), [313](#bookmark265), [327](#bookmark355), [339](#bookmark274), [357](#bookmark279), [365](#bookmark423), [376](#bookmark285), [383](#bookmark444), [389](#bookmark361), [402](#bookmark78), [405](#bookmark297), [428](#bookmark104), [440](#bookmark457), [443](#bookmark458),  [446](#bookmark309), [461](#bookmark129), [466](#bookmark478)]  *Others (12%)*  [[47](#bookmark133), [50](#bookmark108), [58](#bookmark135), [68](#bookmark445), [69](#bookmark84), [72](#bookmark362), [75](#bookmark191), [97](#bookmark85), [101](#bookmark489), [107](#bookmark72), [112](#bookmark201), [135](#bookmark114), [136](#bookmark210), [147](#bookmark214), [152](#bookmark403), [161](#bookmark332), [171](#bookmark441), [174](#bookmark223), [207](#bookmark91), [210](#bookmark235), [224](#bookmark408), [232](#bookmark321), [236](#bookmark243), [248](#bookmark322), [260](#bookmark378), [268](#bookmark80), [278](#bookmark93), [297](#bookmark95), [298](#bookmark446), [301](#bookmark343), [313](#bookmark265), [326](#bookmark270), [330](#bookmark347), [335](#bookmark443), [339](#bookmark274), [341](#bookmark125), [363](#bookmark381), [391](#bookmark291), [393](#bookmark128), [394](#bookmark292), [398](#bookmark356), [401](#bookmark174), [407](#bookmark298), [412](#bookmark324), [416](#bookmark440), [422](#bookmark447), [425](#bookmark500), [450](#bookmark106), [452](#bookmark433), [461](#bookmark129), [467](#bookmark501), [481](#bookmark502)] |

an identiﬁer for each study whose respective reference can be found in the *Primary Studies* bibliography.

**6.1 Automated phases**

Table [2](#bookmark69) shows the phases of the knowledge discovery process that have been automated by the primary studies. Notice that AutoML has not covered yet the automation of the entire process. Even so, a number of primary studies (13%) automate more than one phase. This is particularly common in—but not limited to—proposals that automate the preprocessing operations, usually together with the automation of the data mining phase. In general, not all phases are equally distributed, with the data mining phase receiving the most attention from the AutoML community with up to 93% of primary studies. Although to a lesser extent, AutoML proposals have also faced the automation of the preprocessing (14%) and postprocessing (1%) phases.

As for the preprocessing, the domain understanding phase is tackled by some proposals that support data scientists during data exploration [[107](#bookmark72), [111](#bookmark73), [118](#bookmark74), [168](#bookmark75), [263](#bookmark76), [287](#bookmark77), [402](#bookmark78)]. Regarding the dataset creation problem, it has only been considered by a limited number of proposals [[106](#bookmark79), [268](#bookmark80)]. Observe that more than 81% out of the studies about preprocessing are focused on cleaning and transforming the resulting datasets. More speciﬁcally, 48% of studies refers to data cleaning and preprocessing, and typically deal with data scaling [[331](#bookmark97), [343](#bookmark99), [375](#bookmark102), [428](#bookmark104), [450](#bookmark106), [473](#bookmark107)], missing values handling [[106](#bookmark79), [129](#bookmark15), [273](#bookmark92), [445](#bookmark105)] or both [[61](#bookmark82), [69](#bookmark84), [122](#bookmark87), [158](#bookmark89), [207](#bookmark91), [285](#bookmark21), [297](#bookmark95), [333](#bookmark98), [356](#bookmark100), [361](#bookmark101), [408](#bookmark103)]. Note that, with the exception of [[59](#bookmark81), [62](#bookmark83), [74](#bookmark110), [140](#bookmark88), [278](#bookmark93)], these proposals are not exclusively dedicated to these particular activities, but they also cover the data mining phase. Data reduction and projection are also addressed (79%). In this case, some primary studies address the feature engineering [[50](#bookmark108),[119](#bookmark113),[203](#bookmark117),[206](#bookmark118),[264](#bookmark120),[293](#bookmark121),[341](#bookmark125), [390](#bookmark127), [461](#bookmark129)]. A reduced number of studies are particularly devoted to feature selection [[148](#bookmark115), [316](#bookmark123), [317](#bookmark124)]. As mentioned above, most of these primary studies automate some data mining phase too, such as classiﬁcation [[61](#bookmark82), [122](#bookmark87), [129](#bookmark15), [285](#bookmark21), [295](#bookmark122), [297](#bookmark95), [307](#bookmark22), [331](#bookmark97), [343](#bookmark99), [375](#bookmark102), [395](#bookmark172), [408](#bookmark103), [445](#bookmark105), [450](#bookmark106)], regression [[61](#bookmark82), [207](#bookmark91), [212](#bookmark119), [273](#bookmark92), [297](#bookmark95), [450](#bookmark106)] and clustering [[484](#bookmark131)]. The same happens with feature extraction, which is automated together with the data mining phase [[61](#bookmark82), [69](#bookmark84), [74](#bookmark110), [91](#bookmark111), [93](#bookmark112), [122](#bookmark87), [129](#bookmark15), [135](#bookmark114), [158](#bookmark89), [200](#bookmark116), [297](#bookmark95), [307](#bookmark22), [311](#bookmark96), [343](#bookmark99), [356](#bookmark100), [361](#bookmark101), [375](#bookmark102), [377](#bookmark126), [408](#bookmark103), [428](#bookmark104), [450](#bookmark106), [482](#bookmark130), [484](#bookmark131)]. Just a few studies [[59](#bookmark81), [74](#bookmark110), [393](#bookmark128)] are an exception in this case.

Turning speciﬁcally to the data mining phase, we should take into account whether it is an HPO proposal or not, since the former are usually independent of the data mining tech- nique being conducted. Thus, these studies have been classiﬁed according to the data mining problem(s) being addressed in their experimental validation. They are shown in Table [2](#bookmark69) as a separate list of references marked with the symbol † . Independently of the AutoML task carried out, we can observe that the automation of supervised learning, i.e. classiﬁcation and regression, has attracted most of the attention (91% and 14%, respectively) compared to unsupervised techniques like clustering and outlier detection (2% and <1%). In fact, to the best of our knowledge, other unsupervised problems, such as pattern mining or association rule mining, have not been explored yet in this ﬁeld. As motivated by Ferrari et al. [[128](#bookmark8)], most problems addressed with unsupervised methods lack a previously known solution set. Regardless of this point, most studies automating data mining are related to HPO. However, they rarely give support to more than one phase. The same does not apply for the studies addressing other AutoML tasks. Among them, focusing on supervised learning, proposals automating classiﬁcation or regression also consider some preprocessing methods (44% and 56%, respectively). More speciﬁcally, some automate the data reduction and projection phase

before building the classiﬁcation [[57](#bookmark109), [135](#bookmark114), [212](#bookmark119), [295](#bookmark122), [484](#bookmark131)] and regression [[212](#bookmark119)] models. The data cleaning and preprocessing has been considered too (83%), but always together with data reduction and projection, only with the exception of [[294](#bookmark94)]. In this case, they have been applied before the classiﬁcation [[61](#bookmark82), [69](#bookmark84), [121](#bookmark86), [122](#bookmark87), [129](#bookmark15), [158](#bookmark89), [204](#bookmark90), [207](#bookmark91), [273](#bookmark92), [285](#bookmark21), [297](#bookmark95), [307](#bookmark22), [311](#bookmark96), [331](#bookmark97), [333](#bookmark98), [343](#bookmark99), [356](#bookmark100), [361](#bookmark101), [375](#bookmark102), [408](#bookmark103), [428](#bookmark104), [445](#bookmark105), [473](#bookmark107)] or regression [[61](#bookmark82), [69](#bookmark84), [207](#bookmark91), [273](#bookmark92), [297](#bookmark95), [311](#bookmark96), [361](#bookmark101), [473](#bookmark107)] algorithms. Regarding the unsupervised learning, no primary study has considered any preprocessing method with the only exception of [[484](#bookmark131)], which automates the data reduction and projection before building the clusters.

Finally, the actions performed during the postprocessing phase are inherently human in nature, requiring a certain amount of experience, know-how and intuition. As a consequence, barely 1% of the primary studies automate this part of the knowledge discovery process. In particular, we found proposals dealing with the knowledge interpretation [[248](#bookmark322), [389](#bookmark361)] and

knowledge integration [[72](#bookmark362)] phases.

**6.2 AutoML tasks**

Automation can be conducted in different ways, depending on the operation or task per- formed (see Sect.[2.2)](#bookmark39). Following our taxonomy, Table [3](#bookmark70)lists the studies that carry out some primary or secondary task(s). As can be observed, there is a large difference in the number of contributions referring to each task, with HPO attracting by far the most attention (77% of primary studies). In contrast, a smaller number of proposals conduct secondary tasks (11%), from which 7% are dedicated to support one of the primary tasks,i.e. AS, HPO or WC.

Regarding the primary AutoML tasks, 9% of the reviewed manuscripts perform algorithm selection,i.e. they recommend the best preprocessing or ML algorithms for a given dataset. Despite AS has been largely studied, recent proposals have started to address new problems like clustering [[128](#bookmark8), [324](#bookmark353), [325](#bookmark354), [398](#bookmark356)] or data stream forecasting [[344](#bookmark9)]. Regardless of the automated phase, the most common strategy is to recommend a single algorithm (57%), but other approaches can also select an unordered set (20%) or a ranking (30%) of the best algorithms. Notice that some studies conduct more than one type of recommendation. On the other hand, HPO covers 77% out of the primary studies. According to our taxonomy, these proposals might be divided into two groups depending on whether they are independent of the algorithm whose hyper-parameters are tuned (25%) or not (75%). Also, NAS proposals, which represent the majority of studies (70%), have been differentiated from the rest of HPO algorithm-dependent approaches. It should be noted the existence of approaches optimising neuralnetworkswithoutconsideringtheirarchitecture[[40](#bookmark390),[189](#bookmark227),[251](#bookmark391),[266](#bookmark250),[310](#bookmark345),[386](#bookmark350),[393](#bookmark128),[475](#bookmark325)]. In addition, there are approaches optimising clustering algorithms [[327](#bookmark355), [398](#bookmark356)] and kernel- based algorithms, such as support vector machines [[76](#bookmark317), [283](#bookmark158), [289](#bookmark341), [343](#bookmark99), [401](#bookmark174)],graph kernels methods [[275](#bookmark392)], or conditional mean embeddings [[167](#bookmark145)].

Regarding the primary tasks, workﬂow composition involves 8% of primary studies. Again, this task can be further decomposed into two categories depending on whether the workﬂow structure has been preﬁxed in advance (37%) or not (63%). Regardless of this sub- category, all these proposals choose between different alternative algorithms like decision tree or SVMin order to perform each step of the workﬂow, e.g. classiﬁcation, with the excep- tion of [[71](#bookmark316)] that only optimises hyper-parameters. Most of these WC proposals optimise a workﬂow composed of one or more preprocessing methods and a data mining algorithm. In contrast, other approaches [[57](#bookmark109), [77](#bookmark14), [433](#bookmark179)] only consider classiﬁcation algorithms, thus deﬁn- ing a classiﬁer ensemble as a type of workﬂow outcome. Notice that some proposals have

not considered the optimisation of hyper-parameters [[57](#bookmark109), [69](#bookmark84), [207](#bookmark91), [484](#bookmark131)], even though some recognise it as an interesting future work [[295](#bookmark122)].

Of the 11% of studies that refer to secondary tasks, 66% explore a supporting task. For those supporting AS, some proposals identify the best meta-features describing a dataset [[58](#bookmark135), [109](#bookmark141), [326](#bookmark270)], build a portfolio of algorithms for a posterior selection [[164](#bookmark144)], reduce the execu- tion time of the task [[346](#bookmark167)], or assess the quality of an algorithm [[278](#bookmark93)]. As for the case of supporting HPO, these approaches have been focused on recommending whether it is worth investing resources in the task [[269](#bookmark155), [270](#bookmark156), [347](#bookmark275), [392](#bookmark171)], predicting whether a given conﬁguration will lead to better results [[329](#bookmark271), [335](#bookmark443), [391](#bookmark291)], identifying the most relevant hyper-parameter to be tuned [[260](#bookmark378), [339](#bookmark274), [376](#bookmark285)], distributing the optimisation process in a cluster [[453](#bookmark357)], or look- ing for speeding up the optimisation [[67](#bookmark137), [180](#bookmark442)]. Also, there are other approaches intended to assist NAS by shrinking the search space [[174](#bookmark223)], predicting the performance of an ANN architecture [[301](#bookmark343), [383](#bookmark444)] or ensuring diversity during the search [[171](#bookmark441)]. Finally, with respect to WC, some approaches aim at recommending the best knowledge discovery workﬂow [[207](#bookmark91), [428](#bookmark104)], speeding up the evaluation of workﬂows [[298](#bookmark446)], improving interpretability [[422](#bookmark447)], or generating a complete search space from a partially deﬁned one by adding new preprocess- ing and ML algorithms [[68](#bookmark445)]. On the other hand, ad hoc tasks suppose 34% of all primary studies covering a secondary task. Under this category, both preprocessing and postprocess- ing are considered. With respect to the former, there are approaches aimed at creating tabular datasets [[106](#bookmark79), [268](#bookmark80)], generating data visualisations from the datasets [[111](#bookmark73), [402](#bookmark78)], ﬁxing data inconsistencies [[62](#bookmark83)] and generating new features [[50](#bookmark108), [119](#bookmark113), [203](#bookmark117), [206](#bookmark118), [264](#bookmark120), [293](#bookmark121), [341](#bookmark125), [390](#bookmark127), [461](#bookmark129)]. As for the postprocessing step, approaches explaining the resulting ML models [[248](#bookmark322), [389](#bookmark361)] or generating their source code [[72](#bookmark362)] have been proposed as well.

**6.3 Applied techniques and methods**

Table [4](#bookmark71)compiles the techniques that have been applied in primary studies to carry out their respective AutoML tasksofinterest. Thesetechniques are dividedintotwo well-distinguished categories, search and optimisation (74%) and machine learning (41%), apart from other proposals that do not fall into any of these groups (12%). Also, it is worth noting that only the main techniques of each study are categorised.

Focusing on the searchandoptimisationcategory, evolutionaryalgorithms[[3](#bookmark503)]are the most frequent techniques (35%). Among them, genetic algorithms [[40](#bookmark390), [60](#bookmark187), [63](#bookmark136), [74](#bookmark110), [75](#bookmark191), [78](#bookmark192), [88](#bookmark194), [96](#bookmark196), [100](#bookmark198),[102](#bookmark399),[104](#bookmark328),[125](#bookmark205),[127](#bookmark206),[151](#bookmark372),[153](#bookmark450),[169](#bookmark221),[193](#bookmark148),[201](#bookmark337),[210](#bookmark235),[214](#bookmark236),[221](#bookmark238),[230](#bookmark411),[244](#bookmark504),[249](#bookmark246),[256](#bookmark414)[–259](#bookmark415),[265](#bookmark153), [271](#bookmark251),[284](#bookmark340),[286](#bookmark256),[292](#bookmark259),[309](#bookmark264),[311](#bookmark96),[315](#bookmark266),[336](#bookmark273),[348](#bookmark421),[358](#bookmark454),[360](#bookmark280),[366](#bookmark424),[370](#bookmark323),[378](#bookmark286),[379](#bookmark170),[384](#bookmark427),[400](#bookmark295),[401](#bookmark174),[406](#bookmark365), [439](#bookmark456),[454](#bookmark434),[458](#bookmark436),[475](#bookmark325)],grammatical evolution[[41](#bookmark393),[49](#bookmark184),[51](#bookmark134),[105](#bookmark20),[121](#bookmark86),[122](#bookmark87)],evolution strategy[[74](#bookmark110), [144](#bookmark371), [451](#bookmark432), [479](#bookmark461)] and differential evolution [[74](#bookmark110), [332](#bookmark420), [351](#bookmark276)] are the preferred alternatives. Also, it is common to ﬁnd other approaches based on genetic programming: canonical genetic programming [[57](#bookmark109), [211](#bookmark149), [307](#bookmark22), [337](#bookmark162), [390](#bookmark127)], Cartesian genetic programming [[119](#bookmark113), [371](#bookmark169), [372](#bookmark283)], grammar-guided genetic programming [[345](#bookmark166)] and strongly typed genetic programming [[484](#bookmark131)]. Meta-heuristicmethodslike swarmintelligence(6%)and greedy search(3%)are also applied. Particularly, alltheproposalsbased on swarm intelligencerely on particleswarm optimisation (PSO) algorithms, with the exception of [[183](#bookmark224), [192](#bookmark229), [289](#bookmark341)] that are founded on the behaviour of ants, bats and bees, respectively. Also, note that variants for multi-objective optimisation of these methods have been applied, more speciﬁcally to genetic algorithms [[60](#bookmark187), [63](#bookmark136), [78](#bookmark192), [88](#bookmark194), [153](#bookmark450), [169](#bookmark221), [244](#bookmark504), [249](#bookmark246), [256](#bookmark414)[–259](#bookmark415), [271](#bookmark251), [311](#bookmark96), [348](#bookmark421), [400](#bookmark295)], genetic programming [[307](#bookmark22), [484](#bookmark131)] and PSO [[93](#bookmark112), [200](#bookmark116), [283](#bookmark158), [404](#bookmark175)], among others. Other methods like gradient-based methods (18%), multi-armed bandit (7%), random search (4%) and AI planning (2%) can be found

among the primary studies. Finally, other search and optimisation techniques include bilevel programming [[134](#bookmark209)],racing algorithms [[464](#bookmark389)] or coordinate descent [[98](#bookmark197), [448](#bookmark488)], among others.

Sequential model-based optimisation [[32](#bookmark505)] is the second leading technique (31%). These studies can be categorised according to their surrogate model and acquisition function. As for the surrogate model, Gaussian processes are used by most proposals (48 out of 94). Other common alternatives are artiﬁcial neural networks [[116](#bookmark463), [242](#bookmark465), [321](#bookmark160), [350](#bookmark380), [362](#bookmark281), [364](#bookmark12), [367](#bookmark349)] and random forest [[60](#bookmark187), [70](#bookmark326), [71](#bookmark316), [129](#bookmark15), [144](#bookmark371), [204](#bookmark90), [212](#bookmark119), [267](#bookmark154), [273](#bookmark92), [297](#bookmark95), [373](#bookmark382), [398](#bookmark356)]. Regarding the acquisitionfunction,moststudiesapply expectedimprovement(41outof94). Inthis case, the upper conﬁdence bound[[103](#bookmark199),[194](#bookmark230),[198](#bookmark375),[202](#bookmark233),[231](#bookmark240),[299](#bookmark342),[300](#bookmark261),[385](#bookmark288)],lower conﬁdence bound[[60](#bookmark187), [71](#bookmark316), [304](#bookmark344)] and Thompson Sampling [[162](#bookmark333), [276](#bookmark379), [373](#bookmark382)] are also applied. In general, sequential model-based optimisation methods evaluate a single hyper-parameter conﬁguration at each iteration, although some approaches evaluate a batch of them [[108](#bookmark329), [110](#bookmark200), [146](#bookmark330), [194](#bookmark230), [202](#bookmark233), [299](#bookmark342), [354](#bookmark348),[369](#bookmark282),[436](#bookmark386)]. Finally, it is worth noting the existence of studies that are specially committed to the deﬁnition of new surrogate models (8 out of 94) and/or acquisition functions (25 out of 94).

As for the machine learning category, transfer learning is the most recurrent technique (48%). Nevertheless,it is often used in conjunction with others techniques, such as gradient- based methods (34 out of 87),reinforcement learning (17) and evolutionary algorithms (17). Meta-learning, together with reinforcement learning, is the second leading technique (24%). These proposals can be divided according to the method that is used at the meta-level. Here, classiﬁcation algorithms, such as decision trees [[140](#bookmark88), [205](#bookmark319), [269](#bookmark155), [270](#bookmark156), [316](#bookmark123), [317](#bookmark124), [326](#bookmark270)], neural networks [[99](#bookmark140),[168](#bookmark75),[213](#bookmark320),[269](#bookmark155),[287](#bookmark77),[293](#bookmark121),[317](#bookmark124)],random forest [[95](#bookmark139),[203](#bookmark117),[269](#bookmark155),[270](#bookmark156),[317](#bookmark124),[324](#bookmark353),[326](#bookmark270), [344](#bookmark9)], support vector machines [[99](#bookmark140), [196](#bookmark359), [269](#bookmark155), [270](#bookmark156), [326](#bookmark270)],naive bayes [[269](#bookmark155), [270](#bookmark156), [344](#bookmark9)],and k- nearest neighbors [[67](#bookmark137),[99](#bookmark140),[267](#bookmark154),[269](#bookmark155),[270](#bookmark156),[324](#bookmark353),[411](#bookmark176)], are the preferred alternatives. Also, other regression algorithms, such as random forest [[58](#bookmark135), [59](#bookmark81), [329](#bookmark271)], support vector regression [[338](#bookmark163)], decision trees [[329](#bookmark271)], Gaussian processes [[270](#bookmark156), [393](#bookmark128)] or XGBoost [[397](#bookmark173)], are considered. In fact, Sanderset al. [[347](#bookmark275)] consider both classiﬁcation and regression algorithms during the learning. Although to a lesser extent, clustering algorithms [[325](#bookmark354), [442](#bookmark494)] have been used for the recommendation too. There are methods [[38](#bookmark132),[128](#bookmark8),[340](#bookmark164)] that take advantage of the similarities between datasets—measured by a set of meta-features—without relying on the aforemen- tioned data mining methods. Other recurrent ML approach is ensemble learning (3%), which is commonly applied to combine the best models found during AS [[205](#bookmark319), [342](#bookmark165)], NAS [[51](#bookmark134), [131](#bookmark143), [482](#bookmark130)]or WC [[129](#bookmark15), [433](#bookmark179)]. Finally, others ML techniques (16%) include a variety of techniques like inductive logic programming [[389](#bookmark361)] or multiple kernel learning [[275](#bookmark392)], among others.

In Table [4](#bookmark71), the category “*Others*” includes those proposals not applying neither SO nor ML techniques. It should be noted the existence of a wide range of techniques being applied by these proposals like collaborative ﬁltering [[135](#bookmark114), [207](#bookmark91)],model checking [[278](#bookmark93)], case-based reasoning [[97](#bookmark85)], data envelopment analysis [[401](#bookmark174)], conditional parallel coordinates [[422](#bookmark447)] or fractional factorial design [[260](#bookmark378)], among others. As the study of the area progresses, we envision that this category may be reformulated or broken down into other categories and integrated into the taxonomy.

**6.4 Experimental framework and additional framework**

Presenting an experimental framework to assess the validity of the proposed approach is a common practice in most primary studies (99%). In this section, we analyse the main com- ponents of the experimentation, including the type of case study, the comparative framework

**Table 5** More relevant repositories in GitHub

|  |  |  |  |
| --- | --- | --- | --- |
| Reference | #Commits | Last commit | Popularity |
| [307] | 2368 | 06/01/2021 | 8497 |
| [44] | 11,877 | 31/03/2022 | 6157 |
| [ 129] | 2617 | 18/02/2022 | 6117 |
| [ 144] | 165 | 11/11/2019 | 1447 |
| [368] | 9 | 08/04/2019 | 391 |

or the use of statistical validation. Empirical studies,i.e. those setting a hypothesis and con- structing a method to qualitatively or quantitatively validate it, and discussing ﬁndings that support such a hypothesis are a consolidated practice in the area (98%), in contrast to purely theoretical proposals (1%) or the use of sample executions (1%). A combination of a sample execution with some empirical validation can be found too [[292](#bookmark259)].

Intermsofthe*casestudy*usedforthe experimental validation,thereis a clearpreferencefor those benchmarks and datasets obtained from the literature (98%). The UC Irvine Machine Learning[3](#bookmark507) and the OpenML[4](#bookmark508) are well-known and referenced repositories, as well as the CIFAR dataset[5](#bookmark509) in the case of NAS proposals. In contrast, only 6% out of the primary studies generate their synthetic dataset. In fact, notice that, with the exception of [[161](#bookmark332), [196](#bookmark359), [350](#bookmark380), [357](#bookmark279), [428](#bookmark104)], these studies combine their own datasets with others taken from the literature. Regardless of the datasets being used, the analysis of the *comparison framework* reveals that 93% of the primary studies compare their proposals against other approaches from the literature. In this vein, AutoML approaches are the preferred option (69%) for comparison. However, a large number of primary studies compare their results against other proposals not belonging to this speciﬁc area. For example, NAS approaches can be compared against hand-crafted architectures. Furthermore, we found that 47% out of studies are compared against variants of themselves.

Finally, *statistical methods* allows producing trustworthy analyses and reliable results during the comparison. However, it is notorious that only 19% of primary studies are sup- ported by statistical evidence. Pairwise comparison (14%) and multiple comparison (6%) analyses are the most frequent choices. Regarding the pairwise comparison, the t-testis the preferred option (30 out of 62), as well as the Mann–Whitney-U test (19) and the Wilcoxon signed-rank test (16). As for multiple comparison, the Friedman (20 out of 27) and Nemenyi tests (9) are the most recurrent. Some primary studies also combine both types of statistical tests [[95](#bookmark139), [150](#bookmark402), [161](#bookmark332), [196](#bookmark359), [222](#bookmark151), [270](#bookmark156), [316](#bookmark123), [317](#bookmark124), [372](#bookmark283), [390](#bookmark127), [434](#bookmark307)]. Finally, it is worth noting that some manuscripts claim that they conduct some sort of statistical analysis, but they do not describe the methodology explicitly [[98](#bookmark197), [164](#bookmark144), [260](#bookmark378), [352](#bookmark168), [396](#bookmark293), [422](#bookmark447)].

We have alsoinspectedthe additionalmaterialmade availableby theauthors oftheprimary study, as a way to check their replicability and reproducibility. Software artefacts are publicly available in 33% of cases. From those, sharing the source code is a common practice (96%), as well as providing some manuals, installation instructions or software guides (77%). In this context, Table [5](#bookmark506) shows the top-5 GitHub repositories deployed in the primary studies[.6](#bookmark510) On the one hand, [[307](#bookmark22)] and [[129](#bookmark15)] are related to TPOT and auto-sklearn, respectively, which

3 UCI Machine Learning Repository: <https://archive.ics.uci.edu/> (last accessed: 30/08/2022).

4 OpenML repository: <https://www.openml.org/>(last accessed: 30/08/2022).

5 CIFAR dataset: <https://www.cs.toronto.edu/~kriz/cifar.html/> (last accessed: 30/08/2022).

6 Github accessed on March 31, 2022. Popularity of a repository is measured in terms of its number of starts. Those repositories providing the source code of more than one project/paper are omitted.



Ad hoc tasks -

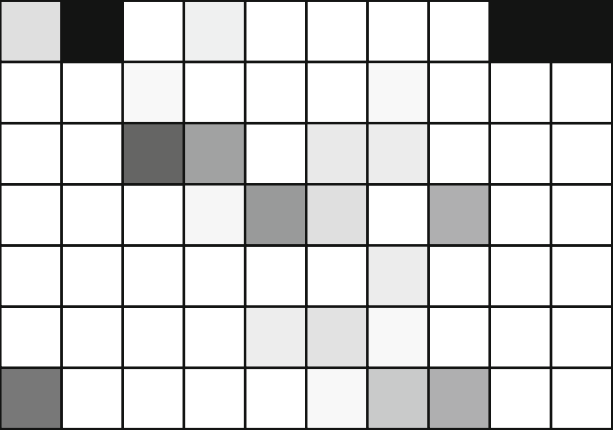
supporting tasks -

workflow composition -

Neural architecture search 

HPO - Algorithm dependent - HPO - Algorithm independent - Algorithm selection -

2





3 2 26 2 1

  33 13 2

6  17

1 9 5 2

1 1 76 16 1

2 4 18 6 4





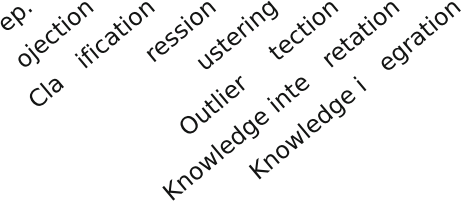


0 . 8 0 . 7



0 . 5 0 . 4 0 . 3 0 . 2



心



**Fig. 5** Relation between phases and tasks

are approaches for workﬂow composition. On the other hand, [[44](#bookmark367)] and [[144](#bookmark371)] correspond to tools implementing different methods for hyper-parameter optimisation. Finally, [[368](#bookmark425)] proposes a NAS method. It is worth noting that some tools [[44](#bookmark367),[129](#bookmark15)] are still under continuous development according to the frequency of commits.

Focusing on those studies providing software artefacts, in terms of their practicability and usability, it is worth noting that only 11% out of studies provide a software with an end-user friendly graphical user interface, as opposed to those proposals based on a command-line interface (73%). Moreover, in addition to software artefacts, some primary studies (36%) include other types of additional information that complement the manuscript. Among them, the most frequent are the incorporation of reports and attached documents (79%), datasets provided for replicability purposes (4%) or raw data of the experimental outcomes (19%). Finally, it is worth noting the existence of other types of supplementary material (17%) like videos [[152](#bookmark403), [161](#bookmark332), [168](#bookmark75), [226](#bookmark409), [247](#bookmark245), [267](#bookmark154), [277](#bookmark253), [280](#bookmark498), [281](#bookmark453), [304](#bookmark344), [311](#bookmark96), [403](#bookmark296), [428](#bookmark104), [449](#bookmark431)],online demo tools [[111](#bookmark73), [329](#bookmark271)] or interactive visualisation of results [[350](#bookmark380), [432](#bookmark306)].

**7 Cross-category data analysis**

To complement the analysis in response to RQ3, we have conducted a cross-analysis of the relationships between the different perspectives of our taxonomy: phases, tasks and techniques. This is intended to identify trends, gaps or potential limitations in the cur- rent application of AutoML. First, we contrast the data on automated phases with the tasks performed (see Sect.[7.1)](#bookmark512). Then, we examine possible relationships between tasks and the techniques applied (see Sect.[7.2)](#bookmark513).

**7.1 Phases and tasks**

The graphical representation in Fig.[5](#bookmark511) illustrates the frequency with which speciﬁc tasks (rows) have been used to automate various phases (columns) of the knowledge discovery process. It is important to note that this ﬁgure presents a grid of all possible combinations

1 灬

of phases and tasks, which may include some that appear unlikely or even impossible. The relative frequency of application of each task for a phase is depicted following a coloured pattern, where black cells represent frequency equal to 1, i.e. only the referred task has been used in the context of the phase. Cells contain the number of primary studies found for each task-phase pair. As can be observed, if we group phases into preprocessing, data mining and postprocessing, there is no clear trend in terms of which tasks are applied in each category, although the marginal use of ad hoc tasks in the data mining phase can be highlighted, while their use is exclusive in the case of postprocessing.

Focusing on preprocessing, note that domain understanding—an inherently human activity—has been only automated by means of AS (5) and ad hoc (2) tasks, focused on selecting [[107](#bookmark72), [118](#bookmark74), [168](#bookmark75), [263](#bookmark76), [287](#bookmark77)] or explicitly generating [[111](#bookmark73), [402](#bookmark78)] the best visualisa- tion technique(s) according to the properties of the dataset. Ad hoc tasks are also deﬁned to automate the creation of a tabular dataset with the attribute-value format required by most ML algorithms. In this vein, De Raedt et al. [[106](#bookmark79)] generate a tabular dataset from a dataset formed by a collection of worksheets containing likely incomplete data. Similarly, Mahuele et al. [[268](#bookmark80)] have used linked open data to build this type of datasets. A wider range of tasks have been conducted to automatedata cleaning and preprocessing, as well as data reduction and projection. However, both phases coincide in the predominant use of workﬂow compo- sition tasks and in the limited use or lack of use of HPO tasks. Regarding WC, algorithms handling missing values, normalising data or extracting and selecting features are commonly used for composing workﬂows. As for HPO, a number of primary studies [[74](#bookmark110), [93](#bookmark112), [200](#bookmark116), [377](#bookmark126)] are considered for data reduction using a NAS method to optimise the architecture of an autoencoder. Having model selection as a primary task, algorithm selection appears in both phases as well. In this case, studies are focused on recommending the best label noise ﬁl- ter [[140](#bookmark88)], providing a feature selection method [[148](#bookmark115), [316](#bookmark123), [317](#bookmark124)] and, more generally, the best preprocessing algorithm including feature extraction, normalisation, discretisation and miss- ing values methods [[59](#bookmark81)]. Also, although they are not a majority in theirrespective phases, some primary studies present proposals that perform secondary supporting tasks to evaluate the effectiveness of a cleansing function [[278](#bookmark93)] or to recommend workﬂows involving both phases [[207](#bookmark91), [428](#bookmark104)]. Ad hoc tasks are proposed to ﬁx data inconsistencies [[62](#bookmark83)] and handle missing values in the context of data cleaning [[106](#bookmark79)]. As for data reduction and projection, they are also considered for feature generation either by combining those existing within the original dataset [[57](#bookmark109), [74](#bookmark110), [119](#bookmark113), [203](#bookmark117), [206](#bookmark118), [264](#bookmark120), [293](#bookmark121), [390](#bookmark127), [461](#bookmark129)] or by extracting them from other structured [[341](#bookmark125)] or unstructured [[50](#bookmark108)] external sources.

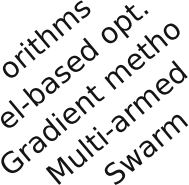
In the case of the data mining phase, it can be observed that, regardless of the particu- lar task, classiﬁcation concentrates the largest number of contributions (391), followed by regression (59). In both cases, the predominance of proposals that carry out NAS is notice- able (229 and 17, respectively). It is worth noting that 70% out of all these NAS proposals have been published since 2020. Also, the use of algorithm-independent proposals for HPO in both classiﬁcation and regression is relevant. However, only [[165](#bookmark219), [194](#bookmark230), [247](#bookmark245), [304](#bookmark344), [367](#bookmark349), [418](#bookmark351), [450](#bookmark106)] consider both phases in their experimental frameworks. For the rest of primary tasks, workﬂow composition has a relative importance, mostly for classiﬁcation. In fact, with the exception of [[71](#bookmark316)], the studies that automate regression also consider classiﬁcation. Then, focusing on AS, it has covered the three subcategories described in our taxonomy, i.e. single [[57](#bookmark109), [58](#bookmark135), [99](#bookmark140), [162](#bookmark333), [164](#bookmark144), [176](#bookmark363), [179](#bookmark364), [213](#bookmark320), [248](#bookmark322), [267](#bookmark154), [342](#bookmark165), [345](#bookmark166), [352](#bookmark168), [395](#bookmark172), [406](#bookmark365)], unordered set [[205](#bookmark319), [338](#bookmark163), [344](#bookmark9), [411](#bookmark176)] and ranking [[38](#bookmark132), [47](#bookmark133), [95](#bookmark139), [340](#bookmark164), [397](#bookmark173)]. Regarding HPO, neural net- works [[40](#bookmark390), [251](#bookmark391), [266](#bookmark250), [310](#bookmark345), [386](#bookmark350), [475](#bookmark325)] and SVM [[63](#bookmark136), [76](#bookmark317), [283](#bookmark158), [289](#bookmark341), [401](#bookmark174)] hyper-parameters are commonlytuned. Inthe case ofclustering andoutlierdetection,algorithmselectionis themost studied task, which has been committed to recommend a single [[196](#bookmark359), [398](#bookmark356)] or a ranking [[128](#bookmark8),

Ad hoc tasks

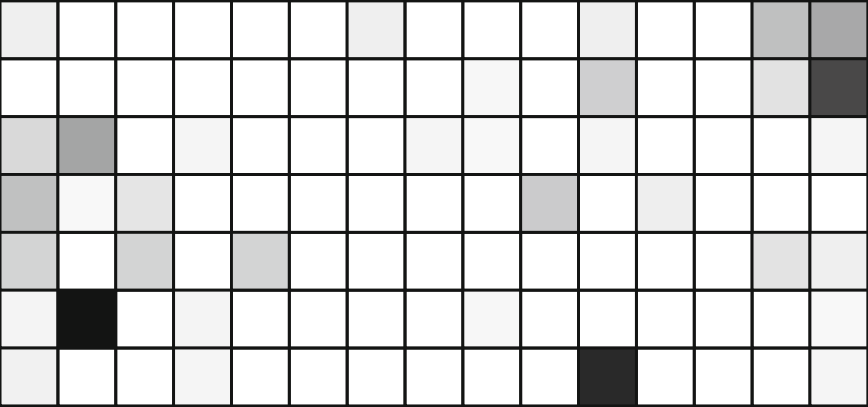
supporting tasks

workflow composition

Neural architecture search

HPO - Algorithm dependent HPO - Algorithm independent Algorithm selection





 2 2 1 1 2 1 5 

 1 1 2 1 7 5 

- lo  1 4 2  4 3 4 2 2 1 4

- 90 20 51 5 17 7 6 13 81 4 38 3 13 16

 4 1 4 1 4 1 3 2

- 9  3 9 4 7 5 2 2 1 6

 4 2 3 2 2  2 2 3





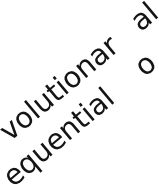
- 0 . 5

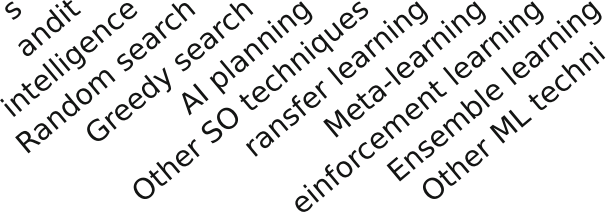
- 0 . 4

- 0 . 3

- 0 . 2





人





**Fig. 6** Relationships between tasks and techniques

[324](#bookmark353), [325](#bookmark354)] of algorithms. Particularly, the only primary studies found for outlier detection are [[196](#bookmark359)] and [[237](#bookmark360)], which builds a SVM model to recommend the best outlier detection method between 14 alternatives and deﬁnes a NAS method for their detection, respectively. As for clustering, it has been also automated by HPO [[327](#bookmark355), [398](#bookmark356), [457](#bookmark358)] and WC [[71](#bookmark316), [484](#bookmark131)] studies. Finally, there is no primary study presenting an ad hoc task in this phase, but we could ﬁnd some supporting tasks for AS, HPO and WC, mainly for classiﬁcation. In fact, the classiﬁcation setting is also considered by the approaches addressing regression [[207](#bookmark91), [301](#bookmark343)] and clustering [[453](#bookmark357)] problems.

As mentioned inSect.[6.1](#bookmark65), postprocessing activities have been only marginally automated, and always by means of ad hoc tasks. On the one hand, knowledge interpretation has been automated with the aim of explaining machine learning models by exploiting linked open data [[389](#bookmark361)] and generating natural language descriptions [[248](#bookmark322)]. On the other hand, Castro- Lopez et al. [[72](#bookmark362)] deﬁned an approach for knowledge integration that automatically generates the source code of a machine learning model.

**7.2 Tasks and techniques**

Similar to Sect.[7.1](#bookmark512), Fig.[6](#bookmark514) shows how often a certain task (rows) has been performed by applying a given technique (columns). Again, the coloured pattern represents the relative frequency of automation of each task by a technique (the darker, the closer to 1). As can be observed, evolutionary algorithms, sequential model-based optimisation (i.e. Bayesian optimisation), multi-armed bandit and meta-learning are the most recurrent techniques to address the largest number of tasks. Next, we discuss all techniques by each particular task. Focusing on algorithm selection, meta-learning is the most common option (53%), regard- less of the number of algorithms being recommended. Other techniques like evolutionary algorithms [[57](#bookmark109), [342](#bookmark165), [345](#bookmark166), [406](#bookmark365)] and greedy search [[248](#bookmark322), [395](#bookmark172)] have been also applied. In contrast, recommending an unordered set of algorithms has been performed only with meta- learning techniques [[140](#bookmark88), [205](#bookmark319), [338](#bookmark163), [344](#bookmark9), [411](#bookmark176)], with the exception of [[118](#bookmark74)]. It is worth noting the existence of other techniques to recommend either a single or a ranking of algorithms, such as sequential model-based optimisation [[162](#bookmark333),[406](#bookmark365)], multi-armed bandit [[176](#bookmark363),[179](#bookmark364),[352](#bookmark168)],

1 灬

traditional classiﬁcation [[263](#bookmark76)] and clustering [[148](#bookmark115)], or racing algorithms [[164](#bookmark144)]. Finally, some approaches build ensembles with the best selected algorithms [[205](#bookmark319), [342](#bookmark165)].

Sequential model-based optimisation is the most frequently applied technique in algorithm-independentHPO(57%). Similarlyto meta-learninginthe context ofAS,Bayesian optimisation has been established as the predominant technique for black-box optimisation, specially for the last years [[19](#bookmark24)]. Following the algorithm-independent setting, other tech- niques considered here are evolutionary algorithms [[44](#bookmark367), [60](#bookmark187), [75](#bookmark191), [144](#bookmark371), [210](#bookmark235), [342](#bookmark165), [345](#bookmark166), [351](#bookmark276), [406](#bookmark365)], multi-armed bandit methods [[44](#bookmark367), [123](#bookmark204), [144](#bookmark371), [184](#bookmark334), [229](#bookmark11), [231](#bookmark240), [247](#bookmark245), [339](#bookmark274), [373](#bookmark382)], or gradient- based methods [[134](#bookmark209), [248](#bookmark322), [320](#bookmark268)], among others. In contrast, Bayesian optimisation has been rarely applied for algorithm-dependent HPO [[398](#bookmark356)]. Instead, evolutionary algorithms [[40](#bookmark390), [63](#bookmark136), [401](#bookmark174), [475](#bookmark325)], gradient-based method [[167](#bookmark145), [266](#bookmark250), [310](#bookmark345), [386](#bookmark350)] and swarm intelligence [[76](#bookmark317), [251](#bookmark391), [283](#bookmark158), [289](#bookmark341)] are the preferred. Focusing on NAS, the predominant technique is evolutionary algorithms (25%), as well as transfer learning (22%), gradient-based methods (14%) and reinforcement learning (10%). In the case of gradient-based methods, they have been applied mainly in recent years [[25](#bookmark515)]. Also, in the particular case of transfer learning, it is worth not- ing that it is generally used jointly to other techniques with the aim of reusing weights of previously trained architectures.

Regarding *workﬂow composition*, search and optimisation techniques are the most commonly applied and, more speciﬁcally, Bayesian optimisation (31%) and evolutionary algorithms (18%). They have been used to generate knowledge discovery workﬂows regard- less of whether their structure is ﬁxed beforehand [[71](#bookmark316),[129](#bookmark15),[135](#bookmark114),[212](#bookmark119),[222](#bookmark151),[294](#bookmark94),[297](#bookmark95),[343](#bookmark99),[361](#bookmark101), [375](#bookmark102), [473](#bookmark107)] or not [[57](#bookmark109), [61](#bookmark82), [121](#bookmark86), [122](#bookmark87), [204](#bookmark90), [211](#bookmark149), [273](#bookmark92), [307](#bookmark22), [311](#bookmark96), [331](#bookmark97), [333](#bookmark98), [356](#bookmark100), [484](#bookmark131)]. However, within the category of search and optimisation techniques, AI planning has been only used to compose workﬂows whose structure is not preset [[204](#bookmark90), [207](#bookmark91), [285](#bookmark21), [295](#bookmark122)]. Since generat- ing workﬂows of unbounded size signiﬁcantly increases the search space, some proposals include mechanisms to overcome this issue, such as meta-learning to warm-start the optimi- sation process [[129](#bookmark15)] or multi-objective evolutionary algorithms to prioritise workﬂows with fewer algorithms [[307](#bookmark22)].

Finally, with regard to secondary tasks, we found that ad hoc tasks rely mostly on ML techniques(40%),whilesupportingtasks are frequentlyaddressedwithmeta-learning(21%). More speciﬁcally, meta-learning has been used in AS to assess the predictive performance of meta-features [[58](#bookmark135)] and also to generate an algorithm portfolio [[164](#bookmark144)]. It has been also applied to predict whether it is worth providing additional resources on hyper-parameter optimisation throughboth classiﬁcation[[269](#bookmark155),[270](#bookmark156),[347](#bookmark275),[392](#bookmark171)]and regression[[329](#bookmark271),[347](#bookmark275)]algorithms. Further- more, meta-learning serves to improve the efﬁciency of the ﬁne-tuning process by identifying the most relevant hyper-parameters [[67](#bookmark137)] or deﬁning a multi-ﬁdelity framework [[180](#bookmark442)]. With respect to WC, different techniques have been applied. Workﬂow recommendation has been addressed by means of collaborative ﬁltering and auto-experiment [[207](#bookmark91)] and a learning-to- rank method based on association rules [[428](#bookmark104)], whereas conditional parallel coordinates have provento make the process more interpretable[[422](#bookmark447)]. Inthe case of adhoctasks,the automatic generation of features has been addressed by a large number of techniques like evolutionary algorithms [[119](#bookmark113), [390](#bookmark127)], meta-learning [[203](#bookmark117), [293](#bookmark121)] and reinforcement learning [[206](#bookmark118)], among others. However, these tasks are commonly performed by techniques falling into the category “*Others*”, such as link traversaland SPARQL queries for creating tabular datasets [[268](#bookmark80)], or a custom compiler that transforms the predictive model markup language (PMML) deﬁnition of ML models into source code [[72](#bookmark362)].

**8 Open gaps, challenges and trends**

Our ﬁndings make it clear that AutoML has drawn signiﬁcant research attention during the last years. In response to Q4, we identify a number ofopen gaps andchallenges,and speculate on related future trends.

**8.1 Phases not covered**

Section [6.1](#bookmark65) reveals that AutoML has not covered the automation of the whole knowledge discovery process yet, which affects the practicality of the proposals in a realistic context. In fact, proposals have been unevenly distributed among the phases that require automation. Actually, this review has shown that 93% out of the primary studies are focused on the data mining phase. Moreover, most of them are committed to supervised methods, specially classiﬁcation. This is seemingly motivated by the fact that there is an already labelled dataset and, consequently, it maybe more practicable to measure and validate the performance of the automatically generated classiﬁer. In contrast, unsupervised learning has been barely studied and problems like pattern mining are completely unexplored. Preprocessing is covered by 14% of studies, most of which are related to data cleaning and preprocessing, and data reduction and projection. Special mention should be made of those activities of the pre- and postprocessing phases that are inherently human and require intuition and know-how. For example, supporting the comprehensibility of the application domain or creating the target dataset remains a challenge for AutoML. Similarly, explaining and exploiting the acquired knowledge is still an area which requires further study, despite early efforts to explain the generated models in natural language [[248](#bookmark322)] or generate their source code [[72](#bookmark362)]. We speculate that in the near future there will be progress in involving humans in the automation of the different phases of the knowledge extraction process, e.g. with interactive algorithms.

**8.2 Lack of interoperability**

As noted in Sect.[6.2](#bookmark66), most AutoML proposals are committed to the automation of a sin- gle phase of the knowledge discovery process. This is particularly common among studies addressing the AS and the HPO of the data mining phase. Similarly, WC proposals gener- ally automate only the preprocessing and data mining phases, leaving aside the support for postprocessing and even some preprocessing steps like domain understanding and dataset creation. As noted above, those proposals that have addressed the automation of these phases have done so in a siloed fashion. At this point,the development ofAutoML systems providing comprehensive support for the knowledge discovery process is computationally challenging. However, this holistic view seems necessary if we intend to provide real support for the work of the data scientist, and even support the democratisation of simple data science tasks. Therefore, we foresee that future proposals should consider possible interoperability between different AutoML approaches, thus enabling their reusability and replicability, improving their applicability and greatly facilitating the development and adaptation to the problem of complex workﬂows.

**8.3 Role of the human**

Analysing the AutoML deﬁnitions from the literature, we identify two potential human roles

in AutoML: data scientists [[129](#bookmark15)] and domain experts [[307](#bookmark22)]. AutoML solutions should assist data scientists by automating and optimising those repetitive and time-consuming steps. On the other hand, to help domain experts it is necessary to go a step further, and support the automation of almost all—if not all—of the knowledge discovery process as a way to reduce the entry barrier. Regardless of the role of the participant, we found that most AutoML approaches operate as black-box methods, so that the human must simply rely on the generated models, regardless of how they can be interpreted by (or explained to) the human. Furthermore, with the proper information, the human could provide valuable inputs that could beneﬁt the automation process. Even though, according to the primary studies, interactive approaches have been barely unexplored in the context of AutoML [[161](#bookmark332), [356](#bookmark100)], its usefulness has been demonstrated in the context of evolutionary computation [[28](#bookmark516)] and machine learning [[18](#bookmark517)]. Also, [[13](#bookmark40)] proposed the human-guided machine learning (HGML), as an hybrid framework where an user interacts and guides an AutoML system to explore different problem settings models, features and datasets, according with their knowledge about the data available. In this regard, we speculate that putting the human in the loop in AutoML proposals will be an interesting line of future research. We also speculate that obtaining interpretable models will gain attention in the community, as will the use of both local and global explainable methods.

**8.4 Empirical validation and replicability of proposals**

Section[6.4](#bookmark68) reveals the existence of some shortcomings in the experimental validation of a considerable number of studies. Firstly, we found that many approaches were only compared against some simple ML baseline and/or a variant or previous version of the same proposal. Ideally, AutoML proposals should be compared with other studies in the area addressing the same task, using the same or different technique, and ensuring fair experimental conditions (e.g. experimental environment set-up, tuning of methods, datasets, etc.). Furthermore, we found that only 19% of the primary studies support their results with an appropriate statistical framework. In terms of replicability, it is noteworthy that less than one-third of the studies provide the source code implementing the proposed method. Interestingly, we found some differences depending on tasks. For example, WC approaches tend to be better documented than proposals for HPO. Nevertheless, the top 5 most popular software repositories in the

area are split between WC, HPO and NAS.

**8.5 Future trends**

Section[6.3](#bookmark67)reveals that a large number of primary studies have formulated the automation of theknowledgediscoveryprocess as anoptimisationproblem. Thisis a common alternative for HPO and WC solutions. However, this can be prohibitively resource-intensive, particularly when dealing with large amounts of data (generating a single model can take hours or even days). This is a common lines of future work identiﬁed in the primary studies. Thus, we observed that some authors plan to address this issue by applying a variety of different mechanisms, such as parallelisation [[61](#bookmark82), [107](#bookmark72), [229](#bookmark11), [255](#bookmark377), [300](#bookmark261)],early stopping [[46](#bookmark182), [210](#bookmark235), [242](#bookmark465), [355](#bookmark278)] or the reuse of knowledge from previous optimisation or learning procedures [[297](#bookmark95), [307](#bookmark22), [351](#bookmark276), [385](#bookmark288), [386](#bookmark350)]. In a different ﬁeld, other authors propose extending the scope of their

approaches with respect to one of the AutoML dimensions (see Sect.[4)](#bookmark31). For example, a number of proposals automating the data mining phase plan to extend their work to give support to other data mining tasks apart from classiﬁcation, such as regression [[58](#bookmark135),[129](#bookmark15),[183](#bookmark224)], semi-supervised learning [[129](#bookmark15)] and/or pattern learning [[183](#bookmark224)]. Also, other authors pretend to cover the preprocessing [[342](#bookmark165)] and postprocessing [[345](#bookmark166)] phases. On the other hand, some authors working on AS [[95](#bookmark139), [397](#bookmark173)] and WC [[57](#bookmark109), [69](#bookmark84)] acknowledge the importance of tuning the hyper-parameters. Finally, although to a lesser extent, we can already ﬁnd the ﬁrst works focused on studying how to improve the experience of the human operating the AutoML system either through the design of graphical user interfaces [[44](#bookmark367), [317](#bookmark124), [428](#bookmark104)] or through the application of interactive techniques [[122](#bookmark87), [287](#bookmark77), [416](#bookmark440)].

**9 Concluding remarks**

The application of the most appropriate and valuable machine learning methods in a given situation often requires a considerable amount of technical knowledge, which prevents these techniques from being accessible and practicable for domain experts. At the same time, this learning process involves a number of repetitive and time-consuming tasks like hyper- parameter optimisation. AutoML has emerged to automate machine learning, thus reducing times and making it more practicable for both data scientists and domain experts. This paper presents the results of a systematic literature review of the ﬁeld of AutoML. With this aim, we have followed a precise methodology, according to which we have selected and examined a total of 447 conference and journal papers as primary studies. It is worth noting that, because of the large number of studies, we had to impose some limitations on the scope of this manuscript. Speciﬁcally, we opted to limit the snow-balling procedure to a single level, which meant that we did not apply it to any papers obtained through this means. Additionally, we excluded papers that had not been published in conferences rated A\* or A according to the CORE ranking system, or those published in journals not indexed in JCR. Based on a preliminary analysis of these studies, we have proposed a taxonomy for AutoML that allows categorising any AutoML approach into three different perspectives: the phase of the knowledge discovery process being automated, the task being performed, and the techniques applied to address the automation procedure.

All primary studies have been reviewed according to this three-dimensional taxonomy, in addition to considering several other aspects such as the experimental framework or the availability of additional material. As a result, we have observed that research in the area has mostly focused on the automation of the data mining phase and, to a lesser extent, also on preprocessing, particularly on data reduction and projection. In contrast, the postprocessing phase and other inherently human activities have been barely taken into consideration. It is worth noting that the number of publications in 2021 has nearly quintupled compared to 2014. Across-dimensional analysishas allowedus to identify current gaps andtrendsfromthe AutoML literature, such as the lack of well-established tasks to automate the postprocessing or the shift from evolutionary algorithms to gradient-based methods as the most widely used technique for NAS. In terms of practicability, it is worth noting that 25% of the primary studies provide instructions on how to install (if required) and execute their proposals, less than4%provide some graphical user interface, andonly32%make their source codepublicly available. Also, our ﬁndings reveal that AutoML proposals tend to address such automation process as a black-box problem, limiting the interpretability of the resulting models or the explanation provided about the process to reach the generated solution.

Finally, from the observed evolution of the area, we can assume that AutoML is still an emerging area facing promising challenges, with a clear dominance of neural architecture search proposals in the last four years. Hopefully the proposed taxonomy herein will con- tribute to the classiﬁcation of future proposals, and we envision that the results of this work will serve as a reference to researchers and practitioners interested in AutoML. For future research, we aim to maintain the taxonomy presented in this review up-to-date as new pro- posals emerge. In addition, we propose developing a repository that enables the querying of AutoML techniques based on their various dimensions of the taxonomy. Similarly, an examination of the most noteworthy AutoML tools, classiﬁed according to the taxonomy, would provide valuable insights.

**Supplementary information**

The complete systematic review protocolisprovidedseparately as additionalmaterial,includ- ing detailed information about the search process, the selection of candidate and primary studies and the data extraction process. In addition, raw search outcomes and the results of thedata extractionprocess are available. Apartfromthematerialinthisrepository([https://doi.](https://doi.org/10.5281/zenodo.7861452)

[org/10.5281/zenodo.7861452](https://doi.org/10.5281/zenodo.7861452)), no further Electronic Supplementary Material is provided.

**Acknowledgements** Grant PID2020-115832GB-I00 funded by MICIN/AEI/10.13039/501100011033; Span- ish Ministry of Education under the FPU program, Grant: FPU17/00799.

**Author Contributions** Conceptualisation: JRR, SV; Methodology: JRR; Literature search: RB; Data analy- sis: All authors; Writing - original draft preparation: RB; Writing - review and editing: JRR, SV; Funding acquisition: SV, JRR; Supervision: JRR, SV. All authors read and approved the ﬁnal manuscript.

**Funding** Funding for open access publishing: Universidad de Córdoba/CBUA **Declarations**

**Conﬂicts of interest** The authors have no relevant ﬁnancial or non-ﬁnancial interests to disclose.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing,adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit [http://creativecommons.org/licenses/by/4.0/ .](http://creativecommons.org/licenses/by/4.0/)

**References**

1. AliS, Smith K(2006)Onlearningalgorithmselectionfor classiﬁcation.Appl SoftComput6(2):119–138.

<https://doi.org/10.1016/j.asoc.2004.12.002>

2. Azevedo A, Santos M (2008) KDD, SEMMA and CRISP-DM: a parallel overview. IADS-DM

3. Baeck T, Schwefel H (1996) Evolutionary computation: an overview. In: Proceedings of the IEEE conference on evolutionary computation, pp 20–29. <https://doi.org/10.1109/ICEC.1996.542329>

4. Bergstra J, Bengio Y (2012) Random search for hyper-parameter optimization. J Mach Learn Res 13:281–305

5. BernsteinA, ProvostF, HillS(2005)Towardintelligent assistancefor a dataminingprocess:an ontology- based approach for cost-sensitive classiﬁcation. IEEE Trans Knowl Data Eng 17(4):503–518. [https://](https://doi.org/10.1109/TKDE.2005.67) [doi.org/10.1109/TKDE.2005.67](https://doi.org/10.1109/TKDE.2005.67)

1 灬

6. Bilalli B, Abelló A, Aluja-Banet T et al (2016) Automated data pre-processing via meta-learning. In: International conference on modelling and data engineering, pp 194–208. [https://doi.org/10.1007/978-](https://doi.org/10.1007/978-3-319-45547-1_16) [3-319-45547-1\_16](https://doi.org/10.1007/978-3-319-45547-1_16)

7. Chapman P, Clinton J, Kerber R etal (2000) CRISP-DM 1.0: step-by-step data mining guide. SPSS inc, p 16

8. Chen L, Collins M, Zhu Y et al (2018) Searching for efﬁcient multi-scale architectures for dense image prediction. Adv. Neural Inf. Process. Syst. 31:8699–8710

9. Coleman S, Göb R, Manco G et al (2016) How can SMEs beneﬁt from big data? Challenges and a path forward. Qual Reliab Eng Int 32(6):2151–2164. <https://doi.org/10.1002/qre.2008>

10. Elsken T, Metzen J,Hutter F (2019) Neural architecture search: a survey. J Mach Learn Res 20(55):1–21

11. EscalanteH, Montes M, SucarL (2009) Particle swarm model selection. J Mach Learn Res 10:405–440

12. Fayyad U, Piatetsky-Shapiro G, Smyth P (1996) Advances in knowledge discovery and data mining. chap from data mining to knowledge discovery: an overview, pp 1–34. [https://doi.org/10.1609/aimag.](https://doi.org/10.1609/aimag.v17i3.1230) [v17i3.1230](https://doi.org/10.1609/aimag.v17i3.1230)

13. Gil Y, Honaker J, Gupta S et al (2019) Towards human-guided machine learning. In: Proceedings of the 24th international conference on intelligent user interfaces, pp 614–624. [https://doi.org/10.1145/](https://doi.org/10.1145/3301275.3302324) [3301275.3302324](https://doi.org/10.1145/3301275.3302324)

14. Guyon I, Bennett K, Cawley G et al (2015) Design of the 2015 chalearn automl challenge. In: 2015 International joint conference on neural networks (IJCNN), pp 1–8. [https://doi.org/10.1109/IJCNN.](https://doi.org/10.1109/IJCNN.2015.7280767) [2015.7280767](https://doi.org/10.1109/IJCNN.2015.7280767)

15. He K, Zhang X, Ren S et al (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778. [https://doi.org/10.1109/](https://doi.org/10.1109/CVPR.2016.90) [CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90)

16. HeX, Zhao K, Chu X (2021) AutoML: a survey of the state-of-the-art. Knowl Syst 212(106):622.[https://](https://doi.org/10.1016/j.knosys.2020.106622) [doi.org/10.1016/j.knosys.2020.106622](https://doi.org/10.1016/j.knosys.2020.106622)

17. Hinton G,DengL, Yu D et al (2012) Deep neural networks for acoustic modeling in speech recognition. IEEE Signal Process Mag. <https://doi.org/10.1109/MSP.2012.2205597>

18. Holzinger A (2016) Interactive machine learning for health informatics: when do we need the human- in-the-loop? Brain Inf 3(2):119–131. <https://doi.org/10.1007/s40708-016-0042-6>

19. Hutter F, Lücke J, Schmidt-ThiemeL (2015) Beyond manual tuning of hyperparameters. KI - Künstliche Intelligenz 29(4):329–337. <https://doi.org/10.1007/s13218-015-0381-0>

20. Hutter F, KotthoffL, Vanschoren J (2019) Automatic machine learning: methods, systems, challenges

21. Jordan M, Mitchell T (2015) Machine learning: trends, perspectives, and prospects. Science 349(6245):255–260. <https://doi.org/10.1126/science.aaa8415>

22. Kitchenham B, Charters S (2007) Guidelines for performing systematic literature reviews in software engineering. In: EBSE 2007-001. Keele University and Durham University Joint Report [https://www.](https://www.elsevier.com/__data/promis_misc/525444systematicreviewsguide.pdf) [elsevier.com/\_\_data/promis\_misc/525444systematicreviewsguide.pdf](https://www.elsevier.com/__data/promis_misc/525444systematicreviewsguide.pdf)

23. KotthoffL (2016) Algorithm selection for combinatorial search problems: a survey. In: Data mining and constraint programming, pp 149–190. <https://doi.org/10.1609/aimag.v35i3.2460>

24. Lee I, ShinY (2020) Machine learning for enterprises: applications, algorithm selection, and challenges. Bus Horiz 63(2):157–170. <https://doi.org/10.1016/j.bushor.2019.10.005>

25. Liu H, Simonyan K, Yang Y (2019) DARTS: differentiable architecture search. In: International conference on learning representation. ICLR

26. Luo G (2016) A review of automatic selection methods for machine learning algorithms and hyper- parameter values. Netw Model Anal Health Inf Bioinform 5(1):18. [https://doi.org/10.1007/s13721-](https://doi.org/10.1007/s13721-016-0125-6) [016-0125-6](https://doi.org/10.1007/s13721-016-0125-6)

27. PatilT, Davenport T(2012)Data scientist:the sexiestjobofthe21st century. Harvard Bus Rev90(10):70– 76

28. Ramirez A, Romero J, Ventura S(2018)Interactive multi-objective evolutionaryoptimization of software architectures. Inf Sci 463:92–109. <https://doi.org/10.1016/j.ins.2018.06.034>

29. Rice J(1976)The algorithmselectionproblem. AdvComput15:65–118.[https://doi.org/10.1016/S0065-](https://doi.org/10.1016/S0065-2458(08)60520-3) [2458(08)60520-3](https://doi.org/10.1016/S0065-2458(08)60520-3)

30. deSá A, Pinto W, Oliveira Let al (2017) Recipe: a grammar-based framework for automatically evolving classiﬁcation pipelines. Genet Program. [https://doi.org/10.1007/978-3-319-55696-3\_ 16](https://doi.org/10.1007/978-3-319-55696-3_16)

31. Serban F, Vanschoren J, Kietz J et al (2013) A survey of intelligent assistants for data analysis. ACM Comput Surv 45(3):31:1-31:35. <https://doi.org/10.1145/2480741.2480748>

32. Shahriari B, Swersky K, Wang Z et al (2016) Taking the human out of the loop: a review of bayesian optimization. Proc IEEE 104(1):148–175. <https://doi.org/10.1109/JPROC.2015.2494218>

33. Snoek J, Larochelle H, Adams R (2012) Practical bayesian optimization of machine learning algorithms. Adv Neural Inf Process Syst 25:2951–2959

34. Thornton C, Hutter F, Hoos H et al (2013) Auto-WEKA: combined selection and hyperparameter opti- mization ofclassiﬁcationalgorithms. In:Proceedingsofthe19thACM SIGKDDinternational conference on knowledge discovery and data mining, KDD ’13, pp 847–855. [https://doi.org/10.1145/2487575.](https://doi.org/10.1145/2487575.2487629) [2487629](https://doi.org/10.1145/2487575.2487629)

35. Tripathy M, Panda A (2017) A study of algorithm selection in data mining using meta-learning. J Eng Sci Technol Rev. <https://doi.org/10.25103/jestr.102.06>

36. Wolpert D, Macready W (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82. <https://doi.org/10.1109/4235.585893>

37. Zakova M, Kremen P, Zelezny F et al (2011) Automating knowledge discovery workﬂow composition through ontology-based planning. IEEE Trans Autom Sci Eng 8(2):253–264. [https://doi.org/10.1109/](https://doi.org/10.1109/TASE.2010.2070838) [TASE.2010.2070838](https://doi.org/10.1109/TASE.2010.2070838)

**Primary Studies (Sources Reviewed in the SLR)**

38. Abdulrahman S, Brazdil P, van Rijn J et al (2018) Speeding up algorithm selection using average ranking and active testing by introducing runtime. Mach Learn 107:79–108. [https://doi.org/10.1007/](https://doi.org/10.1007/s10994-017-5687-8) [s10994-017-5687-8](https://doi.org/10.1007/s10994-017-5687-8)

39. Achararit P, Hanif MA, PutraRVW et al (2020) APNAS: accuracy-and-performance-aware neural archi- tecture search for neural hardware accelerators. IEEE Access 8:16,5319-16,5334. [https://doi.org/10.](https://doi.org/10.1109/ACCESS.2020.3022327) [1109/ACCESS.2020.3022327](https://doi.org/10.1109/ACCESS.2020.3022327)

40. Agrawal S, Sarkar S, Alazab M et al (2021) Genetic CFL: hyperparameter optimization in clustered federated learning. Comput Intell Neurosci. <https://doi.org/10.1155/2021/7156420>

41. Ahmadizar F, SoltanianK,Akhlaghiantab F et al (2015) Artiﬁcial neural network development by means of a novel combination of grammatical evolution and genetic algorithm. Eng Appl Artif Intell 39:1–13.

<https://doi.org/10.1016/j.engappai.2014.11.003>

42. Ahmed M, VaswaniS, Schmidt M (2019) Combining Bayesian optimization and Lipschitz optimization. Mach Learn. <https://doi.org/10.1007/s10994-019-05833-y>

43. Ahn P, Hong H, Kim J (2021) Differentiable architecture search based on coordinate descent. IEEE Access 9:48544–48554. <https://doi.org/10.1109/ACCESS.2021.3068766>

44. Akiba T, Sano S, Yanase T et al (2019) Optuna: a next-generation hyperparameter optimization framework. In: ACM SIGKDD international conference on knowledge discovery and data mining, pp 2623–2631. <https://doi.org/10.1145/3292500.3330701>

45. Akl A, El-Henawy I, Salah A et al (2019) Optimizing deep neural networks hyperparameter positions and values. J Intell Fuzzy Syst. <https://doi.org/10.3233/JIFS-190033>

46. AlbelwiS, Mahmood A(2017)A framework fordesigning the architectures of deep convolutionalneural networks. Entropy. <https://doi.org/10.3390/e19060242>

47. Ali R, Lee S, Chung T (2017) Accurate multi-criteria decision making methodology for recommending machine learning algorithm. Expert Syst Appl 71:257–278.<https://doi.org/10.1016/j.eswa.2016.11.034>

48. Ariafar S, MarietZ, Brooks D et al (2021) Faster and more reliable tuning of neural networks: bayesian optimization with importance sampling. In: International Conference on Artiﬁcial Intelligence and Statistics, pp 3961–3969

49. Assunção F, Lourenço N, Machado P et al (2019) DENSER: deep evolutionary network structured representation. Genet Program Evol Mach 20:5–35. <https://doi.org/10.1007/s10710-018-9339-y>

50. Bafna A, Wiens J (2016) Automated feature learning: mining unstructured data for useful abstractions. In: IEEE international conference on data mining, pp 703–708.<https://doi.org/10.1109/ICDM.2015.115>

51. Baldominos A, Saez Y, Isasi P (2018) Evolutionary design of convolutional neural networks for human activity recognition in sensor-rich environments. Sensors (Switzerland).[https://doi.org/10.3390/](https://doi.org/10.3390/s18041288) [s18041288](https://doi.org/10.3390/s18041288)

52. Basha S, Vinakota S, Pulabaigari V et al (2021) AutoTune: automatically tuning convolutional neural networks for improved transfer learning. Neural Netw 133:112–122. [https://doi.org/10.1016/j.neunet.](https://doi.org/10.1016/j.neunet.2020.10.009) [2020.10.009](https://doi.org/10.1016/j.neunet.2020.10.009)

53. Bello I, Zoph B, Vasudevan V et al (2017) Neural optimizer search with Reinforcement learning. In: International conference on machine learning, pp 712–721

54. Bender G, Kindermans PJ, Zoph B et al (2018) Understanding and simplifying one-shot architecture search. In: International conference on machine learning, pp 883–893

55. Bender G, Liu H, Chen B et al (2020) Can weight sharing outperform random architecture search? An investigation with TuNAS. In: IEEE/CVF conference on computer vision and pattern Recognition, pp 14311–14320. <https://doi.org/10.1109/CVPR42600.2020.01433>

1 灬

56. Berk J, Nguyen V, Gupta S et al (2019) Exploration enhanced expected improvement for bayesian optimization. In: European conference on machine learning and principles and practice of knowledge discovery in databases, pp 621–637. <https://doi.org/10.1007/978-3-030-10928-8_37>

57. Bi Y, Xue B, Zhang M (2019) An automated ensemble learning framework using genetic programming for image classiﬁcation. In: Genetic programming and Evolvable Computing Conference, pp 365–373.

<https://doi.org/10.1145/3321707.3321750>

58. Bilalli B, Abelló A, Aluja-Banet T (2017) On the predictive power of meta-features in OpenML. Int J Appl Math Comput Sci 27:697–712. <https://doi.org/10.1515/amcs-2017-0048>

59. BilalliB, Abelló A,Aluja-BanetTetal (2018) PRESISTANT: data pre-processing assistant. In: Interna- tional conference on advance information system engineering, pp 57–65. [https://doi.org/10.1007/978-](https://doi.org/10.1007/978-3-319-92901-9_6) [3-319-92901-9\_6](https://doi.org/10.1007/978-3-319-92901-9_6)

60. Binder M, Moosbauer J, Thomas J, et al (2020) Multi-objective hyperparameter tuning and feature selectionusingﬁlterensembles. In:Genetic and evolutionarycomputing conference,pp471–479.[https://](https://doi.org/10.1145/3377930.3389815) [doi.org/10.1145/3377930.3389815](https://doi.org/10.1145/3377930.3389815)

61. Binder M, Pﬁsterer F, Lang M et al (2021) mlr3pipelines-ﬂexible machine learning pipelines in r. J Mach Learn Res 22:1–7

62. Boselli R, Cesarini M, Mercorio F et al (2017) An AI planning system for data cleaning. In: European conference on machine learning and knowledge discovery in databases, pp 349–353. [https://doi.org/10.](https://doi.org/10.1007/978-3-319-71273-4_29) [1007/978-3-319-71273-4\_29](https://doi.org/10.1007/978-3-319-71273-4_29)

63. Bouraoui A, JamoussiS, BenAyedY (2018) A multi-objective genetic algorithm for simultaneous model and featureselection for support vector machines. Artif Intell Rev 50:261–281.[https://doi.org/10.1007/](https://doi.org/10.1007/s10462-017-9543-9) [s10462-017-9543-9](https://doi.org/10.1007/s10462-017-9543-9)

64. BulatA,MartinezB,Tzimiropoulos G(2020)BATS:binary architecturesearch. In:Europeanconference on computer vision, pp 309–325. [https://doi.org/10.1007/978-3-030-58592-1\_ 19](https://doi.org/10.1007/978-3-030-58592-1_19)

65. Cai H, Yang J, Zhang W et al (2018) Path-level network transformation for efﬁcient architecture search. In: International conference on machine learning, pp 1069–1080

66. Cai H, Lin J, Lin Yet al (2020) AutoML for architecting efﬁcient and specialized neural networks. IEEE Micro 40:75–82. <https://doi.org/10.1109/MM.2019.2953153>

67. Cai Z, Long Y, Shao L (2019) Classiﬁcation complexity assessment for hyper-parameter optimization. Pattern Recognit Lett 125:396–403. <https://doi.org/10.1016/j.patrec.2019.05.021>

68. Cambronero J, Cito J, Rinard M (2020) AMS: Generating AutoML search spaces from weak speciﬁca- tions. In: ACM joint European software engineering conference and symposium on the foundations of software engineering, pp 763–774. <https://doi.org/10.1145/3368089.3409700>

69. Cambronero JP, Rinard MC (2019) AL: autogenerating supervised learning programs. ACM Program Lang. <https://doi.org/10.1145/3360601>

70. Camero A, Wang H, Alba E et al (2021) Bayesian neural architecture search using a training-free performance metric. Appl Soft Comput. <https://doi.org/10.1016/j.asoc.2021.107356>

71. Candelieri A, Archetti F (2019) Global optimization in machine learning: the design of a predictive analytics application. Soft Comput 23:2969–2977. <https://doi.org/10.1007/s00500-018-3597-8>

72. Castro-Lopez O, Vega-Lopez I (2019) Multi-target Compiler for the Deployment of Machine Learning

Models. In: IEEE/ACM international symposium on code generation and optimization, pp 280–281.

<https://doi.org/10.1109/CGO.2019.8661199>

73. Chandrashekaran A, Lane I (2017) Speeding up hyper-parameter optimization by extrapolation of learn- ing curves using previous builds. In: European conference on machine learning and knowledge discovery in databases, pp 477–492. <https://doi.org/10.1007/978-3-319-71249-9_29>

74. CharteF, Rivera A, Martínez F et al (2020) EvoAAA: an evolutionary methodology for automated neural autoencoder architecture search. Integr ComputEng 27:211–231. <https://doi.org/10.3233/ICA-200619>

75. ChatzimparmpasA, MartinsR, Kucher Ketal(2021)VisEvol:visualanalyticstosupporthyperparameter search through evolutionary optimization. Comput Graph Forum 40(3):201–214. [https://doi.org/10.](https://doi.org/10.1111/cgf.14300) [1111/cgf.14300](https://doi.org/10.1111/cgf.14300)

76. Che J, Yang Y, Li L et al (2017) A modiﬁed support vector regression: integrated selection of training subset and model. Appl Soft Comput J 53:308–322. <https://doi.org/10.1016/j.asoc.2016.12.053>

77. Chen B, Wu H, Mo W et al (2018) Autostacker: a compositional evolutionary learning system. In: Genetic programming and evolvable computing conference, pp 402–409. [https://doi.org/10.1145/](https://doi.org/10.1145/3205455.3205586) [3205455.3205586](https://doi.org/10.1145/3205455.3205586)

78. Chen D, Shen H, Shen Y (2021) DcaNAS: efﬁcient convolutional network design for desktop CPU platforms. Appl Intell 51(7):4353–4366. <https://doi.org/10.1007/s10489-020-02133-0>

79. Chen H, Zhang B, Xue S et al (2020a) Anti-bandit neural architecture search for model defense. In: European conference on computer vision, pp 70–85. <https://doi.org/10.1007/978-3-030-58601-0_5>

80. ChenH,ZhuoL,ZhangBetal(2021)Binarizedneuralarchitecturesearchfor efﬁcientobjectrecognition. Int J Comput Vis 129:501–516. <https://doi.org/10.1007/s11263-020-01379-y>

81. Chen J, Gao J, Chen Y et al (2021c) GraphPAS: parallel architecture search for graph neural networks. In: International ACM SIGIR conference on research and development in information retrieval, pp 2182–2186. <https://doi.org/10.1145/3404835.3463007>

82. Chen M,Fu J, Ling H (2021d) One-shot neural ensemble architecture search by diversity-guided search space shrinking. In:IEEE/CVF conference on computervisionandpatternrecognition,pp16525–16534.

<https://doi.org/10.1109/CVPR46437.2021.01626>

83. Chen S, Wu J, Liu X (2021) EMORL: effective multi-objective reinforcement learning method for hyperparameter optimization. Eng Appl Artif Intell. <https://doi.org/10.1016/j.engappai.2021.104315>

84. Chen X, Duan Y, Chen Z et al (2020b) CATCH: context-based meta reinforcement learning for trans- ferrable architecture search. In: European conference on computer vision, pp 185–202. [https://doi.org/](https://doi.org/10.1007/978-3-030-58529-7_12) [10.1007/978-3-030-58529-7\_12](https://doi.org/10.1007/978-3-030-58529-7_12)

85. Chen X, Xie L, Wu J et al (2021) Progressive DARTS: bridging the optimization gap for NAS in the wild. Int J Comput Vis 129:638–655. <https://doi.org/10.1007/s11263-020-01396-x>

86. Chen Y, Meng G, Zhang Q, et al (2019a) RENAS: reinforced evolutionary neural architecture search. In: IEEE/CVF computer vision and pattern recognition conference, pp 4782–4791. [https://doi.org/10.](https://doi.org/10.1109/CVPR.2019.00492) [1109/CVPR.2019.00492](https://doi.org/10.1109/CVPR.2019.00492)

87. Chen Y, Zhu K, Zhu L et al (2019) Automatic design of convolutional neural network for hyperspectral imageclassiﬁcation. IEEE Trans Geosci Remote Sens 57:7048–7066. [https://doi.org/10.1109/TGRS.](https://doi.org/10.1109/TGRS.2019.2910603) [2019.2910603](https://doi.org/10.1109/TGRS.2019.2910603)

88. Chen Y, Gao R, Liu F et al (2021) ModuleNet: knowledge-inherited neural architecture search. IEEE Trans Cybern. <https://doi.org/10.1109/TCYB.2021.3078573>

89. Chen Y, Guo Y, Chen Q et al (2021h) Contrastive neural architecture search with neural architecture comparators. In: IEEE/CVF computer vision and pattern recognition conference, pp 9497–9506.[https://](https://doi.org/10.1109/CVPR46437.2021.00938) [doi.org/10.1109/CVPR46437.2021.00938](https://doi.org/10.1109/CVPR46437.2021.00938)

90. Chen Z, Li B (2020) Efﬁcient evolution for neural architecture search. In: International joint conference on neural network. <https://doi.org/10.1109/IJCNN48605.2020.9207545>

91. Cheng ZQ, Wu X, Huang S et al (2018) Learning to transfer: generalizable attribute learning with multitask neural model search. In: ACM multimedia conference, pp 90–98. [https://doi.org/10.1145/](https://doi.org/10.1145/3240508.3240518) [3240508.3240518](https://doi.org/10.1145/3240508.3240518)

92. Chiu C, Zhan J (2019) An evolutionary approach to compact DAG neural network optimization. IEEE Access 7:178331–178341. <https://doi.org/10.1109/ACCESS.2019.2954795>

93. Chouikhi N, Ammar B, Hussain A et al (2019) Bi-level multi-objective evolution of a multi-layered echo-state network autoencoder for data representations. Neurocomputing 341:195–211. [https://doi.](https://doi.org/10.1016/j.neucom.2019.03.012) [org/10.1016/j.neucom.2019.03.012](https://doi.org/10.1016/j.neucom.2019.03.012)

94. Chu X, Zhou T, Zhang B et al (2020) Fair DARTS: eliminating unfair advantages in differentiable architecture search. In: European conference on computer vision, pp 465–480. [https://doi.org/10.1007/](https://doi.org/10.1007/978-3-030-58555-6_28) [978-3-030-58555-6\_28](https://doi.org/10.1007/978-3-030-58555-6_28)

95. Cohen-ShapiraN, RokachL,ShapiraBetal(2019)AutoGRD:modelrecommendationthroughgraphical dataset representation. In: ACM international conference on information and knowledge management, pp 821–830. <https://doi.org/10.1145/3357384.3357896>

96. Cordeiro J, Raimundo A, Postolache O et al (2021) Neural architecture search for 1d CNNS. Different approaches tests and measurements. Sensors 8:9. <https://doi.org/10.3390/s21237990>

97. Corrales D, Ledezma A, Corrales J (2020) A case-based reasoning system for recommendation of data cleaning algorithms in classiﬁcation and regression tasks. Appl Soft Comput J. [https://doi.org/10.1016/](https://doi.org/10.1016/j.asoc.2020.106180) [j.asoc.2020.106180](https://doi.org/10.1016/j.asoc.2020.106180)

98. Cortes C, Gonzalvo X, Kuznetsov V et al (2017) AdaNet: adaptive structural learning of artiﬁcial neural networks. In: International conference on machine learning, pp 1452–1466

99. Da Silva R, Canuto A, Barreto C et al (2021) Automatic recommendation method for classiﬁer ensemble structure using meta-learning. IEEE Access 9:106254–106268.[https://doi.org/10.1109/ACCESS.2021.](https://doi.org/10.1109/ACCESS.2021.3099689) [3099689](https://doi.org/10.1109/ACCESS.2021.3099689)

100. Dai X, Zhang P, Wu B et al (2019a) ChamNet: towards efﬁcient network design through platform- aware model adaptation. In: IEEE/CVF conference on computer vision and pattern recognition, pp 11390–11399. <https://doi.org/10.1109/CVPR.2019.01166>

101. Dai X, Chen D, Liu M et al (2020) DA-NAS: data adapted pruning for efﬁcient neural architecture search. In: European conference on computer vision, pp 584–600. [https://doi.org/10.1007/978-3-030-](https://doi.org/10.1007/978-3-030-58583-9_35) [58583-9\_35](https://doi.org/10.1007/978-3-030-58583-9_35)

1 灬

102. DaiX,WanA,Zhang Petal(2021)FbNetv3:joint architecture-recipe search using predictor pretraining. In: IEEE/CVF IEEE conference on computer vision and pattern recognition, pp 16271–16280. [https://](https://doi.org/10.1109/CVPR46437.2021.01601) [doi.org/10.1109/CVPR46437.2021.01601](https://doi.org/10.1109/CVPR46437.2021.01601)

103. Dai Z, Yu H, Low BKH et al (2019b) Bayesian optimization meets bayesian optimal stopping. In: International conference on machine learning, pp 1496–1506

104. Dale M (2018) Neuroevolution of hierarchical reservoir computers. Genet Evol Comput Conf. [https://](https://doi.org/10.1145/3205455.3205520) [doi.org/10.1145/3205455.3205520](https://doi.org/10.1145/3205455.3205520)

105. De CamposL,De OliveiraR, RoisenbergM(2016)Optimization ofneuralnetworksthroughgrammatical evolution and a genetic algorithm. Expert Syst Appl 56:368–384. [https://doi.org/10.1016/j.eswa.2016.](https://doi.org/10.1016/j.eswa.2016.03.012) [03.012](https://doi.org/10.1016/j.eswa.2016.03.012)

106. De Raedt L, Blockeel H, Kolb S et al (2018) Elements of an automatic data scientist. In: International symposium on intelligent data analysis, pp 3–14. [https://doi.org/10.1007/978-3-030-01768-2\_ 1](https://doi.org/10.1007/978-3-030-01768-2_1)

107. Demiralp C, Haas P, Parthasarathy S et al (2017) Foresight: recommending visual insights. In: International conference on very large databases, pp 1937–1940. [https://doi.org/10.14778/3137765.](https://doi.org/10.14778/3137765.3137813) [3137813](https://doi.org/10.14778/3137765.3137813)

108. Desautels T, Krause A, Burdick J (2014) Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization. J Mach Learn Res 15:3873–3923

109. Dias L, Miranda P, Nascimento A et al (2021) ImageDataset2Vec: an image dataset embedding for algorithm selection. Expert Syst Appl. <https://doi.org/10.1016/j.eswa.2021.115053>

110. Diaz G, Fokoue-Nkoutche A, Nannicini G et al (2017) An effective algorithm for hyperparameter optimization of neural networks. IBM J Res Dev. <https://doi.org/10.1147/JRD.2017.2709578>

111. Dibia V, Demiralp C (2019) Data2Vis: automatic generation of data visualizations using sequence- to-sequence recurrent neural networks. IEEE Comput Graph Appl 39:33–46. [https://doi.org/10.1109/](https://doi.org/10.1109/MCG.2019.2924636) [MCG.2019.2924636](https://doi.org/10.1109/MCG.2019.2924636)

112. Ding M, Lian X, Yang L et al (2021a) HR-NAS: searching efﬁcient high-resolution neural architectures with lightweight transformers. In: IEEE/CVF conference on computer vision and pattern recognition, pp 2981–2991. <https://doi.org/10.1109/CVPR46437.2021.00300>

113. Ding Y, Yao Q, Zhao H etal (2021b) DiffMG: differentiable meta graph search for heterogeneous graph neural networks. In: ACM SIGKDD conference on knowledge discovery and data mining, pp 279–288.

<https://doi.org/10.1145/3447548.3467447>

114. Ding Z, Chen Y, Li N et al (2021) BNAS: efﬁcient neural architecture search using broad scalable architecture. IEEE Trans Neural Netw Learn Syst. <https://doi.org/10.1109/TNNLS.2021.3067028>

115. Domhan T, Springenberg J, Hutter F (2015) Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In: International joint conference on artiﬁcial intelligence, pp 3460–3468

116. Dong JD, Cheng AC, Juan DC et al (2018) DPP-net: device-aware progressive search for pareto-optimal neural architectures. In: European conference on computer vision, pp 540–555.[https://doi.org/10.1007/](https://doi.org/10.1007/978-3-030-01252-6_32) [978-3-030-01252-6\_32](https://doi.org/10.1007/978-3-030-01252-6_32)

117. Dong X, Yang Y (2019) Searching for a robust neural architecture in four GPU hours. In: IEEE/CVF conference on computer vision and pattern recognition, pp 1761–1770. [https://doi.org/10.1109/CVPR.](https://doi.org/10.1109/CVPR.2019.00186) [2019.00186](https://doi.org/10.1109/CVPR.2019.00186)

118. Ehsan H, Sharaf M, Chrysanthis P (2016) MuVE: efﬁcient multi-objective view recommendation for visual data exploration. In: IEEE international conference on data engineering, pp 731–742. [https://doi.](https://doi.org/10.1109/ICDE.2016.7498285) [org/10.1109/ICDE.2016.7498285](https://doi.org/10.1109/ICDE.2016.7498285)

119. Elola A,Del SerJ, Bilbao Metal(2017)Hybridizingcartesiangeneticprogrammingandharmony search for adaptive feature construction in supervised learning problems. Appl Soft Comput J 52:760–770.

<https://doi.org/10.1016/j.asoc.2016.09.049>

120. Elsken T, Stafﬂer B, Metzen J et al (2020) Meta-learning of neural architectures for few-shot learning. In: IEEE/CVF conference on computer vision and pattern recognition, pp 12362–12372.[https://doi.org/](https://doi.org/10.1109/CVPR42600.2020.01238) [10.1109/CVPR42600.2020.01238](https://doi.org/10.1109/CVPR42600.2020.01238)

121. Estevez-Velarde S, Gutiérrez Y, Montoyo A et al (2019) AutoML strategy based on grammatical evo- lution: a case study about knowledge discovery from text. In: Annual meeting of the association for computational linguistics, pp 4356–4365. <https://doi.org/10.18653/v1/P19-1428>

122. Estevez-VelardeS, Gutiérrez Y, Almeida-Cruz Y et al (2021) General-purpose hierarchical optimisation of machine learning pipelines with grammatical evolution. Inf Sci 543:58–71. [https://doi.org/10.1016/](https://doi.org/10.1016/j.ins.2020.07.035) [j.ins.2020.07.035](https://doi.org/10.1016/j.ins.2020.07.035)

123. Falkner S, Klein A, Hutter F (2018) BOHB: robust and efﬁcient hyperparameter optimization at scale. In: International conference on machine learning, pp 2323–2341

124. Fan Y, Zhou G, Shen J et al (2021) Toward gradient bandit-based selection of candidate architectures in AutoGAN. Soft Comput 25(6):4367–4378. <https://doi.org/10.1007/s00500-020-05446-x>

125. Fatyanosa T, Aritsugi M (2021) An automatic convolutional neural network optimization using a diversity-guided genetic algorithm. IEEE Access 9:91410–91426. [https://doi.org/10.1109/ACCESS.](https://doi.org/10.1109/ACCESS.2021.3091729) [2021.3091729](https://doi.org/10.1109/ACCESS.2021.3091729)

126. Fernandes Junior F, Yen G (2019) Particle swarm optimization of deep neural networks architectures for image classiﬁcation. Swarm Evol Comput 49:62–74. <https://doi.org/10.1016/j.swevo.2019.05.010>

127. Fernando C, Banarse D, Reynolds M et al (2016) Convolution by evolution: differentiable pattern producing networks. In: Genetic and evolvable computing conference, pp 109–116. [https://doi.org/10.](https://doi.org/10.1145/2908812.2908890) [1145/2908812.2908890](https://doi.org/10.1145/2908812.2908890)

128. Ferrari D, De Castro L (2015) Clustering algorithm selection by meta-learning systems: a new distance- based problem characterization and ranking combination methods. Inf Sci 301:181–194.[https://doi.org/](https://doi.org/10.1016/j.ins.2014.12.044) [10.1016/j.ins.2014.12.044](https://doi.org/10.1016/j.ins.2014.12.044)

129. Feurer M, Klein A, Eggensperger K et al (2015) Efﬁcient and robust automated machine learning. In: Annual conference on neural information processing systems, pp 2962–2970

130. Fielding B, Zhang L (2020) Evolving deep denseblock architecture ensembles for image classiﬁcation. Electronics (Switzerland) 9:1–31. <https://doi.org/10.3390/electronics9111880>

131. Fielding B, Lawrence T, Zhang L (2019) Evolving and ensembling deep CNN architectures for image classiﬁcation. In: International joint conference on neural networks. [https://doi.org/10.1109/IJCNN.](https://doi.org/10.1109/IJCNN.2019.8852369) [2019.8852369](https://doi.org/10.1109/IJCNN.2019.8852369)

132. Figueredo M, Paraiso E, Nievola J (2016) A constructive algorithm for neural networks inspired on decision trees and evolutionary algorithms. In: International annual conference on neural information processing systems, pp 1120–1127. <https://doi.org/10.1109/IJCNN.2016.7727323>

133. Florea AC, Andonie R (2019) Weighted random search for hyperparameter optimization. Int J Comput Commun Control 14:154–169. <https://doi.org/10.15837/ijccc.2019.2.3514>

134. Franceschi L, Frasconi P, Salzo S et al (2018) Bilevel programming for hyperparameter optimization and meta-learning. In: International conference on machine learning, pp 2537–2548

135. FusiN, ShethR, Elibol M (2018) Probabilistic matrix factorization for automated machine learning. In: Annual conference on neural information processing systems, pp 3348–3357

136. GallicchioC, MicheliA, PedrelliL (2018) Design of deep echo state networks. Neural Netw 108:33–47.

<https://doi.org/10.1016/j.neunet.2018.08.002>

137. Gao C, Chen Y, Liu S et al (2020a) AdversarialNAS: adversarial neural architecture search for GANs. In: IEEE/CVF conference on computer vision and pattern recognition, pp 5679–5688. [https://doi.org/](https://doi.org/10.1109/CVPR42600.2020.00572) [10.1109/CVPR42600.2020.00572](https://doi.org/10.1109/CVPR42600.2020.00572)

138. Gao Y, Bai H, Jie Z etal (2020b) MTL-NAS: task-agnostic neural architecture search towards general- purpose multi-task learning. In: IEEE/CVF conference on computer vision and pattern recognition, pp 11540–11549. <https://doi.org/10.1109/CVPR42600.2020.01156>

139. Gao Y, Yang H, ZhangPetal(2020c)Graphneuralarchitecturesearch. In:InternationalJointconferences on artiﬁcial intelligence, pp 1403–1409. <https://doi.org/10.24963/ijcai.2020/195>

140. Garcia L, Lorena A, Matwin S et al (2016) Ensembles of label noise ﬁlters: a ranking approach. Data Min Knowl Discov 30:1192–1216. <https://doi.org/10.1007/s10618-016-0475-9>

141. Gardner J, Kusner M, Xu Z et al (2014) Bayesian optimization with inequality constraints. In: International Conference on Machine Learning, pp 2581–2591

142. Garro B, Vázquez R (2015) Designing artiﬁcial neural networks using particle swarm optimization algorithms. Comput Intell Neurosci. <https://doi.org/10.1155/2015/369298>

143. Gelbart M, Snoek J, Adams R (2014) Bayesian optimization with unknown constraints. In: Conference on uncertainity in artiﬁcial intelligence, pp 250–259

144. Golovin D, Solnik B, Moitra Set al (2017) Google vizier: a service for black-box optimization. In: ACM SIGKDD international conference on knowledge discovery and data mining, pp 1487–1496.[https://doi.](https://doi.org/10.1145/3097983.3098043) [org/10.1145/3097983.3098043](https://doi.org/10.1145/3097983.3098043)

145. Gong C, Jiang Z, Wang D et al (2019) Mixed precision neural architecture search for energy efﬁcient deep learning. In: IEEE/ACM international conference on computer design, pp 1–7. [https://doi.org/10.](https://doi.org/10.1109/ICCAD45719.2019.8942147) [1109/ICCAD45719.2019.8942147](https://doi.org/10.1109/ICCAD45719.2019.8942147)

146. González J, Dai Z, Hennig P et al (2016) Batch bayesian optimization via local penalization. In: International conference on artiﬁcial intelligence and statistics, pp 648–657

147. Gordon A, Eban E, Nachum O et al (2018) MorphNet: fast and simple resource-constrained structure learning of deep networks. In: IEEE/CVF conference on computer vision and pattern recognition, pp 1586–1595. <https://doi.org/10.1109/CVPR.2018.00171>

148. Goswami S, ChakrabartiA, Chakraborty B (2016) A proposal for recommendation of feature selection algorithm based on data set characteristics. J Univ Comput Sci 22:760–781. [https://doi.org/10.3217/](https://doi.org/10.3217/jucs-022-06-0760) [jucs-022-06-0760](https://doi.org/10.3217/jucs-022-06-0760)

1 灬

149. Green S, Vineyard C, Helinski R et al (2020) RAPDARTS: resource-aware progressive differen- tiable architecture search. In: International joint conference on neural network. [https://doi.org/10.1109/](https://doi.org/10.1109/IJCNN48605.2020.9206969) [IJCNN48605.2020.9206969](https://doi.org/10.1109/IJCNN48605.2020.9206969)

150. Gulcu A, KusZ(2020)Hyper-parameterselectionin convolutionalneuralnetworksusingmicrocanonical optimization algorithm. IEEE Access 8:52528–52540.<https://doi.org/10.1109/ACCESS.2020.2981141>

151. Guo B, Hu J, Wu W etal (2019) The Tabu genetic algorithm: a novel method for hyper-parameter opti- mization of learning algorithms. Electronics (Switzerland). <https://doi.org/10.3390/electronics8050579>

152. Guo M, Yang Y, Xu R etal (2020a) When NAS meets robustness: search of robust architectures against adversarial attacks. In: IEEE/CVF conference on computer vision and pattern recognition, pp 628–637.

<https://doi.org/10.1109/CVPR42600.2020.00071>

153. Guo R, Lin C, Li C et al (2020b) Powering one-shot topological NAS with stabilized share-parameter proxy. In: European conference on computer vision, pp 625–641. [https://doi.org/10.1007/978-3-030-](https://doi.org/10.1007/978-3-030-58568-6_37) [58568-6\_37](https://doi.org/10.1007/978-3-030-58568-6_37)

154. Gupta T, Raza K (2020) Optimizing deep feedforward neural network architecture: a Tabu search based approach. Neural Process Lett 51:2855–2870. <https://doi.org/10.1007/s11063-020-10234-7>

155. He C, Ye H, Shen L et al (2020) MiLeNAS: efﬁcient neural architecture search via mixed-level reformu- lation. In: IEEE/CVF conference on computer vision and pattern recognition, pp 11990–11999. [https://](https://doi.org/10.1109/CVPR42600.2020.01201) [doi.org/10.1109/CVPR42600.2020.01201](https://doi.org/10.1109/CVPR42600.2020.01201)

156. He C, Tan H, Huang Set al(2021)Efﬁcient evolutionaryneuralarchitecturesearchbymodularinheritable crossover. Swarm Evol Comput. <https://doi.org/10.1016/j.swevo.2021.100894>

157. He Y, Lin J, Liu Z et al (2018) AMC: AutoML for model compression and acceleration on mobile devices. In: European conference on computer vision, pp 815–832. [https://doi.org/10.1007/978-3-030-](https://doi.org/10.1007/978-3-030-01234-2_48) [01234-2\_48](https://doi.org/10.1007/978-3-030-01234-2_48)

158. HeffetzY, Vainshtein R, Katz G et al (2020) DeepLine: AutoML tool for pipelines generation using deep reinforcement learning and hierarchical actions ﬁltering. In: ACM SIGKDD international conference on knowledge discovery and data mining, pp 2103–2113. <https://doi.org/10.1145/3394486.3403261>

159. Hernández-Lobato J, Hoffman M, Ghahramani Z (2014) Predictive entropy search for efﬁcient global optimization of black-box functions. In: Annual conference on neural information processing systems, pp 918–926

160. Hernández-Lobato J, Gelbart M, Adams R et al (2016) A general framework for constrained Bayesian optimization using information-based search. J Mach Learn Res 17:28

161. Higuchi K, Sano S, Igarashi T (2021) Interactive hyperparameter optimization with paintable timelines. In: ACM designing interactive systems nowhere and everywhere, pp 1518–1528. [https://doi.org/10.](https://doi.org/10.1145/3461778.3462077) [1145/3461778.3462077](https://doi.org/10.1145/3461778.3462077)

162. Hoffman M, Shahriari B, De Freitas N (2014) On correlation and budget constraints in model-based bandit optimization with application to automatic machine learning. In: International conference on artiﬁcial intelligence and statistics, pp 365–374

163. Hong W,LiG,Zhang Wetal(2020)DropNAS:groupedoperationdropoutfordifferentiablearchitecture search. In: International Joint conferences on artiﬁcial intelligence, pp 2326–2332. [https://doi.org/10.](https://doi.org/10.24963/ijcai.2020/322) [24963/ijcai.2020/322](https://doi.org/10.24963/ijcai.2020/322)

164. Horn D, Schork K, Wagner T (2016) Multi-objective selection of algorithm portfolios: experimental validation. In: International conference in parallel problem solving from nature, pp 421–430.[https://doi.](https://doi.org/10.1007/978-3-319-45823-6_39) [org/10.1007/978-3-319-45823-6\_39](https://doi.org/10.1007/978-3-319-45823-6_39)

165. Horvath S, Klein A, Richtarik P et al (2021) Hyperparameter transfer learning with adaptive complexity. In: International conference on artiﬁcial intelligence and statistics, pp 1378–1386

166. Hosseini R, Yang X, Xie P (2021) DsRNA: differentiable search of robust neural architectures. In: IEEE/CVF IEEE conference on computer vision and pattern recognition, pp 6192–6201. [https://doi.](https://doi.org/10.1109/CVPR46437.2021.00613) [org/10.1109/CVPR46437.2021.00613](https://doi.org/10.1109/CVPR46437.2021.00613)

167. HsuK, Nock R, Ramos F (2019) Hyperparameter learning for conditional kernel mean embeddings with Rademacher complexity bounds. In: European conference in machine learning and knowledge discovery in databases, pp 227–242. [https://doi.org/10.1007/978-3-030-10928-8\_ 14](https://doi.org/10.1007/978-3-030-10928-8_14)

168. Hu K, Bakker M, Li S et al (2019a) VizML: a machine learning approach to visualization recom- mendation. In: Conference on human factors in computing systems. [https://doi.org/10.1145/3290605.](https://doi.org/10.1145/3290605.3300358) [3300358](https://doi.org/10.1145/3290605.3300358)

169. Hu S, Cheng R, He C et al (2021) Accelerating multi-objective neural architecture search by random- weight evaluation. Complex Intell Syst. <https://doi.org/10.1007/s40747-021-00594-5>

170. Hu W, Jin J, Liu TY et al (2020) Automatically design convolutional neural networks by optimization with submodularity and supermodularity. IEEE Trans Neural Netw Learn Syst 31:3215–3229. [https://](https://doi.org/10.1109/TNNLS.2019.2939157) [doi.org/10.1109/TNNLS.2019.2939157](https://doi.org/10.1109/TNNLS.2019.2939157)

171. HuW, LiM, Yuan Cet al(2020b)Diversityinneuralarchitecturesearch. In:Internationaljointconference on neural network. <https://doi.org/10.1109/IJCNN48605.2020.9206793>

172. Hu W, Zhou A, Zhang G (2020c) A classiﬁcation surrogate model based evolutionary algorithm for neural network structure learning. In: International joint conference on neural network. [https://doi.org/](https://doi.org/10.1109/IJCNN48605.2020.9207558) [10.1109/IJCNN48605.2020.9207558](https://doi.org/10.1109/IJCNN48605.2020.9207558)

173. Hu Y, Jiang X, Liu X et al (2020d) NAS-count: counting-by-density with neural architecture search. In: European conference on computer vision, pp 747–766.<https://doi.org/10.1007/978-3-030-58542-6_45>

174. Hu Y, Liang Y, Guo Z et al (2020e) Angle-based search space shrinking for neural architecture search. In: European conference on computer vision, pp 119–134. [https://doi.org/10.1007/978-3-030-58529-](https://doi.org/10.1007/978-3-030-58529-7_8) [7\_8](https://doi.org/10.1007/978-3-030-58529-7_8)

175. Hu Y, Wu X, He R (2020f) TF-NAS: rethinking three search freedoms of latency-constrained differen- tiable neural architecture search. In: European conference on computer vision, pp 123–139. [https://doi.](https://doi.org/10.1007/978-3-030-58555-6_8) [org/10.1007/978-3-030-58555-6\_8](https://doi.org/10.1007/978-3-030-58555-6_8)

176. Hu Y, Liu X, Li S et al (2021) Cascaded algorithm selection with extreme-region UCB bandit. IEEE Trans Pattern Anal Mach Intell. <https://doi.org/10.1109/TPAMI.2021.3094844>

177. Hu Y, Wang X, Li L et al (2021) Improving one-shot NAS with shrinking-and-expanding supernet. Pattern Recognit. <https://doi.org/10.1016/j.patcog.2021.108025>

178. Hu YQ, Yu Y, Zhou ZH (2018) Experienced optimization with reusable directional model for hyper- parameter search. In: International Joint conferences on artiﬁcial intelligence, pp 2276–2282. [https://](https://doi.org/10.24963/ijcai.2018/315) [doi.org/10.24963/ijcai.2018/315](https://doi.org/10.24963/ijcai.2018/315)

179. Hu YQ, Yu Y, Liao JD (2019b) Cascaded algorithm-selection and hyper-parameter optimization with extreme-region upper conﬁdence bound bandit. In: Joint conferences on artiﬁcial intelligence, pp 2528–

2534. <https://doi.org/10.24963/ijcai.2019/351>

180. Hu YQ, Yu Y, Tu WW etal (2019c) Multi-ﬁdelity automatic hyper-parameter tuning via transfer series expansion. In: AAAI conferences on artiﬁcial intelligence, pp 3846–3853.[https://doi.org/10.1609/aaai.](https://doi.org/10.1609/aaai.v33i01.33013846) [v33i01.33013846](https://doi.org/10.1609/aaai.v33i01.33013846)

181. Huang S, Li X, Cheng ZQ et al (2018) GNAS: a greedy neural architecture search method for multi- attribute learning. In: ACM multimedia conference, pp 2049–2057. [https://doi.org/10.1145/3240508.](https://doi.org/10.1145/3240508.3240588) [3240588](https://doi.org/10.1145/3240508.3240588)

182. IlievskiI, Akhtar T, Feng Jet al(2017)Efﬁcienthyperparameteroptimization ofdeep learningalgorithms using deterministic RBF surrogates. In: AAAI conferences on artiﬁcial intelligence, pp 822–829

183. Jaddi N, Abdullah S, Hamdan A (2015) Optimization of neural network model using modiﬁed bat- inspired algorithm. Appl Soft Comput J 37:71–86. <https://doi.org/10.1016/j.asoc.2015.08.002>

184. Jamieson K,TalwalkarA(2016)Non-stochasticbest arm identiﬁcation andhyperparameteroptimization. In: International conference on artiﬁcial intelligence and statistics, pp 240–248

185. Ji Y, Chen Y, Fu H et al (2017) An EnKF-based scheme to optimize hyper-parameters and features for SVM classiﬁer. Pattern Recognit 62:202–213. <https://doi.org/10.1016/j.patcog.2016.08.014>

186. Jiang J, Han F, Ling Q et al (2020) Efﬁcient network architecture search via multiobjective parti- cle swarm optimization based on decomposition. Neural Netw 123:305–316. [https://doi.org/10.1016/j.](https://doi.org/10.1016/j.neunet.2019.12.005) [neunet.2019.12.005](https://doi.org/10.1016/j.neunet.2019.12.005)

187. Jiang W, Yang L, Sha EM etal (2020) Hardware/software co-exploration of neural architectures. IEEE Trans Comput Des Integr Circuits Syst 39:4805–4815. <https://doi.org/10.1109/TCAD.2020.2986127>

188. Jie R, Gao J(2021)Differentiableneuralarchitecturesearchforhigh-dimensionaltime seriesforecasting. IEEE Access 9:20922–20932. <https://doi.org/10.1109/ACCESS.2021.3055555>

189. Jie R, Gao J, Vasnev A et al (2020) HyperTube: a framework for population-based online hyperparameter optimization with resource constraints. IEEE Access8:69038–69057.[https://doi.org/10.1109/ACCESS.](https://doi.org/10.1109/ACCESS.2020.2986456) [2020.2986456](https://doi.org/10.1109/ACCESS.2020.2986456)

190. Jing K,Xu J,Zhang Z(2021)A neural architecturegeneratorfor efﬁcient search space. Neurocomputing.

<https://doi.org/10.1016/j.neucom.2021.10.118>

191. Jing W, Ren Q, Zhou J et al (2020) AutoRSISC: automatic design of neural architecture for remote sensing image scene classiﬁcation. Pattern RecognitLett 140:186–192.[https://doi.org/10.1016/j.patrec.](https://doi.org/10.1016/j.patrec.2020.09.034) [2020.09.034](https://doi.org/10.1016/j.patrec.2020.09.034)

192. Jlassi S,JdeyI, Ltiﬁ H (2021) Bayesian hyperparameter optimization of deep neural network algorithms based on ant colony optimization. In: International conference on document analysis and recognition, pp 585–594. <https://doi.org/10.1007/978-3-030-86334-0_38>

193. Johnson F, Valderrama A, Valle C et al (2020) Automating conﬁguration of convolutional neural net- work hyperparameters using genetic algorithm. IEEE Access 8:156139–156152. [https://doi.org/10.](https://doi.org/10.1109/ACCESS.2020.3019245) [1109/ACCESS.2020.3019245](https://doi.org/10.1109/ACCESS.2020.3019245)

194. Joy T, Rana S, Gupta S et al (2020) Batch Bayesian optimization using multi-scale search. Knowl Syst.

<https://doi.org/10.1016/j.knosys.2019.06.026>

1 灬

195. Joy T, Rana S, Gupta S et al (2020) Fast hyperparameter tuning using Bayesian optimization with directional derivatives. Knowl Syst. <https://doi.org/10.1016/j.knosys.2020.106247>

196. Kandanaarachchi S, Muñoz M, Hyndman R et al (2019) On normalization and algorithm selection for unsupervised outlier detection. Data Min Knowl Discov. <https://doi.org/10.1007/s10618-019-00661-z>

197. Kandasamy K, Schneider J, Póczos B (2015) High dimensional Bayesian optimisation and bandits via additive models. In: International conference on machine learning, pp 295–304

198. Kandasamy K, Dasarathy G, Schneider J et al (2017) Multi-ﬁdelity Bayesian optimisation with continuous approximations. In: International conference on machine learning, pp 2861–2878

199. Kandasamy K, Neiswanger W, Schneider J et al (2018) Neural architecture search with Bayesian opti- misation and optimal transport. In: Annual conference on neural information processing systems, pp 2016–2025

200. KanwalS, YounasI, BashirM(2021)Evolving convolutional autoencodersusingmulti-objective particle swarm optimization. Comput Electr Eng. <https://doi.org/10.1016/j.compeleceng.2021.107108>

201. Kapanova K, Dimov I, Sellier J (2018) A genetic approach to automatic neural network architecture optimization. Neural Comput and Appl 29:1481–1492. <https://doi.org/10.1007/s00521-016-2510-6>

202. Kathuria T, Deshpande A, Kohli P (2016) Batched Gaussian process bandit optimization via deter- minantal point processes. In: Annual conference on neural information processing systems, pp 4213–4221

203. Katz G, Shin E, Song D (2017) ExploreKit: automatic feature generation and selection. In: IEEE international Conference on Data Mining, pp 979–984. <https://doi.org/10.1109/ICDM.2016.176>

204. Katz M, Ram P, Sohrabi S et al (2020) Exploring context-free languages via planning: the case for automating machine learning. In: International conference on automated planning and scheduling, pp 403–411. <https://doi.org/10.1609/icaps.v30i1.6686>

205. KhiariJ, Moreira-Matias L, Shaker A et al (2019) MetaBags: bagged meta-decision trees for regression. In:European conferenceinmachinelearningandknowledgediscovery indatabases, pp637–652.[https://](https://doi.org/10.1007/978-3-030-10925-7_39) [doi.org/10.1007/978-3-030-10925-7\_39](https://doi.org/10.1007/978-3-030-10925-7_39)

206. Khurana U, Samulowitz H, Turaga D (2018) Feature engineering for predictive modeling using rein- forcement learning. In: AAAI conferences on artiﬁcial intelligence, pp 3407–3414. [https://doi.org/10.](https://doi.org/10.1609/aaai.v32i1.11678) [1609/aaai.v32i1.11678](https://doi.org/10.1609/aaai.v32i1.11678)

207. KietzJU,SerbanF, Fischer S et al (2014) “Semantics inside!” but let’s not tell the data miners: intelligent support for data mining. In: International conference in the semantic web: trends and challenges, pp 706–720. <https://doi.org/10.1007/978-3-319-07443-6_47>

208. Kim Y, Chung M (2019) An approach to hyperparameter optimization for the objective function in machine learning. Electronics (Switzerland). <https://doi.org/10.3390/electronics8111267>

209. Klein A, Falkner S, Bartels S et al (2017) Fast Bayesian optimization of machine learning hyperparam- eters on large datasets. In: International conference on artiﬁcial intelligence and statistics

210. Koch P, Wujek B, Golovidov O et al (2018) Autotune: a derivative-free optimization framework for hyperparameter tuning. In: ACM SIGKDD conference on knowledge discovery and data mining, pp 443–452. <https://doi.org/10.1145/3219819.3219837>

211. Kordík P, erný J, Frýda T (2018) Discovering predictive ensembles for transfer learning and meta- learning. Mach Learn 107:177–207. <https://doi.org/10.1007/s10994-017-5682-0>

212. Kotthoff L, Thornton C, Hoos H et al (2017) Auto-WEKA 2.0: automatic model selection and hyperparameter optimization in WEKA. J Mach Learn Res 18:1–5

213. KuckM, CroneS, Freitag M (2016)Meta-learning with neural networks and landmarking for forecasting model selection an empirical evaluation of different feature sets applied to industry data. In: International joint conference on neural networks, pp 1499–1506. <https://doi.org/10.1109/IJCNN.2016.7727376>

214. Kumar P, BatraS, Raman B(2021)Deepneuralnetworkhyper-parametertuningthroughtwofold genetic approach. Soft Comput 25(13):8747–8771. <https://doi.org/10.1007/s00500-021-05770-w>

215. Lacoste A, Larochelle H, Marchand M et al (2014) Sequential model-based ensemble optimization. In: Conference on uncertainity in artiﬁcial intelligence, pp 440–448

216. Lakhmiri D, Digabel S, Tribes C (2021) HyperNOMAD: hyperparameter optimization of deep neural networks using mesh adaptive direct search. ACM Trans Math Softw. <https://doi.org/10.1145/3450975>

217. LaubeK, ZellA (2019) ShufﬂeNASNets: efﬁcient CNN models through modiﬁed efﬁcient neural archi- tecture search. In: International joint conference on neural networks. [https://doi.org/10.1109/IJCNN.](https://doi.org/10.1109/IJCNN.2019.8852294) [2019.8852294](https://doi.org/10.1109/IJCNN.2019.8852294)

218. Lawrence T, Zhang L, Lim C et al (2021) Particle swarm optimization for automatically evolving convolutional neural networks for image classiﬁcation. IEEE Access 9:14369–14386. [https://doi.org/](https://doi.org/10.1109/ACCESS.2021.3052489) [10.1109/ACCESS.2021.3052489](https://doi.org/10.1109/ACCESS.2021.3052489)

219. Lawrence T, Zhang L, Rogage K et al (2021) Evolving deep architecture generation with residual connectionsforimageclassiﬁcationusingparticleswarm optimization. Sensors.[https://doi.org/10.3390/](https://doi.org/10.3390/s21237936) [s21237936](https://doi.org/10.3390/s21237936)

220. Lee J, Rhim J, Kang Detal (2021) S3NAS: fast hardware-aware neural architecture search methodology. IEEE Trans Comput Des Integr Circuits Syst. <https://doi.org/10.1109/TCAD.2021.3134843>

221. Lee S, Kim J, Kang H et al (2021) Genetic algorithm based deep learning neural network structure and hyperparameter optimization. Appl Sci (Switzerland) 11(2):1–12.<https://doi.org/10.3390/app11020744>

222. Lévesque JC, GagnéC, Sabourin R(2016)Bayesianhyperparameteroptimizationfor ensemblelearning. In: Conference on uncertainity in artiﬁcial intelligence, pp 437–446

223. Li C, Gupta S, Rana S et al (2017) High dimensional Bayesian optimization using dropout. In: Inter- national joint conferences on artiﬁcial intelligence, pp 2096–2102. [https://doi.org/10.24963/ijcai.2017/](https://doi.org/10.24963/ijcai.2017/291) [291](https://doi.org/10.24963/ijcai.2017/291)

224. Li C, Fan X, Geng Y et al (2020) ENAS oriented layer adaptive data scheduling strategy for resource limited hardware. Neurocomputing 381:29–39. <https://doi.org/10.1016/j.neucom.2019.11.005>

225. Li CL, Kandasamy K, Póczos B et al (2016) High dimensional Bayesian optimization via restricted projectionpursuitmodels. In:International conference on artiﬁcialintelligenceand statistics,pp884–892

226. Li G, Qian G, Delgadillo I et al (2020b) SGAS: sequential greedy architecture search. In: IEEE/CVF conference on computer vision and pattern recognition, pp 1617–1627. [https://doi.org/10.1109/](https://doi.org/10.1109/CVPR42600.2020.00169) [CVPR42600.2020.00169](https://doi.org/10.1109/CVPR42600.2020.00169)

227. LiG, MandalS,OgrasUet al(2021)FLASH:fastneuralarchitecturesearchwithhardware optimization. ACM Trans Embed Comput Syst. <https://doi.org/10.1145/3476994>

228. Li L, Talwalkar A (2019) Random search and reproducibility for neural architecture search. In: Conference on uncertainity in artiﬁcial intelligence

229. LiL, Jamieson K, DeSalvo Get al (2018) Hyperband: a novel bandit-based approach to hyperparameter optimization. J Mach Learn Res 18:1–52

230. Li S, Sun Y, Yen G et al (2021) Automatic design of convolutional neural network architectures under resource constraints. IEEE Trans Neural Netw Learn Syst.[https://doi.org/10.1109/TNNLS.2021.](https://doi.org/10.1109/TNNLS.2021.3123105) [3123105](https://doi.org/10.1109/TNNLS.2021.3123105)

231. Li T, Zhong J, Liu J, et al (2018b) Ease.ml: towards multitenant resource sharing for machine learning workloads. In: International conference on very large databases, pp 607–620. [https://doi.org/10.1145/](https://doi.org/10.1145/3177732.3177737) [3177732.3177737](https://doi.org/10.1145/3177732.3177737)

232. Li T, Zhang J, Bao K et al (2020c) AutoST: efﬁcient neural architecture search for spatio-temporal prediction. In: ACM SIGKDD international conference on knowledge discovery and data mining, pp 794–802. <https://doi.org/10.1145/3394486.3403122>

233. Li X, Lin C, Li C et al (2020d) Improving one-shot NAS by suppressing the posterior fading. In: IEEE/CVF conference on computer vision and pattern recognition, pp 13833–13842.[https://doi.org/10.](https://doi.org/10.1109/CVPR42600.2020.01385) [1109/CVPR42600.2020.01385](https://doi.org/10.1109/CVPR42600.2020.01385)

234. Li Y, King I (2020) AutoGraph: automated graph neural network. In: International conference on neural information processing, pp 189–201. [https://doi.org/10.1007/978-3-030-63833-7\_ 16](https://doi.org/10.1007/978-3-030-63833-7_16)

235. Li Y, Xiao J, Chen Y et al (2019) Evolving deep convolutional neural networks by quantum behaved particle swarm optimization with binary encoding for image classiﬁcation. Neurocomputing 362:156–

165. <https://doi.org/10.1016/j.neucom.2019.07.026>

236. Li Y, Jin X, Mei J et al (2020e) Neural architecture search for lightweight non-local networks. In: IEEE/CVF conference on computer vision and pattern recognition, pp 10294–10303.[https://doi.org/10.](https://doi.org/10.1109/CVPR42600.2020.01031) [1109/CVPR42600.2020.01031](https://doi.org/10.1109/CVPR42600.2020.01031)

237. Li Y, Chen Z, Zha D et al (2021c) AutoOD: neural architecture search for outlier detection. In: IEEE international conference on data engineering, pp 2117–2122.[https://doi.org/10.1109/ICDE51399.2021.](https://doi.org/10.1109/ICDE51399.2021.00210) [00210](https://doi.org/10.1109/ICDE51399.2021.00210)

238. Li Y, Dong M, Xu Y et al (2021) Neural architecture tuning with policy adaptation. Neurocomputing.

<https://doi.org/10.1016/j.neucom.2021.10.095>

239. Li Z, Xi T, Deng J et al (2020f) GP-NAS: Gaussian process based neural architecture search. In: IEEE/CVF conference on computer vision and pattern recognition, pp 11930–11939. [https://doi.org/](https://doi.org/10.1109/CVPR42600.2020.01195) [10.1109/CVPR42600.2020.01195](https://doi.org/10.1109/CVPR42600.2020.01195)

240. Liang J, Meyerson E, Hodjat B et al (2019) Evolutionary neural automl for deep learning. In: Genetic and evolvable computing conference, pp 401–409. <https://doi.org/10.1145/3321707.3321721>

241. Liang Y, Lu L, Jin Y et al (2021) An efﬁcient hardware design for accelerating sparse CNNs with NAS- based models. IEEE Trans Comput Des Integr Circuits Syst 41(3):597–613. [https://doi.org/10.1109/](https://doi.org/10.1109/TCAD.2021.3066563) [TCAD.2021.3066563](https://doi.org/10.1109/TCAD.2021.3066563)

242. Liu C,ZophB, Neumann Metal(2018a)Progressive neuralarchitecturesearch. In:Europeanconference on computer vision, pp 19–35. <https://doi.org/10.1007/978-3-030-01246-5_2>

1 灬

243. Liu H, Nicolae B, Di S et al (2021a) Accelerating DNN architecture search at scale using selective weight transfer. In: IEEE international conference on cluster computing, pp 82–93. [https://doi.org/10.](https://doi.org/10.1109/Cluster48925.2021.00051) [1109/Cluster48925.2021.00051](https://doi.org/10.1109/Cluster48925.2021.00051)

244. Liu J, Jin Y (2021) Multi-objective search of robust neural architectures against multiple types of adversarial attacks. Neurocomputing 453:73–84. <https://doi.org/10.1016/j.neucom.2021.04.111>

245. Liu J, Gong M, Miao Q et al (2018) Structure learning for deep neural networks based on multiobjec- tive optimization. IEEE Trans Neural Netw Learn Syst 29:2450–2463.[https://doi.org/10.1109/TNNLS.](https://doi.org/10.1109/TNNLS.2017.2695223) [2017.2695223](https://doi.org/10.1109/TNNLS.2017.2695223)

246. Liu J, Zhou S, Wu Y et al (2021) Block proposal neural architecture search. IEEE Trans Image Process 30:15–25. <https://doi.org/10.1109/TIP.2020.3028288>

247. Liu Y, Wang X, Xu X et al (2021c) Meta hyperparameter optimization with adversarial proxy subsets sampling. In: Conference on information and knowledge management, pp 1109–1118. [https://doi.org/](https://doi.org/10.1145/3459637.3482368) [10.1145/3459637.3482368](https://doi.org/10.1145/3459637.3482368)

248. Lloyd J, Duvenaud D, Grosse R et al (2014) Automatic construction and natural-language description of nonparametric regression models. In: AAAI conference on artiﬁcial intelligence, pp 1242–1250

249. Loni M,SinaeiS, Zoljodi A et al (2020) DeepMaker: a multi-objective optimization framework for deep neural networks in embedded systems. Microprocess Microsyst.[https://doi.org/10.1016/j.micpro.2020.](https://doi.org/10.1016/j.micpro.2020.102989) [102989](https://doi.org/10.1016/j.micpro.2020.102989)

250. Lorenzo P, Nalepa J (2018) Memetic evolution of deep neural networks. In: Genetic and evolvable computing conference, pp 505–512. <https://doi.org/10.1145/3205455.3205631>

251. Lorenzo P, Nalepa J, Kawulok M et al (2017) Particle swarm optimization for hyper-parameterselection in deep neural networks. In: Genetic and evolvable computing conference, pp 481–488. [https://doi.org/](https://doi.org/10.1145/3071178.3071208) [10.1145/3071178.3071208](https://doi.org/10.1145/3071178.3071208)

252. Louati H, Bechikh S, Louati A et al (2021) Deep convolutional neural network architecture design as a bi-level optimization problem. Neurocomputing 439:44–62. [https://doi.org/10.1016/j.neucom.2021.01.](https://doi.org/10.1016/j.neucom.2021.01.094) [094](https://doi.org/10.1016/j.neucom.2021.01.094)

253. Lu H, DuM, HeX et al(2021)An adaptive neuralarchitecturesearchdesignfor collaborative edge-cloud computing. IEEE Netw 35(5):83–89. <https://doi.org/10.1109/MNET.201.2100069>

254. Lu X, Huang H, Dong W et al (2020a) Beyond network pruning: a joint search-and-training approach. In: International joint conference on artiﬁcial intelligence, pp 2583–2590.[https://doi.org/10.24963/ijcai.](https://doi.org/10.24963/ijcai.2020/358) [2020/358](https://doi.org/10.24963/ijcai.2020/358)

255. Lu Z, Chen L, Chiang CK et al (2019a) Hyper-parameter tuning under a budget constraint. In: Inter- national joint conference on artiﬁcial intelligence, pp 5744–5750. [https://doi.org/10.24963/ijcai.2019/](https://doi.org/10.24963/ijcai.2019/796) [796](https://doi.org/10.24963/ijcai.2019/796)

256. Lu Z, Whalen I, Boddeti V et al (2019b) NSGA-Net: neural architecture search using multi-objective genetic algorithm. In: Genetic and evolvable computing conference, pp 419–427. [https://doi.org/10.](https://doi.org/10.1145/3321707.3321729) [1145/3321707.3321729](https://doi.org/10.1145/3321707.3321729)

257. Lu Z, Deb K, Goodman E et al (2020b) NSGANetV2: evolutionary multi-objective surrogate-assisted neural architecture search. In: European conference on computer vision, pp 35–51. [https://doi.org/10.](https://doi.org/10.1007/978-3-030-58452-8_3) [1007/978-3-030-58452-8\_3](https://doi.org/10.1007/978-3-030-58452-8_3)

258. LuZ,Sreekumar G, Goodman E et al (2021) Neural architecture transfer. IEEE Trans Pattern Anal Mach Intell 43(9):2971–2989. <https://doi.org/10.1109/TPAMI.2021.3052758>

259. Lu Z, Whalen I, Dhebar Y et al (2021) Multiobjective evolutionary design of deep convolutional neural networks for image classiﬁcation. IEEE Trans Evol Comput 25(2):277–291. [https://doi.org/10.1109/](https://doi.org/10.1109/TEVC.2020.3024708) [TEVC.2020.3024708](https://doi.org/10.1109/TEVC.2020.3024708)

260. Lujan-Moreno G, Howard P, Rojas O et al (2018) Design of experiments and response surface method- ology to tune machine learning hyperparameters, with a random forest case-study. Expert Syst Appl 109:195–205. <https://doi.org/10.1016/j.eswa.2018.05.024>

261. Luo R, Tian F, Qin T et al (2018a) Neural architecture optimization. In: Annual conference on neural information processing systems, pp 7816–7827

262. Luo X, Liu D, Huai S et al (2021) Designing efﬁcient DNNs via hardware-aware neural architecture search and beyond. IEEE Trans Comput Des Integr Circuits Syst. [https://doi.org/10.1109/TCAD.2021.](https://doi.org/10.1109/TCAD.2021.3100249) [3100249](https://doi.org/10.1109/TCAD.2021.3100249)

263. Luo Y, QinX, Tang Net al(2018b)Deepeye:towards automaticdatavisualization. In:IEEE international conference on data engineering, pp 101–112. <https://doi.org/10.1109/ICDE.2018.00019>

264. Luo Y, Yao Q, Wang M et al (2019) Autocross: automatic feature crossing for tabular data in real-world applications. In: ACM SIGKDD international conference on knowledge discovery and data mining, pp 1936–1945. <https://doi.org/10.1145/3292500.3330679>

265. Ma B, Li X, Xia Y et al (2020) Autonomous deep learning: a genetic DCNN designer for image classiﬁcation. Neurocomputing 379:152–161. <https://doi.org/10.1016/j.neucom.2019.10.007>

266. Maclaurin D, Duvenaud D, Adams R (2015) Gradient-based hyperparameter optimization through reversible learning. In: International conference on machine learning, ICML’15, pp 2113–2122

267. Maher M, SakrS (2019) SmartML: a metalearning-based framework for automated selection and hyper- parameter tuning for machine learning algorithms. In: International conference on extending database technology, pp 554–557. <https://doi.org/10.5441/002/edbt.2019.54>

268. Mahule R, Vyas O (2016) Leveraging linked open data information extraction for data mining applications. Turkish J ElectrEng Comput Sci 24:4874–4884. <https://doi.org/10.3906/elk-1412-28>

269. Mantovani R, Rossi A, Vanschoren J et al (2015) To tune or not to tune: recommending when to adjust SVM hyper-parameters via meta-learning. In: International joint conference on neural networks.[https://](https://doi.org/10.1109/IJCNN.2015.7280644) [doi.org/10.1109/IJCNN.2015.7280644](https://doi.org/10.1109/IJCNN.2015.7280644)

270. MantovaniR, RossiA,AlcobaçaEet al(2019)A meta-learningrecommender systemforhyperparameter tuning: predicting when tuning improves SVM classiﬁers. Inf Sci 501:193–221.[https://doi.org/10.1016/](https://doi.org/10.1016/j.ins.2019.06.005) [j.ins.2019.06.005](https://doi.org/10.1016/j.ins.2019.06.005)

271. Marchisio A, Massa A, Mrazek V et al (2020) NASCaps: a framework for neural architecture search to optimize the accuracy and hardware efﬁciency of convolutional capsule networks. In: IEEE/ACM international conference on computer design. <https://doi.org/10.1145/3400302.3415731>

272. Martín A, Lara-Cabrera R, Fuentes-Hurtado F et al (2018) EvoDeep: a new evolutionary approach for automatic deep neural networks parametrisation. J Parallel Distrib Comput 117:180–191. [https://doi.](https://doi.org/10.1016/j.jpdc.2017.09.006) [org/10.1016/j.jpdc.2017.09.006](https://doi.org/10.1016/j.jpdc.2017.09.006)

273. Martin Salvador M, Budka M, Gabrys B (2019) Automatic composition and optimization of multi- component predictive systems with an extended auto-WEKA. IEEE Trans Autom Sci Eng 16:946–959.

<https://doi.org/10.1109/TASE.2018.2876430>

274. Martinez-Cantin R (2019) Funneled Bayesian optimization for design, tuning and control of autonomous systems. IEEE Trans Cybern 49:1489–1500. <https://doi.org/10.1109/TCYB.2018.2805695>

275. Massimo C,NavarinN, SperdutiA (2016) Hyper-parameter tuning for graph kernels via multiple kernel learning. In: International conference on neural information processing, pp 214–223. [https://doi.org/10.](https://doi.org/10.1007/978-3-319-46672-9_25) [1007/978-3-319-46672-9\_25](https://doi.org/10.1007/978-3-319-46672-9_25)

276. McInerney J (2017) An empirical bayes approach to optimizing machine learning algorithms. In: Annual conference on neural information processing systems, pp 2713–2722

277. MendisH, Kang CK, HsiuPC (2021) Intermittent-aware neural architecture search. ACM Trans Embed Comput Syst. <https://doi.org/10.1145/3476995>

278. MezzanzanicaM, BoselliR, Cesarini M et al (2015) A model-based evaluation of data quality activities in KDD. Inf Process Manag 51:144–166. <https://doi.org/10.1016/j.ipm.2014.07.007>

279. Miao F, Yao L, Zhao X (2021) Evolving convolutional neural networks by symbiotic organisms search algorithm for imageclassiﬁcation. Appl Soft Comput. <https://doi.org/10.1016/j.asoc.2021.107537>

280. Mills K, Han F, Salameh M et al (2021a) L2NAS: learning to optimize neural architectures via continuous-action reinforcement learning. In: ACM international conference on information and knowledge management, pp 1284–1293. <https://doi.org/10.1145/3459637.3482360>

281. Mills K, Han F, Zhang J et al (2021b) Proﬁling neural blocks and design spaces for mobile neural architecture search. In: ACM international conference on information and knowledge management, pp 4026–4035. <https://doi.org/10.1145/3459637.3481944>

282. Mills K, SalamehM, Niu Detal(2021)Exploringneural architecturesearch space viadeep deterministic sampling. IEEE Access 9:110962–110974. <https://doi.org/10.1109/ACCESS.2021.3101975>

283. Miranda P, Prudêncio R, de Carvalho A et al (2014) A hybrid meta-learning architecture for multi- objective optimization of SVM parameters. Neurocomputing 143:27–43. [https://doi.org/10.1016/j.](https://doi.org/10.1016/j.neucom.2014.06.026) [neucom.2014.06.026](https://doi.org/10.1016/j.neucom.2014.06.026)

284. Mohammadi R, Fatemi Ghomi S, Zeinali F (2014) A new hybrid evolutionary based RBF networks method for forecasting time series: a case study of forecasting emergency supply demand time series. Eng Appl Artif Intell 36:204–214. <https://doi.org/10.1016/j.engappai.2014.07.022>

285. MohrF, WeverM, HüllermeierE(2018)ML-Plan:automatedmachinelearningviahierarchicalplanning. Mach Learn 107:1495–1515. <https://doi.org/10.1007/s10994-018-5735-z>

286. Montecino D, Perez C, Bowyer K (2021) Two-level genetic algorithm for evolving convolutional neural networks for pattern recognition. IEEE Access 9:126856–126872. [https://doi.org/10.1109/ACCESS.](https://doi.org/10.1109/ACCESS.2021.3111175) [2021.3111175](https://doi.org/10.1109/ACCESS.2021.3111175)

287. Muhammad T, Halim Z (2016) Employing artiﬁcial neural networks for constructing metadata-based model to automatically select an appropriate data visualization technique. Appl Soft Comput J 49:365–

384. <https://doi.org/10.1016/j.asoc.2016.08.039>

288. Muravev A, Raitoharju J, Gabbouj M (2021) Neural architecture search by estimation of network structure distributions. IEEE Access 9:15304–15319. <https://doi.org/10.1109/ACCESS.2021.3052996>

1 灬

289. Mustaffa Z, Yusof Y (2014) LSSVM parameters tuning with enhanced artiﬁcial bee colony. Int Arab J Inf Technol 11:236–242

290. Nakai K, Matsubara T, Uehara K (2020) Att-DARTS: differentiable neural architecture search for atten- tion. In: International joint conference on neural network. [https://doi.org/10.1109/IJCNN48605.2020.](https://doi.org/10.1109/IJCNN48605.2020.9207447) [9207447](https://doi.org/10.1109/IJCNN48605.2020.9207447)

291. Nakai K, Matsubara T, Uehara K (2021) Neural architecture search for convolutional neural networks withattention. IEICE Trans InfSystE104D(2):312–321.<https://doi.org/10.1587/transinf.2020EDP7111>

292. Nambiar V, Khalil-Hani M, Marsono M et al (2014) Optimization of structure and system latency in evolvable block-based neural networks using genetic algorithm. Neurocomputing 145:285–302.[https://](https://doi.org/10.1016/j.neucom.2014.05.033) [doi.org/10.1016/j.neucom.2014.05.033](https://doi.org/10.1016/j.neucom.2014.05.033)

293. Nargesian F, SamulowitzH, Khurana U etal (2017) Learning feature engineering for classiﬁcation. In: International conference on artiﬁcial intelligence, pp 2529–2535. [https://doi.org/10.24963/ijcai.2017/](https://doi.org/10.24963/ijcai.2017/352) [352](https://doi.org/10.24963/ijcai.2017/352)

294. Nguyen DA, Kong J, Wang H et al (2021a) Improved automated CASH optimization with tree parzen estimatorsforclass imbalanceproblems. In:IEEE international conference on datascience and advanced analytics, pp 1–9. <https://doi.org/10.1109/DSAA53316.2021.9564147>

295. Nguyen P, HilarioM, Kalousis A (2014) Using meta-mining to support data mining workﬂow planning and optimization. J Artif Intell Res 51:605–644. <https://doi.org/10.1613/jair.4377>

296. Nguyen T, GuptaS, Rana S, et al (2017a) Stable bayesian optimization. In: Paciﬁc-Asia Conf. on Knowl. Discov. and Data Min., pp 578–591, <https://doi.org/10.1007/978-3-319-57529-2_45>

297. Nguyen T, Musial K, Gabrys B (2021) AutoWeka4MCPS-AVATAR: accelerating automated machine learning pipeline composition and optimisation. Expert Syst Appl 185(115):643. [https://doi.org/10.](https://doi.org/10.1016/j.eswa.2021.115643) [1016/j.eswa.2021.115643](https://doi.org/10.1016/j.eswa.2021.115643)

298. Nguyen TD, Maszczyk T, Musial K et al (2020) AVATAR-machine learning pipeline evaluation using surrogate model. In: International conference on intelligent data analysis, pp 352–365. [https://doi.org/](https://doi.org/10.1007/978-3-030-44584-3_28) [10.1007/978-3-030-44584-3\_28](https://doi.org/10.1007/978-3-030-44584-3_28)

299. Nguyen V, Rana S, Gupta S et al (2017b) Budgeted batch Bayesian optimization. In: IEEE international conference on data mining, pp 1107–1112. <https://doi.org/10.1109/ICDM.2016.52>

300. Nguyen V, Gupta S, Rana S et al (2019) Filtering Bayesian optimization approach in weakly speciﬁed search space. Knowl Inf Syst 60:385–413. <https://doi.org/10.1007/s10115-018-1238-2>

301. Ning X, Zheng Y, Zhao T, et al (2020) A Generic Graph-Based Neural Architecture Encoding Scheme for Predictor-Based NAS. In: Eur. Conf. on Comput. Vis., pp 189–204, [https://doi.org/10.1007/978-3-](https://doi.org/10.1007/978-3-030-58601-0_12) [030-58601-0\_12](https://doi.org/10.1007/978-3-030-58601-0_12)

302. Ning X, Ge G, Li W, et al (2021) FTT-NAS: Discovering Fault-tolerant Convolutional Neural Architecture. ACM Trans on Des Autom of Electronic Syst 26(6). <https://doi.org/10.1145/3460288>

303. Niu S, Wu J, Zhang Y et al (2021) Disturbance-immune weight sharing for neural architecture search. Neural Netw 144:553–564. <https://doi.org/10.1016/j.neunet.2021.09.002>

304. Nomura M, Saito Y (2021) Efﬁcient hyperparameter optimization under multi-source covariate shift. In: ACM international conference on knowledge management, pp 1376–1385. [https://doi.org/10.1145/](https://doi.org/10.1145/3459637.3482336) [3459637.3482336](https://doi.org/10.1145/3459637.3482336)

305. NoyA, NaymanN, RidnikT et al (2020) ASAP: architecture search, anneal and prune. In: International conference on artiﬁcial intelligence and statistics, pp 493–503

306. Oh C, Gavves E, Welling M (2018) BOCK: Bayesian optimization with cylindrical kernels. In: International conference on machine learning, pp 6201–6210

307. Olson RS, Bartley N, Urbanowicz RJ, Moore JH (2016) Evaluation of a tree-based pipeline optimization tool for automating data science. In: Genetic and evolvable computing conference, pp 485–492. [https://](https://doi.org/10.1145/2908812.2908918) [doi.org/10.1145/2908812.2908918](https://doi.org/10.1145/2908812.2908918)

308. O’Neill D, Xue B, Zhang M (2020) Neural architecture search for sparse DenseNets with dynamic compression. In: Genetic and evolvable computing conference, pp 386–394. [https://doi.org/10.1145/](https://doi.org/10.1145/3377930.3390178) [3377930.3390178](https://doi.org/10.1145/3377930.3390178)

309. O’Neill D, Xue B, Zhang M (2021) Evolutionary neural architecture search for high-dimensional skip- connection structures on DenseNet style networks. IEEE Trans Evol Comput 25(6):1118–1132. [https://](https://doi.org/10.1109/TEVC.2021.3083315) [doi.org/10.1109/TEVC.2021.3083315](https://doi.org/10.1109/TEVC.2021.3083315)

310. Öztürk M, Cankaya I, Ipekçi D (2020) Optimizing echo state network through a novel ﬁsher maxi- mization based stochastic gradient descent. Neurocomputing 415:215–224. [https://doi.org/10.1016/j.](https://doi.org/10.1016/j.neucom.2020.07.034) [neucom.2020.07.034](https://doi.org/10.1016/j.neucom.2020.07.034)

311. Gijsbers P, Vanschoren J (2021) GAMA: a general automated machine learning assistant. In: European conference in machine learning and knowledge discovery in databases, pp 560–564. [https://doi.org/10.](https://doi.org/10.1007/978-3-030-67670-4_39) [1007/978-3-030-67670-4\_39](https://doi.org/10.1007/978-3-030-67670-4_39)

312. Pan Z, Ke S, Yang Xetal (2021a)AutoSTG: neural architecture search for predictionsof spatio-temporal graph. In: Worldwide web conference, pp 1846–1855. <https://doi.org/10.1145/3442381.3449816>

313. Pan Z, Zeng J, Cheng R et al (2021) PNAS: a privacy preserving framework for neural architecture search services. Inf Sci 573:370–381. <https://doi.org/10.1016/j.ins.2021.05.073>

314. Pang D, Le X, Guan X (2021) RL-DARTS: differentiable neural architecture search via reinforcement- learning-based meta-optimizer. Knowl Syst. <https://doi.org/10.1016/j.knosys.2021.107585>

315. Park KM, Shin D, Chi SD (2021) Modiﬁed neural architecture search using the chromosome non- disjunction. Appl Sci (Switzerland). <https://doi.org/10.3390/app11188628>

316. Parmezan A, Lee H, Wu F (2017) Metalearning for choosing featureselection algorithms in data mining: proposal of a new framework. Expert Syst Appl 75:1–24. <https://doi.org/10.1016/j.eswa.2017.01.013>

317. Parmezan A, Lee H, Spolaôr N et al (2021) Automatic recommendation of feature selection algorithms based on dataset characteristics. Expert Syst Appl. <https://doi.org/10.1016/j.eswa.2021.115589>

318. Pasunuru R, Bansal M (2020) Continual and multi-task architecture search. In: Annual meeting of the association for computational linguistics, pp 1911–1922. <https://doi.org/10.18653/v1/P19-1185>

319. Pauletto L, Amini MR, Babbar R et al (2020) Neural architecture search for extreme multi-label text classiﬁcation. In: International conference on neural information processing, pp 282–293. [https://doi.](https://doi.org/10.1007/978-3-030-63836-8_24) [org/10.1007/978-3-030-63836-8\_24](https://doi.org/10.1007/978-3-030-63836-8_24)

320. Pedregosa F (2016) Hyperparameter optimization with approximate gradient. In: International confer- ence on machine learning, pp 1150–1159

321. Perez-Rua J, Vielzeuf V, Pateux S et al (2019) MFAS: multimodal fusion architecture search. In: IEEE/CVF conference on computer vision and pattern recognition, pp 6959–6968. [https://doi.org/10.](https://doi.org/10.1109/CVPR.2019.00713) [1109/CVPR.2019.00713](https://doi.org/10.1109/CVPR.2019.00713)

322. Perrone V, Jenatton R, Seeger M et al (2018) Scalable hyperparameter transfer learning. In: Annual conference on neural information processing systems, pp 6845–6855

323. Pham H, Guan M, Zoph B et al (2018) Efﬁcient neural architecture search via parameter sharing. In: International conference on machine learning, pp 6522–6531

324. Pimentel B, de Carvalho A (2019) A new data characterization for selecting clustering algorithms using meta-learning. Inf Sci 477:203–219. <https://doi.org/10.1016/j.ins.2018.10.043>

325. Pimentel B, De Carvalho A (2019) Unsupervised meta-learning for clustering algorithm recommen- dation. In: International joint conference on neural networks. [https://doi.org/10.1109/IJCNN.2019.](https://doi.org/10.1109/IJCNN.2019.8851989) [8851989](https://doi.org/10.1109/IJCNN.2019.8851989)

326. Pinto F, Soares C, Mendes-Moreira J (2016) Towards automatic generation of metafeatures. In: Paciﬁc- Asia conference on advance in conference on knowledge discovery and data mining, pp 215–226.[https://](https://doi.org/10.1007/978-3-319-31753-3_18) [doi.org/10.1007/978-3-319-31753-3\_ 18](https://doi.org/10.1007/978-3-319-31753-3_18)

327. Pourrajabi M, Zimek A, Moulavi D et al (2014) Model selection for semi-supervised clustering. In: International conference on extending database technology, pp 331–342. [https://doi.org/10.5441/002/](https://doi.org/10.5441/002/edbt.2014.31) [edbt.2014.31](https://doi.org/10.5441/002/edbt.2014.31)

328. Prellberg J, Kramer O (2018) Lamarckian evolution of convolutional neural networks. In: International conference on parallel problem solving from nature, pp 424–435. [https://doi.org/10.1007/978-3-319-](https://doi.org/10.1007/978-3-319-99259-4_34) [99259-4\_34](https://doi.org/10.1007/978-3-319-99259-4_34)

329. Probst P, Boulesteix A,Bischl B (2019) Tunability: importance of hyperparameters of machine learning algorithms. J Mach Learn Res 20:1–32

330. Qiao J, Li F, Han H etal (2017) Growing echo-state network with multiple subreservoirs. IEEE Trans Neural Netw Learn Syst 28:391–404. <https://doi.org/10.1109/TNNLS.2016.2514275>

331. Quemy A (2020) Two-stage optimization for machine learning workﬂow. Inf Syst. [https://doi.org/10.](https://doi.org/10.1016/j.is.2019.101483) [1016/j.is.2019.101483](https://doi.org/10.1016/j.is.2019.101483)

332. Rakhshani H,Ismail Fawaz H, Idoumghar L et al (2020) Neural architecture search for time series clas- siﬁcation. In: International joint conference on neural networks. [https://doi.org/10.1109/IJCNN48605.](https://doi.org/10.1109/IJCNN48605.2020.9206721) [2020.9206721](https://doi.org/10.1109/IJCNN48605.2020.9206721)

333. Rakotoarison H, Schoenauer M, Sebag M (2019) Automated machine learning with Monte-Carlo tree search. In: International joint conference on artiﬁcial intelligence, pp 3296–3303. [https://doi.org/10.](https://doi.org/10.24963/ijcai.2019/457) [24963/ijcai.2019/457](https://doi.org/10.24963/ijcai.2019/457)

334. Rana S, Li C, Gupta S et al (2017) High dimensional Bayesian optimization with elastic Gaussian process. In: International conference on machine learning, pp 4407–4415

335. Rasley J, He Y, Yan F et al (2017) HyperDrive: exploring hyperparameters with POP scheduling. In: ACM/IFIP/USENIX middleware Conference, pp 1–13. <https://doi.org/10.1145/3135974.3135994>

336. Rawat W, Wang Z (2019) Hybrid stochastic GA-Bayesian search for deep convolutional neural network model selection. J Univ Comput Sci 25:647–666. <https://doi.org/10.3217/jucs-025-06-0647>

337. Real E, Moore S, Selle A et al (2017) Large-scale evolution of image classiﬁers. In: International conference on machine learning, pp 4429–4446

1 灬

338. Reif M, Shafait F, Goldstein M et al (2014) Automatic classiﬁer selection for non-experts. Pattern Anal Appl 17:83–96. <https://doi.org/10.1007/s10044-012-0280-z>

339. van Rijn J,Hutter F (2018) Hyperparameter importance across datasets. In: ACM SIGKDD international conference on knowledge discovery and data mining, pp 2367–2376. [https://doi.org/10.1145/3219819.](https://doi.org/10.1145/3219819.3220058) [3220058](https://doi.org/10.1145/3219819.3220058)

340. van Rijn J, Abdulrahman S, Brazdil P et al (2015) Fast algorithm selection using learning curves. In: International symposium on intelligent data analysis, pp 298–309. [https://doi.org/10.1007/978-3-319-](https://doi.org/10.1007/978-3-319-24465-5_26) [24465-5\_26](https://doi.org/10.1007/978-3-319-24465-5_26)

341. Ristoski P (2015) Towards linked open data enabled data mining: strategies for feature generation, propositionalization, selection, and consolidation. In: European Semantic Web Conference, pp 772–782.

<https://doi.org/10.1007/978-3-319-18818-8_50>

342. Rosales-Pérez A, Gonzalez J, Coello Coello C et al (2014) Multi-objective model type selection. Neurocomputing 146:83–94. <https://doi.org/10.1016/j.neucom.2014.05.077>

343. Rosales-Pérez A, Gonzalez J, Coello Coello C et al (2015) Surrogate-assisted multi-objective model selectionfor support vectormachines. Neurocomputing150:163–172.[https://doi.org/10.1016/j.neucom.](https://doi.org/10.1016/j.neucom.2014.08.075) [2014.08.075](https://doi.org/10.1016/j.neucom.2014.08.075)

344. Rossi A, de Carvalho A, Soares C et al (2014) MetaStream: a meta-learning based method for peri- odic algorithm selection in time-changing data. Neurocomputing 127:52–64. [https://doi.org/10.1016/j.](https://doi.org/10.1016/j.neucom.2013.05.048) [neucom.2013.05.048](https://doi.org/10.1016/j.neucom.2013.05.048)

345. de Sá A, Freitas A, Pappa G (2018) Automated selection and conﬁguration of multi-label classiﬁcation algorithms with grammar-based genetic programming. In: International conference on parallel problem solving from nature, pp 308–320. <https://doi.org/10.1007/978-3-319-99259-4_25>

346. Sabharwal A, Samulowitz H, Tesauro G (2016) Selecting near-optimal learners via incremental data allocation. In: AAAI conferences on artiﬁcial intelligence, pp 2007–2015. [https://doi.org/10.1609/aaai.](https://doi.org/10.1609/aaai.v30i1.10316) [v30i1.10316](https://doi.org/10.1609/aaai.v30i1.10316)

347. Sanders S, Giraud-Carrier C (2017) Informing the use of hyperparameter optimization through met- alearning. In: IEEE international conference on data mining, pp 1051–1056. [https://doi.org/10.1109/](https://doi.org/10.1109/ICDM.2017.137) [ICDM.2017.137](https://doi.org/10.1109/ICDM.2017.137)

348. Sapra D, Pimentel A (2021) Designing convolutional neural networks with constrained evolutionary piecemeal training. Appl Intell. <https://doi.org/10.1007/s10489-021-02679-7>

349. Saxena S, Verbeek J (2016) Convolutional neural fabrics. In: Annual conference on neural networks information processing system, pp 4060–4068

350. Schilling N, WistubaM, DrumondLetal(2015)Hyperparameteroptimization withfactorizedmultilayer perceptrons. In: European conference in machine learning and knowledge discovery in databases, pp 87–103. <https://doi.org/10.1007/978-3-319-23525-7_6>

351. Schmidt M, Safarani S, Gastinger J et al (2019) On the performance of differential evolution for hyper- parameter tuning. In: International joint conference on neural networks.[https://doi.org/10.1109/IJCNN.](https://doi.org/10.1109/IJCNN.2019.8851978) [2019.8851978](https://doi.org/10.1109/IJCNN.2019.8851978)

352. Schmidt M, Gastinger J, Nicolas S et al (2020) HAMLET—a learning curve-enabled multi-armed bandit for algorithm selection. In: International joint conference on neural networks. [https://doi.org/10.1109/](https://doi.org/10.1109/IJCNN48605.2020.9207233) [IJCNN48605.2020.9207233](https://doi.org/10.1109/IJCNN48605.2020.9207233)

353. Sen R, Kandasamy K, Shakkottai S (2018) Multi-ﬁdelity black-box optimization with hierarchical partitions. In: International conference on machine learning, pp 7212–7224

354. Shah A, Ghahramani Z (2015) Parallel predictive entropy search for batch global optimization of expensive objective functions. In: Annual conference on neural information processing systems, pp 3330–3338

355. ShahriariB, Bouchard-Côté A, de Freitas N (2016) Unbounded bayesian optimization via regularization. In: International conference on artiﬁcial intelligence and statistics, pp 1168–1176

356. Shang Z, Zgraggen E, Buratti B et al (2019) Democratizing data science through interactive curation of ML pipelines. In: International conference on management of data, pp 1171–1188. [https://doi.org/10.](https://doi.org/10.1145/3299869.3319863) [1145/3299869.3319863](https://doi.org/10.1145/3299869.3319863)

357. da Silva A, de Oliveira W, Ludermir T (2016) Weightless neural network parameters and architecture selection in a quantum computer. Neurocomputing 183:13–22. [https://doi.org/10.1016/j.neucom.2015.](https://doi.org/10.1016/j.neucom.2015.05.139) [05.139](https://doi.org/10.1016/j.neucom.2015.05.139)

358. Singh L, Paul S (2021) Hybrid evolutionary network architecture search for convolution class of deep neural networks with applications. Expert Syst. <https://doi.org/10.1111/exsy.12690>

359. Singh P, Chaudhury S, Panigrahi B (2021) Hybrid MPSO-CNN: multi-level particle swarm opti- mized hyperparameters of convolutional neural network. Swarm Evol Comput.[https://doi.org/10.1016/](https://doi.org/10.1016/j.swevo.2021.100863) [j.swevo.2021.100863](https://doi.org/10.1016/j.swevo.2021.100863)

360. SinhaN, Chen KW (2021) Evolving neural architecture using one shot model. In: Genetic and evolvable computing conference, pp 910–918. <https://doi.org/10.1145/3449639.3459275>

361. Smith M, Sala C, Kanter J et al (2020) The machine learning bazaar: harnessing the ML ecosystem for effective system development. In: ACM SIGMOD international conference on management of data, pp 785–800. <https://doi.org/10.1145/3318464.3386146>

362. Smithson S, Yang G, Gross W et al (2016) Neural networks designing neural networks: multi-objective hyper-parameter optimization. In: IEEE/ACM international conference on computer design.[https://doi.](https://doi.org/10.1145/2966986.2967058) [org/10.1145/2966986.2967058](https://doi.org/10.1145/2966986.2967058)

363. Snoek J, Swersky K, Zemel R etal (2014) Input warping for Bayesian optimization of non-stationary functions. In: International conference on machine learning, pp 3654–3662

364. Snoek J, Ripped O, Swersky K et al (2015) Scalable Bayesian optimization using deep neural networks. In: International conference on machine learning, pp 2161–2170

365. Song H (2021) A method for gradient differentiable network architecture search by selecting and clustering candidate operations. Appl Sci (Switzerland). <https://doi.org/10.3390/app112311436>

366. Soniya PaulS, Singh L (2020) Application and need-based architecture design of deep neural networks. Int J Pattern Recognit Artif Intell. <https://doi.org/10.1142/S021800142052014X>

367. Springenberg J, Klein A, Falkner S et al (2016) Bayesian optimization with Robust Bayesian neural networks. In: Annual conference on neural information processing systems, pp 4141–4149

368. Stamoulis D, Ding R, Wang D et al (2020) Single-path NAS: designing hardware-efﬁcient ConvNets in less than 4 hours. In: European conference in machine learning and knowledge discovery in databases, pp 481–497. <https://doi.org/10.1007/978-3-030-46147-8_29>

369. Stein B, Wang H, Back T (2019) Automatic conﬁguration of deep neural networks with parallel efﬁ- cient global optimization. In: International joint conference on neural network. [https://doi.org/10.1109/](https://doi.org/10.1109/IJCNN.2019.8851720) [IJCNN.2019.8851720](https://doi.org/10.1109/IJCNN.2019.8851720)

370. Stojanovic B, Milivojevic M, Milivojevic N et al (2016) A self-tuning system for dam behavior mod- eling based on evolving artiﬁcial neural networks. AdvEng Softw 97:85–95. [https://doi.org/10.1016/j.](https://doi.org/10.1016/j.advengsoft.2016.02.010) [advengsoft.2016.02.010](https://doi.org/10.1016/j.advengsoft.2016.02.010)

371. SuganumaM, Shirakawa S,NagaoT(2017)A geneticprogrammingapproachtodesigning convolutional neural network architectures. In: Genetic and evolvable computing conference, pp 497–504. [https://doi.](https://doi.org/10.1145/3071178.3071229) [org/10.1145/3071178.3071229](https://doi.org/10.1145/3071178.3071229)

372. Suganuma M, Kobayashi M, Shirakawa S et al (2020) Evolution of deep convolutional neural networks using cartesian genetic programming. Evol Comput 28:141–163.<https://doi.org/10.1162/evco_a_00253>

373. Sui G, Yu Y(2020)Bayesiancontextualbanditsforhyperparameteroptimization. IEEEAccess8:42971–

42979. <https://doi.org/10.1109/ACCESS.2020.2977129>

374. Sun M, Dou H, Yan J (2020a) Efﬁcient transfer learning via joint adaptation of network architecture and weight. In: European conference on computer vision, pp 463–480. [https://doi.org/10.1007/978-3-030-](https://doi.org/10.1007/978-3-030-58601-0_28) [58601-0\_28](https://doi.org/10.1007/978-3-030-58601-0_28)

375. Sun X, Lin J, Bischl B (2020b) ReinBo: machine learning pipeline conditional hierarchy search and conﬁguration with bayesian optimization embedded reinforcement learning. In: Joint European confer-

ence on machine learning and principle and problem of knowledge discovery in databases, pp 68–84.

<https://doi.org/10.1007/978-3-030-43823-4_7>

376. Sun Y, Gong H, Li Y et al (2019) Hyperparameter importance analysis based on N-rrelieff algorithm. Int J Comput Commun Control 14:557–573. <https://doi.org/10.15837/ijccc.2019.4.3593>

377. SunY, Xue B, Zhang Metal (2019) A particle swarm optimization-based ﬂexible convolutional autoen- coder for imageclassiﬁcation. IEEE Trans Neural Netw Learn Syst 30:2295–2309. [https://doi.org/10.](https://doi.org/10.1109/TNNLS.2018.2881143) [1109/TNNLS.2018.2881143](https://doi.org/10.1109/TNNLS.2018.2881143)

378. Sun Y, Xue B, Zhang M et al (2020) Completely automated CNN architecture design based on blocks. IEEE Trans Neural Netw Learn Syst 31:1242–1254. <https://doi.org/10.1109/TNNLS.2019.2919608>

379. Sun Y, Xue B, Zhang M et al (2020) Automatically designing CNN architectures using the genetic algorithm for imageclassiﬁcation. IEEE Trans Cybern 50:3840–3854. [https://doi.org/10.1109/TCYB.](https://doi.org/10.1109/TCYB.2020.2983860) [2020.2983860](https://doi.org/10.1109/TCYB.2020.2983860)

380. SunZ, HuY, YangLetal(2021)STC-NAS:fastneuralarchitecturesearchwith source-target consistency. Neurocomputing. <https://doi.org/10.1016/j.neucom.2021.11.082>

381. Szwarcman D, Civitarese D, Vellasco M (2019) Quantum-inspired neural architecture search. In: International joint conference on neural network. <https://doi.org/10.1109/IJCNN.2019.8852453>

382. Tan H, Cheng R, Huang S et al (2021) RelativeNAS: relative neural architecture search via slow–fast learning. IEEE Trans Neural Netw Learn Syst. <https://doi.org/10.1109/TNNLS.2021.3096658>

383. Tang Y, Wang Y, Xu Y et al (2020) A semi-supervised assessor of neural architectures. In: IEEE/CVF conference on computer vision and pattern recognition, pp 1807–1816. [https://doi.org/10.1109/](https://doi.org/10.1109/CVPR42600.2020.00188) [CVPR42600.2020.00188](https://doi.org/10.1109/CVPR42600.2020.00188)

1 灬

384. Termritthikun C, Jamtsho Y, Ieamsaard J et al (2021) EEEA-Net: an early exit evolutionary neural architecture search. Eng Appl Artif Intell. <https://doi.org/10.1016/j.engappai.2021.104397>

385. Theckel Joy T, Rana S, Gupta S et al (2019) A ﬂexible transfer learning framework for Bayesian optimization with convergence guarantee. Expert Syst Appl 115:656–672. [https://doi.org/10.1016/j.](https://doi.org/10.1016/j.eswa.2018.08.023) [eswa.2018.08.023](https://doi.org/10.1016/j.eswa.2018.08.023)

386. Thiede L, Parlitz U (2019) Gradient based hyperparameter optimization in echo state networks. Neural Netw 115:23–29. <https://doi.org/10.1016/j.neunet.2019.02.001>

387. Tian Y, Liu C, Xie L et al (2021) Discretization-aware architecture search. Pattern Recognit. [https://doi.](https://doi.org/10.1016/j.patcog.2021.108186) [org/10.1016/j.patcog.2021.108186](https://doi.org/10.1016/j.patcog.2021.108186)

388. Tian Y, Shen L, Shen L et al (2021) AlphaGAN: fully differentiable architecture search for genera- tive adversarial networks. IEEE Trans Pattern Anal Mach Intell. [https://doi.org/10.1109/TPAMI.2021.](https://doi.org/10.1109/TPAMI.2021.3099829) [3099829](https://doi.org/10.1109/TPAMI.2021.3099829)

389. Tiddi I, D’Aquin M, Motta E (2014) Dedalo: looking for clusters explanations in a labyrinth of linked data. In: International conference in the semantic web: trends and challenges, pp 333–348. [https://doi.](https://doi.org/10.1007/978-3-319-07443-6_23) [org/10.1007/978-3-319-07443-6\_23](https://doi.org/10.1007/978-3-319-07443-6_23)

390. Tran B, Xue B, Zhang M (2016) Genetic programming for feature construction and selection in classiﬁ- cation on high-dimensional data. Memetic Comput 8:3–15. [https://doi.org/10.1007/s12293-015-0173-](https://doi.org/10.1007/s12293-015-0173-y) [y](https://doi.org/10.1007/s12293-015-0173-y)

391. Tran LT, Ali M, Bae SH (2021) A feature fusion based indicator for training-free neural architecture search. IEEE Access 9:133914–133923. <https://doi.org/10.1109/ACCESS.2021.3115911>

392. Tran N, Schneider JG, Weber I et al (2020) Hyper-parameter optimization in classiﬁcation: to-do or not-to-do. Pattern Recognit. <https://doi.org/10.1016/j.patcog.2020.107245>

393. TuK, MaJ, Cui Petal (2019) Autone: hyperparameter optimization for massive network embedding. In: ACM SIGKDD international conference on knowledge discovery and data mining, pp 216–225.[https://](https://doi.org/10.1145/3292500.3330848) [doi.org/10.1145/3292500.3330848](https://doi.org/10.1145/3292500.3330848)

394. Turner J, Crowley E, O’Boyle M (2021) Neural architecture search as program transformation explo- ration. In: International conference on arch support for programming languages and operating systems, pp 915–927. <https://doi.org/10.1145/3445814.3446753>

395. UkilA,SahuI,PuriCetal(2018)AutoModeling:integratedapproachfor automatedmodel generationby ensembleselection of feature subset and classiﬁer. In: International joint conference on neural network.

<https://doi.org/10.1109/IJCNN.2018.8489730>

396. Vahdat A, Mallya A, Liu MYet al (2020) UNAS: differentiable architecture search meets reinforcement learning. In:IEEE/CVF conference on computervisionandpatternrecognition,pp11263–11272.[https://](https://doi.org/10.1109/CVPR42600.2020.01128) [doi.org/10.1109/CVPR42600.2020.01128](https://doi.org/10.1109/CVPR42600.2020.01128)

397. Vainshtein R, Greenstein-Messica A, Katz G et al (2018) A hybrid approach for automatic model recommendation. In: ACM international conference on information and knowledge management, pp 1623–1626. <https://doi.org/10.1145/3269206.3269299>

398. Van Craenendonck T, Blockeel H (2017) Constraint-based clustering selection. Mach Learn 106:1497–

1521. <https://doi.org/10.1007/s10994-017-5643-7>

399. Veniat T, Denoyer L (2018) Learning time/memory-efﬁcient deep architectures with budgeted super networks. In: IEEE/CVF conference on computer vision and pattern recognition, pp 3492–3500.[https://](https://doi.org/10.1109/CVPR.2018.00368) [doi.org/10.1109/CVPR.2018.00368](https://doi.org/10.1109/CVPR.2018.00368)

400. Vidnerová P, Neruda R (2020) Multi-objective evolution for deep neural network architecture search. In: International conference on neural information processing, pp 270–281.[https://doi.org/10.1007/978-3-](https://doi.org/10.1007/978-3-030-63836-8_23) [030-63836-8\_23](https://doi.org/10.1007/978-3-030-63836-8_23)

401. Vijaya Saradhi V, Girish K (2014) Effective parameter tuning of SVMs using radius/margin bound through data envelopment analysis. Neural Process Lett 41:125–138. [https://doi.org/10.1007/s11063-](https://doi.org/10.1007/s11063-014-9338-9) [014-9338-9](https://doi.org/10.1007/s11063-014-9338-9)

402. Völcker C, Molina A, Neumann J et al (2020) DeepNotebooks: deep probabilistic models construct python notebooks for reporting datasets. In: Joint European conference on machine learning and prin- ciples and practice of knowledge discovery in databases, pp 28–43.[https://doi.org/10.1007/978-3-030-](https://doi.org/10.1007/978-3-030-43823-4_3) [43823-4\_3](https://doi.org/10.1007/978-3-030-43823-4_3)

403. Wan A, Dai X, Zhang P et al (2020) FBNetV2: differentiable neural architecture search for spatial and channel dimensions. In: IEEE/CVF conference on computer vision and pattern recognition, pp 12962–12971. <https://doi.org/10.1109/CVPR42600.2020.01298>

404. Wang B, Sun Y, Xue B et al (2019a) Evolving deep neural networks by multi-objective particle swarm optimization for image classiﬁcation. In: Genetic and evolvable computing conference, pp 490–498.

<https://doi.org/10.1145/3321707.3321735>

405. Wang B, Xue B, Zhang M (2021) Surrogate-assisted particle swarm optimization for evolving variable- length transferable blocks for imageclassiﬁcation. IEEE Trans Neural Netw Learn Syst.[https://doi.org/](https://doi.org/10.1109/TNNLS.2021.3054400) [10.1109/TNNLS.2021.3054400](https://doi.org/10.1109/TNNLS.2021.3054400)

406. Wang C,Wang H, MuT et al (2020a) Auto-model: utilizing research papers and HPO techniques to deal with the CASH problem. In: IEEE international conference on data engineering, pp 1906–1909. [https://](https://doi.org/10.1109/ICDE48307.2020.00200) [doi.org/10.1109/ICDE48307.2020.00200](https://doi.org/10.1109/ICDE48307.2020.00200)

407. Wang C, Wang H, Zhou C et al (2021) Experience thinking: constrained hyperparameter optimization based on knowledge and pruning. Knowl Syst. <https://doi.org/10.1016/j.knosys.2020.106602>

408. Wang D, Andres J, Weisz J (2021c) Autods: towards human-centered automation of data science. In: Conference on human factors in computing systems. <https://doi.org/10.1145/3411764.3445526>

409. Wang D, Li M, Gong C et al (2021d) AttentiveNAS: improving neural architecture search via attentive sampling. In: IEEE/CVF conference on computer vision and pattern recognition, pp 6414–6423.[https://](https://doi.org/10.1109/CVPR46437.2021.00635) [doi.org/10.1109/CVPR46437.2021.00635](https://doi.org/10.1109/CVPR46437.2021.00635)

410. Wang E, Xu S, Chen CM et al (2021) Neural-architecture-search-based multiobjective cognitive automation system. IEEE Syst J 15(2):2918–2925. <https://doi.org/10.1109/JSYST.2020.3002428>

411. Wang G, Song Q, Zhang X et al (2014) A generic multilabel learning-based classiﬁcation algorithm recommendation method. ACM Trans Knowl Discov Data. <https://doi.org/10.1145/2629474>

412. Wang G, Ben Sassi M, Grosu R (2017) ZIZO: a novel zoom-in-zoom-out search algorithm for the global parameters of echo-state networks. Can J Electr Comput Eng 40:210–216. [https://doi.org/10.](https://doi.org/10.1109/CJECE.2017.2703093) [1109/CJECE.2017.2703093](https://doi.org/10.1109/CJECE.2017.2703093)

413. Wang H, Yang R, Huang D et al (2021) iDARTS: improving DARTS by node normalization and decorrelationdiscretization. IEEE Trans Neural Netw Learn Syst.[https://doi.org/10.1109/TNNLS.2021.](https://doi.org/10.1109/TNNLS.2021.3105698) [3105698](https://doi.org/10.1109/TNNLS.2021.3105698)

414. Wang L, Feng M, Zhou B et al (2015) Efﬁcient hyper-parameter optimization for NLP applications. In: Conference on empirical methods in natural language processing, pp 2112–2117. [https://doi.org/10.](https://doi.org/10.18653/v1/d15-1253) [18653/v1/d15-1253](https://doi.org/10.18653/v1/d15-1253)

415. Wang L, Xie S, Li T et al (2021) Sample-efﬁcient neural architecture search by learning actions for Monte Carlo tree search. IEEE Trans Pattern Anal Mach Intell. [https://doi.org/10.1109/TPAMI.2021.](https://doi.org/10.1109/TPAMI.2021.3071343) [3071343](https://doi.org/10.1109/TPAMI.2021.3071343)

416. Wang Q,Ming Y, Jin Z etal (2019b) AtmSeer: increasing transparency and controllability in automated machine learning. In: CHI conference on human factors in computer system. [https://doi.org/10.1145/](https://doi.org/10.1145/3290605.3300911) [3290605.3300911](https://doi.org/10.1145/3290605.3300911)

417. Wang X, Xue C, Yan J et al (2020b) MergeNAS: merge operations into one for differentiable architecture search. In: International joint conferences on artiﬁcial intelligence, pp 3065–3072. [https://doi.org/10.](https://doi.org/10.24963/ijcai.2020/424) [24963/ijcai.2020/424](https://doi.org/10.24963/ijcai.2020/424)

418. Wang Z,JegelkaS(2017)Max-value entropy searchfor efﬁcient Bayesianoptimization. In:International conference on machine learning, pp 5530–5543

419. Wang Z, Hutter F, Zoghi M et al (2016) Bayesian optimization in a billion dimensions via random embeddings. J Artif Intell Res 55:361–367. <https://doi.org/10.1613/jair.4806>

420. Wang Z, Zhou B, Jegelka S (2016b) Optimization as estimation with Gaussian processes in bandit settings. In: International conference on artiﬁcial intelligence and statistics, pp 1022–1031

421. Wang Z, Lu D, Wang H et al (2021) Evolutionary convolutional neural network optimization with cross-tasks transfer strategy. Electronics (Switzerland). <https://doi.org/10.3390/electronics10151857>

422. Weidele D, Weisz J, Oduor E et al (2020) AutoAIViz: opening the blackbox of automated artiﬁcial intelligence with conditional parallel coordinates. In: ACM international conference on intelligent user interfaces, pp 308–312. <https://doi.org/10.1145/3377325.3377538>

423. Welchowski T, Schmid M (2016) A framework for parameter estimation and model selection in kernel deep stacking networks. Artif Intell Med 70:31–40. <https://doi.org/10.1016/j.artmed.2016.04.002>

424. Wen W, Liu H, Chen Y et al (2020a) Neural predictor for neural architecture search. In: European conference on computer vision, pp 660–676. <https://doi.org/10.1007/978-3-030-58526-6_39>

425. WenW, Yan F, Chen Y, etal(2020b)AutoGrow:automaticlayer growingindeep convolutionalnetworks.

In: ACM SIGKDD international conference on knowledge discovery and data mining, pp 833–841.

<https://doi.org/10.1145/3394486.3403126>

426. Wen YW, Peng SH, Ting CK (2021) Two-stage evolutionary neural architecture search for transfer learning. IEEE Trans Evol Comput 25(5):928–940. <https://doi.org/10.1109/TEVC.2021.3097937>

427. Weng Y, Zhou T, Liu L et al (2019) Automatic convolutional neural architecture search for image classiﬁcation under different scenes. IEEE Access 7:38495–38506. [https://doi.org/10.1109/ACCESS.](https://doi.org/10.1109/ACCESS.2019.2906369) [2019.2906369](https://doi.org/10.1109/ACCESS.2019.2906369)

1 灬

428. Werneck R, de Almeida W, Stein B et al (2018) Kuaa: a uniﬁed framework for design, deployment, execution, and recommendation of machine learning experiments. Futur Gener Comput Syst 78:59–76.

<https://doi.org/10.1016/j.future.2017.06.013>

429. Wistuba M (2019) Deep learning architecture search by neuro-cell-based evolution with function- preserving mutations. In: European conference on machine learning and principles and practice of knowledge discovery in databases, pp 243–258. [https://doi.org/10.1007/978-3-030-10928-8\_ 15](https://doi.org/10.1007/978-3-030-10928-8_15)

430. Wistuba M (2021) XferNAS: transfer neural architecture search. In: European conference in machine learningand knowledgediscovery indatabases, pp247–262.[https://doi.org/10.1007/978-3-030-67664-](https://doi.org/10.1007/978-3-030-67664-3_15) [3\_15](https://doi.org/10.1007/978-3-030-67664-3_15)

431. WistubaM,Schilling N, Schmidt-ThiemeL (2015) Hyperparameter search space pruning—a new com- ponent for sequential model-based hyperparameter optimization. In: European conference on machine learning and principles and practice of knowledge discovery in databases, pp 104–119. [https://doi.org/](https://doi.org/10.1007/978-3-319-23525-7_7) [10.1007/978-3-319-23525-7\_7](https://doi.org/10.1007/978-3-319-23525-7_7)

432. Wistuba M, Schilling N, Schmidt-Thieme L (2016) Sequential model-free hyperparameter tuning. In: IEEE international conference on data mining, pp 1033–1038. <https://doi.org/10.1109/ICDM.2015.20>

433. WistubaM,Schilling N, Schmidt-ThiemeL(2017)Automatic frankensteining:creating complex ensem- bles autonomously. In: SIAM international conference on data mining, pp 741–749. [https://doi.org/10.](https://doi.org/10.1137/1.9781611974973.83) [1137/1.9781611974973.83](https://doi.org/10.1137/1.9781611974973.83)

434. WistubaM,Schilling N, Schmidt-ThiemeL (2018) Scalable Gaussian process-based transfer surrogates for hyperparameter optimization. Mach Learn 107:43–78. <https://doi.org/10.1007/s10994-017-5684-y>

435. Wu B, Dai X, Zhang P et al (2019) FBNet: hardware-aware efﬁcient ConvNet design via differentiable neural architecture search. In: IEEE/CVF conference on computer vision and pattern recognition, pp 10726–10734. <https://doi.org/10.1109/CVPR.2019.01099>

436. Wu J, Frazier P (2016) The parallel knowledge gradient method for batch Bayesian optimization. In: Annual conference on neural information processing systems, pp 3134–3142

437. Wu J, Chen S, Liu X (2020) Efﬁcient hyperparameter optimization through model-based reinforcement learning. Neurocomputing 409:381–393. <https://doi.org/10.1016/j.neucom.2020.06.064>

438. Xie H, Zhang L, Lim C (2020) Evolving CNN-LSTM models for time series prediction using enhanced grey wolf optimizer. IEEE Access 8:161519–161541. <https://doi.org/10.1109/ACCESS.2020.3021527>

439. Xie L, Yuille A (2017) Genetic CNN. In: IEEE international conference on computer vision, pp 1388–

1397. <https://doi.org/10.1109/ICCV.2017.154>

440. Xu Y, Wang Y, Han K et al (2021a) ReNAS: relativistic evaluation of neural architecture search. In: IEEE/CVF conference on computer vision and pattern recognition, pp 4409–4418. [https://doi.org/10.](https://doi.org/10.1109/CVPR46437.2021.00439) [1109/CVPR46437.2021.00439](https://doi.org/10.1109/CVPR46437.2021.00439)

441. Xu Y, XieL, Dai W et al (2021) Partially-connected neural architecture search for reduced computational redundancy. IEEE Trans Pattern Anal Mach Intell 43(9):2953–2970. [https://doi.org/10.1109/TPAMI.](https://doi.org/10.1109/TPAMI.2021.3059510) [2021.3059510](https://doi.org/10.1109/TPAMI.2021.3059510)

442. Xue C, Yan J, Yan R et al (2019) Transferable AutoML by model sharing over grouped datasets. In: IEEE/CVF conference on computer vision and pattern recognition, pp 8994–9003. [https://doi.org/10.](https://doi.org/10.1109/CVPR.2019.00921) [1109/CVPR.2019.00921](https://doi.org/10.1109/CVPR.2019.00921)

443. Xue S, Chen H, Xie C et al (2021) Fast and unsupervised neural architecture evolution for visual representation learning. IEEE Comput Intell Mag 16(3):22–32. [https://doi.org/10.1109/MCI.2021.](https://doi.org/10.1109/MCI.2021.3084394) [3084394](https://doi.org/10.1109/MCI.2021.3084394)

444. Xue Y, Jiang P, Neri F etal (2021) A multi-objective evolutionary approach based on graph-in-graph for neural architecture search of convolutional neural networks. Int J Neural Syst. [https://doi.org/10.1142/](https://doi.org/10.1142/S0129065721500350) [S0129065721500350](https://doi.org/10.1142/S0129065721500350)

445. Yakovlev A, Moghadam HF, Moharrer A et al (2020) Oracle AutoML: a fast and predictive AutoML pipeline. In: International conference on very large databases, pp 3166–3180.[https://doi.org/10.14778/](https://doi.org/10.14778/3415478.3415542) [3415478.3415542](https://doi.org/10.14778/3415478.3415542)

446. Yan C, Chang X, Li Z et al (2021) ZeroNAS: differentiable generative adversarial networks search for zero-shot learning. IEEE Trans Pattern Anal Mach Intell.<https://doi.org/10.1109/TPAMI.2021.3127346>

447. Yan Z, DaiX, Zhang Petal (2021b) FP-NAS: fast probabilistic neural architecture search. In: IEEE/CVF conference on computer vision and pattern recognition, pp 15134–15143. [https://doi.org/10.1109/](https://doi.org/10.1109/CVPR46437.2021.01489) [CVPR46437.2021.01489](https://doi.org/10.1109/CVPR46437.2021.01489)

448. Yang TJ, Liao YL, Sze V (2021a) NetAdaptV2: efﬁcient neural architecture search with fast super- networktraining andarchitectureoptimization. In:IEEE/CVF conference on computervisionandpattern recognition, pp 2402–2411. <https://doi.org/10.1109/CVPR46437.2021.00243>

449. Yang Y, You S, Li H et al (2021b) Towards improving the consistency, efﬁciency, and ﬂexibility of differentiable neural architecture search. In: IEEE/CVF conference on computer vision and pattern recognition, pp 6663–6672. <https://doi.org/10.1109/CVPR46437.2021.00660>

450. Yang Z, Zhang A (2021) Hyperparameter optimization via sequential uniform designs. J Mach Learn Res 22:6592–6638

451. Yang Z, Wang Y, Chen X et al (2020) CARS: continuous evolution for efﬁcient neural architecture search. In: IEEE/CVF conference on computer vision and pattern recognition, pp 1826–1835. [https://](https://doi.org/10.1109/CVPR42600.2020.00190) [doi.org/10.1109/CVPR42600.2020.00190](https://doi.org/10.1109/CVPR42600.2020.00190)

452. Yang Z, Wang Y, Chen X et al (2021c) HourNAS: extremely fast neural architecture search through an hourglass lens. In: IEEE/CVF conference on computer vision and pattern recognition, pp 10891–10901.

<https://doi.org/10.1109/CVPR46437.2021.01075>

453. Yao Y, Cao J, Ma Z (2018) A cost-effective deadline-constrained scheduling strategy for a hyperpa- rameter optimization workﬂow for machine learning algorithms. In: International conference on system computing, pp 870–878. <https://doi.org/10.1007/978-3-030-03596-9_62>

454. Ye Q, Sun Y, Zhang J et al (2021) A distributed framework for EA-based NAS. IEEE Trans Parallel Distrib Syst 32(7):1753–1764. <https://doi.org/10.1109/TPDS.2020.3046774>

455. Yogatama D, Mann G (2014) Efﬁcient transfer learning method for automatic hyperparameter tuning. In: International conference on artiﬁcial intelligence and statistics, pp 1077–1085

456. You S, Huang T, Yang M et al (2020) Greedynas: towards fast one-shot NAS with greedy supernet. In: IEEE/CVF conference on computer vision and pattern recognition, pp 1996–2005. [https://doi.org/10.](https://doi.org/10.1109/CVPR42600.2020.00207) [1109/CVPR42600.2020.00207](https://doi.org/10.1109/CVPR42600.2020.00207)

457. Yu Y, Qian H, Hu YQ (2016) Derivative-free optimization via classiﬁcation. In: AAAI conference on artiﬁcial intelligence, pp 2286–2292

458. Zainel Q, KhorsheedM, Darwish Set al (2021) An optimized convolutional neural network architecture based on evolutionary ensemble learning. Comput Mater Contin 69(3):3813–3828. [https://doi.org/10.](https://doi.org/10.32604/cmc.2021.014759) [32604/cmc.2021.014759](https://doi.org/10.32604/cmc.2021.014759)

459. Zhang B, Zhou G (2021) Control the number of skip-connects to improve robustness of the NAS algorithm. IET Comput Vis 15(5):356–365. <https://doi.org/10.1049/cvi2.12036>

460. Zhang H, Jin Y, Cheng R et al (2021) Efﬁcient evolutionary search of attention convolutional networks via sampled training and node inheritance. IEEE Trans Evol Comput 25(2):371–385.[https://doi.org/10.](https://doi.org/10.1109/TEVC.2020.3040272) [1109/TEVC.2020.3040272](https://doi.org/10.1109/TEVC.2020.3040272)

461. Zhang J, Fogelman-Soulié F, Largeron C (2018a) Towards automatic complex feature engineering. In: International conference on web information system engineering, pp 312–322. [https://doi.org/10.1007/](https://doi.org/10.1007/978-3-030-02925-8_22) [978-3-030-02925-8\_22](https://doi.org/10.1007/978-3-030-02925-8_22)

462. Zhang M, Li H, Pan S et al (2020) One-shot neural architecture search via novelty driven sampling. In: International joint conference on artiﬁcial intelligence, pp 3188–3194. [https://doi.org/10.24963/ijcai.](https://doi.org/10.24963/ijcai.2020/441) [2020/441](https://doi.org/10.24963/ijcai.2020/441)

463. Zhang M, LiH, Pan Set al (2021)One-shot neural architecture search: maximising diversityto overcome catastrophic forgetting. IEEE Trans Pattern Anal Mach Intell 43(9):2921–2935.[https://doi.org/10.1109/](https://doi.org/10.1109/TPAMI.2020.3035351) [TPAMI.2020.3035351](https://doi.org/10.1109/TPAMI.2020.3035351)

464. Zhang T, Georgiopoulos M, Anagnostopoulos G (2016) Multi-objective model selection via racing. IEEE Trans Cybern 46:1863–1876. <https://doi.org/10.1109/TCYB.2015.2456187>

465. Zhang T, Lei C, Zhang Z et al (2021) AS-NAS: adaptive scalable neural architecture search with rein- forced evolutionary algorithm for deep learning. IEEE Trans Evol Comput 25(5):830–841. [https://doi.](https://doi.org/10.1109/TEVC.2021.3061466) [org/10.1109/TEVC.2021.3061466](https://doi.org/10.1109/TEVC.2021.3061466)

466. Zhang T, Waqas M, Shen H et al (2021) A neural network architecture optimizer based on DARTS and generative adversarial learning. Inf Sci 581:448–468. <https://doi.org/10.1016/j.ins.2021.09.041>

467. Zhang X,Chen X, Yao Let al(2019a)Deep neuralnetworkhyperparameteroptimization withorthogonal array tuning. In: International conference on neural information processing, pp 287—-295. [https://doi.](https://doi.org/10.1007/978-3-030-36808-1_31) [org/10.1007/978-3-030-36808-1\_31](https://doi.org/10.1007/978-3-030-36808-1_31)

468. Zhang X, Hou P, Zhang X et al (2021e) Neural architecture search with random labels. In: IEEE/CVF conference on computer vision and pattern recognition, pp 10902–10911. [https://doi.org/10.1109/](https://doi.org/10.1109/CVPR46437.2021.01076) [CVPR46437.2021.01076](https://doi.org/10.1109/CVPR46437.2021.01076)

469. Zhang X, Huang Z, Wang N et al (2021) You only search once: single shot neural architecture search via direct sparse optimization. IEEE Trans Pattern Anal Mach Intell 43(9):2891–2904. [https://doi.org/](https://doi.org/10.1109/TPAMI.2020.3020300) [10.1109/TPAMI.2020.3020300](https://doi.org/10.1109/TPAMI.2020.3020300)

470. Zhang Y,DaiZ, Low B(2019b)Bayesian optimization with binaryauxiliary information. In: Conference on uncertainity in artiﬁcial intelligence

471. ZhangZ,Han J,Qian Ket al(2018b)Evolvinglearningfor analysingmood-relatedinfant vocalisation. In: Annual international speech communication, pp 142–146. [https://doi.org/10.21437/Interspeech.2018-](https://doi.org/10.21437/Interspeech.2018-1914) [1914](https://doi.org/10.21437/Interspeech.2018-1914)

1 灬

472. Zhao Z, Kang Y, Hou A et al (2021) SP-DARTS: synchronous progressive differentiable neural archi- tecturesearch for image classiﬁcation. IEICE Trans Inf Syst E104D(8):1232–1238. [https://doi.org/10.](https://doi.org/10.1587/transinf.2020BDP0009) [1587/transinf.2020BDP0009](https://doi.org/10.1587/transinf.2020BDP0009)

473. Zheng X, Zhang Y, Hong S et al (2021) Evolving fully automated machine learning via life-long knowl- edge anchors. IEEE Trans Pattern Anal Mach Intell 43(9):3091–3107. [https://doi.org/10.1109/TPAMI.](https://doi.org/10.1109/TPAMI.2021.3069250) [2021.3069250](https://doi.org/10.1109/TPAMI.2021.3069250)

474. Zhong G, Jiao W, Gao W et al (2020) Automatic design of deep networks with neural blocks. Cogn Comput. <https://doi.org/10.1007/s12559-019-09677-5>

475. Zhong S,Xie X, LinL et al (2017) Genetic algorithm optimized double-reservoir echo state network for multi-regime time series prediction. Neurocomputing 238:191–204. [https://doi.org/10.1016/j.neucom.](https://doi.org/10.1016/j.neucom.2017.01.053) [2017.01.053](https://doi.org/10.1016/j.neucom.2017.01.053)

476. Zhong Z, Yan J, Wu W et al (2018) Practical block-wise neural network architecture generation. In: IEEE/CVF conference on computer vision and pattern recognition, pp 2423–2432. [https://doi.org/10.](https://doi.org/10.1109/CVPR.2018.00257) [1109/CVPR.2018.00257](https://doi.org/10.1109/CVPR.2018.00257)

477. Zhong Z, Yang Z, Deng B et al (2021) BlockQNN: efﬁcient block-wise neural network architecture generation. IEEE Trans Pattern Anal Mach Intell 43(7):2314–2328. [https://doi.org/10.1109/TPAMI.](https://doi.org/10.1109/TPAMI.2020.2969193) [2020.2969193](https://doi.org/10.1109/TPAMI.2020.2969193)

478. Zhou D, Zhou X, Zhang W et al (2020a) EcoNAS: ﬁnding proxies for economical neural architecture search. In: IEEE/CVF conference on computer vision and pattern recognition, pp 11393–11401.[https://](https://doi.org/10.1109/CVPR42600.2020.01141) [doi.org/10.1109/CVPR42600.2020.01141](https://doi.org/10.1109/CVPR42600.2020.01141)

479. Zhou Y, Jin Y, Ding J (2020) Surrogate-assisted evolutionary search of spiking neural architectures in liquid state machines. Neurocomputing 406:12–23. <https://doi.org/10.1016/j.neucom.2020.04.079>

480. Zhu B, Al-Ars Z, Hofstee H (2020) NASB: neural architecture search for binary convolutional neural networks. In: International joint conference on neural network. [https://doi.org/10.1109/IJCNN48605.](https://doi.org/10.1109/IJCNN48605.2020.9207674) [2020.9207674](https://doi.org/10.1109/IJCNN48605.2020.9207674)

481. Zhuo L, Zhang B, Chen Hetal (2020) CP-NAS: child–parent neural architecture search for 1-bit CNNs. In: International joint conference on artiﬁcial intelligence, pp 1033–1039.[https://doi.org/10.24963/ijcai.](https://doi.org/10.24963/ijcai.2020/144) [2020/144](https://doi.org/10.24963/ijcai.2020/144)

482. Zimmer L, Lindauer M, Hutter F (2021) Auto-Pytorch: multi-ﬁdelity MetaLearning for efﬁcient and robust AutoDL. IEEE Trans Pattern Anal MachIntell43(9):3079–3090.[https://doi.org/10.1109/TPAMI.](https://doi.org/10.1109/TPAMI.2021.3067763) [2021.3067763](https://doi.org/10.1109/TPAMI.2021.3067763)

483. Zoph B, Vasudevan V, Shlens J et al (2018) Learning transferable architectures for scalable image recognition. In: IEEE/CVF conference on computer vision and pattern recognition, pp 8697–8710.

<https://doi.org/10.1109/CVPR.2018.00907>

484. Zutty J, Long D, Adams H et al (2015) Multiple objective vector-based genetic programming using human-derived primitives. In: Genetic and evolvable computing conference, pp 1127–1134. [https://doi.](https://doi.org/10.1145/2739480.2754694) [org/10.1145/2739480.2754694](https://doi.org/10.1145/2739480.2754694)

**Publisher’s Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional afﬁliations.

**Rafael Barbudo** is a Ph.D. Student of Computer Science at the Uni- versity of Córdoba, Spain. He received both his B.S. and M.S. in Information and Computer Science at the University of Córdoba. His current research is focused on the area of automated machine learn- ing (AutoML). Speciﬁcally, it is dedicated to the development of tools that assist both domain experts and data scientists in the process of knowledge extraction by optimizing and automating those repetitive and time-consuming phases.

**Sebastián Ventura** is a Full Professor in the Department of Computer Science and Numerical Analysis at the University of Córdoba, where he leads the Knowledge Discovery and Intelligent Systems Research Laboratory. He has published three books and about 300 papers in journals and scientiﬁc conferences and edited about ten books and mul- tiple special issues in international journals in his area of expertise. He currently holds different positions at the editorial board of multiple top- ranked journals and serves as Editor in Chief at the Progress in Artiﬁ- cial Intelligence journal. His main research interests are in the ﬁelds of data science, computational intelligence, and their applications.

**José Raúl Romero** is an Associate Professor of Computer Science at the University of Córdoba, Spain. He received his PhD from the University of Málaga (Spain). He has published more than 110 papers in journals and scientiﬁc conferences and edited one volume in his area of exper- tise. His research interests focus on democratization of data science, particularly on the development of new methods for the automation of machine learning, human-centered machine learning, and explica- bility of artiﬁcial intelligence. He is an active researcher on artiﬁcial intelligence for software engineering.

1 灬