Journal of Data-centric Machine Learning Research (2026) Submitted 10/25; Revised 04/26; Published 07/2026

AutoML to advance Earth Observation

Julia Wąsala j.wasala@liacs.leidenuniv.nl
Leiden Institute for Advanced Computer Science (LIACS)
Leiden University
Leiden, the Netherlands

Joannes D. Maasakkers
SRON Space Research Organisation Netherlands
Leiden, the Netherlands

Ilse Aben
SRON Space Research Organisation Netherlands
Leiden, the Netherlands

Rochelle Schneider
Φ-lab Explore Office
ESA/ESRIN
Frascati, Italy

Holger Hoos
Chair for Artificial Intelligence Methodology (AIM)
RWTH Aachen
Aachen, Germany

Mitra Baratchi
Leiden Institute for Advanced Computer Science (LIACS)
Leiden University
Leiden, the Netherlands

Reviewed on OpenReview: *https://openreview.net/forum?id=j9gQSilGq2*

Editor: Matthias Feurer

Abstract

Machine learning (ML) techniques have become crucial enablers of progress in Earth observation (EO) research by providing data-driven solutions to effectively leverage the large volumes of available satellite imagery. However, configuring ML pipelines demands substantial time and expertise, making it challenging to consistently develop robust, high-performing ML pipelines for EO applications. Automated machine learning (AutoML), the study of automatic configuration and design of methods and tools for ML systems, has emerged as a solution to this problem. Key AutoML techniques, such as neural architecture search (NAS) and hyperparameter optimisation (HPO), are increasingly applied to EO-related problems. Developing AutoML for processing EO data is challenging, as automation removes human interventions and domain knowledge that can ensure the development of robust Machine Learning for Earth observation (ML4EO) models for new applications. While both AutoML and ML for EO (ML4EO) techniques have been widely studied and reviewed independently, no existing survey provides a systematic overview of their intersection, including the domain-specific challenges that arise when applying AutoML in an EO context. To address this gap, we present a survey of AutoML4EO through both data-centric and model-centric perspectives, covering the application of existing AutoML systems to EO problems and the design of entirely new systems. Finally, we highlight specific characteristics of EO data that are relevant to the development and evaluation of AutoML4EO systems and provide a perspective on challenges that must be addressed to create the next generation of AutoML4EO systems.

Keywords: AutoML, Neural Architecture Search, Satellite images, Hyperparameter optimisation, ML4EO, Earth observation, AI for Science

1 Introduction

The growing amount of Earth observation (EO) data gathered by satellites and other remote sensing technologies necessitates the use of machine learning (ML) to create data-driven models to automatically process this data. ML models complement established modelling approaches, such as physics-based models. ML models can address various subtasks in regression or classification, from predicting missing variables in regions or periods with incomplete data (e.g., cloud inpainting, Arp et al., 2024) to predicting phenomena without prior assumptions about the underlying physics (e.g., anomalous ship detection, Kurchaba et al., 2023; creating land cover maps through pixel-level classification, Li et al., 2021; applying regression to predict landslide susceptibility, Bruzón et al., 2021; or crop yield, Darra et al., 2023; and more). However, developing performant ML models requires significant time and expertise in ML algorithms, evaluation and interpretation of results. Furthermore, ML for EO (ML4EO) researchers invest considerable efforts in designing ML pipelines for processing raw EO data that often contains noise and other imperfections. These barriers impede progress on impactful EO tasks such as greenhouse gas monitoring and disaster forecasting. To address these challenges, automated machine learning (AutoML, Hutter et al., 2019; Alcobaça and de Carvalho, 2025; Kim et al., 2022; Zöller and Huber, 2021; Salehin et al., 2024; Baratchi et al., 2024), a research area focusing on the automatic design and configuration of ML pipelines, offers a promising approach in solving EO problems, as it simplifies and accelerates the creation of high-performing ML pipelines. Here, we review the use of AutoML for Earth observation, surveying nearly 100 works on the topic.

Designing an ML pipeline often involves making design choices regarding components (e.g., pre-processing or model class) and hyperparameters. For instance, we need to choose the model type (e.g., a neural network or random forest) and set its hyperparameters (e.g., the learning rate of a neural network or the number of features in a split node of a decision tree). Each new application requires a new pipeline configuration, since no single model can *a priori* be assumed to perform best for all possible datasets (Wolpert and Macready, 1997), and the choice of hyperparameters significantly impacts performance on different datasets (Hutter et al., 2014). Even ML experts often cannot anticipate the performance of a configuration, due to the dependencies between hyperparameters and the task or data itself. As a result, manual model design is a time-consuming, nonlinear process primarily guided by human expertise. It is often also difficult to later explain how a set of hyperparameters or configurations was chosen, making the process difficult to reproduce. AutoML aims to address the problems of difficulty, extensive time requirements, and lack of reproducibility in ML model design and configuration by studying principled, automated procedures for designing and configuring ML models. AutoML systems find performance-optimised models

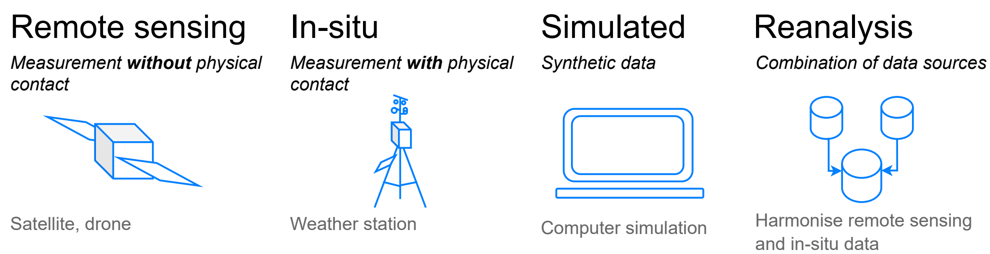

Figure 1: There are three main types of EO data: remote sensing, *in situ*, and simulated data. Additionally, reanalysis products are obtained by updating physics models to produce data that is more similar to observational data (including remote sensing and in situ).

and hyperparameters for a given pipeline using efficient search algorithms, reducing the need for ML expertise.

When applying ML for EO, the final goal is not necessarily to configure the model with the best performance in terms of metrics such as accuracy, but rather to understand a specific domain problem (Reichstein et al., 2019), reflecting the AI for Science paradigm that emphasises scientific insight over predictive performance. At the core of this issue are the unique properties of the data and how we can model them with ML. There are three categories of EO data (Figure 1): (i) remote sensing, (ii) *in situ*, and (iii) simulated data (Karpatne et al., 2019; Salcedo-Sanz et al., 2020). Remote sensing data is obtained by measuring radiation emitted or reflected by an object from a distance, for instance, with sensors onboard drones, aircraft, satellites, or ground-based measuring stations. *in situ* data is collected from weather balloons, planes, ground-based measuring stations, sensors such as anemometers measuring wind speed and direction, or data collection campaigns for concentrations of greenhouse gases. These campaigns require collecting samples on location (e.g., in mangroves) and often processing samples in laboratory settings. Simulated data, often produced by physical models, are used to improve our understanding of variables and processes driving observations (e.g., emissions). In ML4EO, simulated data is also used when remote sensing or *in situ* data is hard or impossible to obtain, such as problems related to atmospheric parameters (e.g., emulating atmospheric radiative transfer models, Vicent Servera et al., 2023). These three data types are sometimes combined to create harmonised data products such as reanalysis data (e.g., the Copernicus Atmospheric Monitoring Service Reanalysis data, Inness et al., 2015), which is created by running historical observations (both remote sensing and *in situ*) through a physics model to fill gaps and create a consistent resolution.

Automating the design of ML4EO models through AutoML presents new challenges and concerns by removing human interventions and domain knowledge that are essential in developing pipelines for new applications that generalise well to deployment conditions, including class imbalance, bias, and noise. Earth systems and EO data possess unique characteristics that require the development of specialised models, often guided by domain knowledge. This expertise is particularly crucial when designing training and evaluation datasets, which are often not available for new tasks. Domain experts can identify subgroups in the data prone to misclassification and highlight essential features for specific applications. Given the central role of data quality and characteristics, a significant portion of the challenges in AutoML4EO, such as feature selection and engineering, can be defined as data-centric ML problems (Oala et al., 2024; Zha et al., 2025). Furthermore, ML4EO applications often involve multi-faceted problem statements that require careful integration of different ML techniques rather than using single out-of-the-box solutions. As a result, many ML4EO algorithms (e.g., Zhang et al., 2022a,d; Zhang and Ma, 2024; Kurchaba et al., 2023; Wąsala et al., 2025) target highly specific problems and datasets, as no generally applicable method has emerged that works across diverse data types and applications. This issue calls for greater attention to the development of specialised AutoML systems for ML4EO.

A key reason generic AutoML systems are insufficient for EO is that they rest on two assumptions that frequently fail in practice:

1. The first is that the training signal reliably guides optimisation. In EO, labels are scarce; may be biased (see, e.g., McGovern et al., 2022, 2024); are potentially spatiotemporally autocorrelated, which can lead to overly optimistic performance estimates if not addressed (see, e.g., Koldasbayeva et al., 2024); and may contain errors or uncertainty in the labels as a result of inter-labeller disagreement or the presence of difficult classes, leading to label noise (see, e.g., Hechinger et al. (2025)). Together, these properties lead to an **optimisation gap**: the difficulty of reliably identifying well-performing architectures or configurations in the given search space, and sometimes also at a lower level, the difficulty of learning the intended relationships rather than spurious correlations that happen to hold within the labelled data (see, e.g., Xiong et al., 2022; Wąsala et al., 2026).

2. The second assumption is that optimising a metric on available labelled data is a reliable proxy for real-world performance. In EO, this **evaluation or generalisation gap** is widened by the fact that models are routinely applied to new geographic regions, seasons, sensors, or time periods not represented in the training data. Class imbalance, distribution shift between training and deployment conditions, and the difficulty of capturing generalisation in a single metric can lead even well-optimised models to fail to perform as expected in deployment.

Addressing both gaps simultaneously remains an open challenge; we discuss strategies for mitigating these issues in Section 6.

In this paper, we provide an overview of AutoML systems for EO that covers relevant challenges and opportunities at the intersection of AutoML and ML4EO. Many surveys have covered relevant ML and EO topics, such as Geospatial Artificial Intelligence (GeoAI) (Mai et al., 2025; Koldasbayeva et al., 2024; Gao et al., 2023; Rolnick et al., 2022), ML4EO deep learning (Reichstein et al., 2019; Hoeser and Kuenzer, 2020; Hoeser et al., 2020), datasets (Han et al., 2023; Affek and Szymański, 2024), data fusion (Salcedo-Sanz et al., 2020; Ghamisi et al., 2019), data-centric ML4EO (Roscher et al., 2024), responsible ML4EO (Ghamisi et al., 2024; McGovern et al., 2022, 2024), and explainability (Gevaert, 2022; Höhl et al., 2024). In parallel, AutoML has been extensively covered in various surveys (see, e.g., Hutter et al., 2019; Alcobaça and de Carvalho, 2025; Kim et al., 2022; Zöller and Huber, 2021; Salehin et al., 2024; Baratchi et al., 2024), and following the success of deep learning

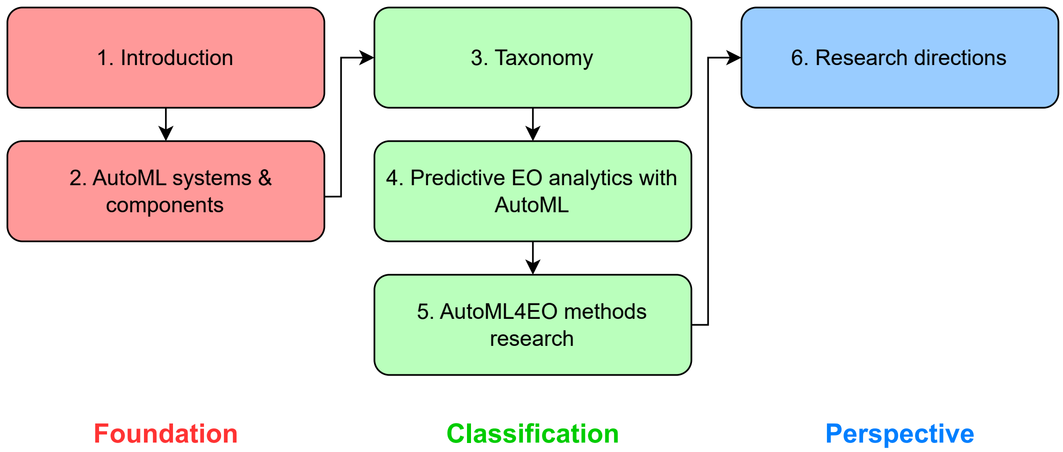

Figure 2: The sections of this survey are arranged chronologically: the background covers the foundations of AutoML4EO; the heart of the survey covers our classification of AutoML4EO research; the final sections provide our perspectives on AutoML4EO.

approaches in computer vision, AutoML researchers have studied the automatic design of neural networks for dense image tasks (Yusof et al., 2024) and methods for reducing the high computational complexity of AutoML for deep learning (Heuillet et al., 2024). However, existing surveys do not comprehensively study AutoML for EO, focusing on either ML4EO or AutoML, with limited exploration of AutoML applications to spatial data by Wen and Li (2022). We aim to address this gap by presenting a thorough overview of AutoML4EO, providing sufficient background for ML researchers in the EO domain and EO scientists using (Auto)ML to develop next-generation AutoML systems or to effectively select and calibrate existing systems. Our contributions include the following:

- We provide the necessary background on AutoML and identify critical challenges affecting the design of AutoML4EO systems.

- We present a taxonomy of AutoML4EO approaches, summarising how AutoML systems are applied to EO-related problems and categorising the most critical themes in their design. We further provide information to help select AutoML4EO systems for specific applications.

- We discuss open challenges in AutoML4EO and present future research directions.

The remainder of this article is structured as follows (see Fig. 2). The components and types of AutoML systems are covered in Section 2. Our taxonomy of AutoML4EO is presented in Section 3. We then consider the application of conventional AutoML systems to EO problems (Section 4), followed by a categorisation of AutoML4EO methods research (Section 5). Finally, in Section 6, we discuss four research directions for creating next-generation AutoML4EO systems.

2 AutoML systems and components

In the following, we give a brief overview of AutoML, covering the background required for understanding the remainder of this survey. Interested readers are also referred to previously published, more comprehensive overviews of AutoML (see, e.g., Hutter et al., 2019; Alcobaça and de Carvalho, 2025; Bischl et al., 2023; Talbi, 2021; Baratchi et al., 2024; Salmani Pour Avval et al., 2025; Baymurzina et al., 2022; White et al., 2023; Zöller and Huber, 2021; Kim et al., 2022; Salehin et al., 2024). We describe the main components of AutoML systems and introduce three types of AutoML systems. Finally, we briefly discuss automated data preparation.

2.1 AutoML components

AutoML systems consist of three components (Figure 3): (i) a search space $\mathcal{S}$ of all relevant design choices, where each element $s \in \mathcal{S}$ represents a candidate pipeline or configuration, (ii) a search strategy for traversing $\mathcal{S}$, and (iii) a performance evaluation strategy that assigns a performance measure $\mathcal{L}(s, \mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}})$ to any $s \in \mathcal{S}$, given a dataset $\mathcal{D}$ split into training and validation sets $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{val}}$. In general, s may encode an algorithm choice $A^{(j)} \in \mathcal{A}$ and/or a hyperparameter configuration $\lambda \in \Lambda^{(j)}$, such that $\mathcal{S} \subseteq \mathcal{A} \times \Lambda$, where $\Lambda = \bigcup_j \Lambda^{(j)}$. These components enable the automatic creation of a machine learning pipeline for a given dataset.

2.1.1 Search space

The search space is defined by the set of design choices considered in the context of constructing or configuring a machine learning pipeline, such as possible preprocessors, machine learning models, neural network architectures, and values of hyperparameters that control specific aspects of these. The search space contains all possible combinations (or configurations) of these design choices, possibly subject to feasibility constraints. The search space is the most critical component to specify or adjust when adapting AutoML systems to new application domains or types of data, because it determines the type of ML pipelines that can be obtained.

A key design tradeoff exists between expressiveness and efficiency: a broader search space allows more configurations to be explored and can yield novel and potentially better solutions, but increases the computational cost and makes optimisation harder, making it more likely to get stuck in a local minimum. Conversely, incorporating prior knowledge narrows the search space, reducing the computational cost required to search the space, but potentially excluding better solutions. Most search spaces are heavily inspired by state-of-the-art manually designed pipelines or architectures in the application domain, leaning towards narrower or more biased search spaces (White et al., 2023).

2.1.2 Search strategy

The search strategy determines how an AutoML system selects candidate configurations from the search space for evaluation. AutoML search spaces defined based on state-of-the-art machine learning models can become extremely large and contain many poor-performing configurations. As a result, sophisticated search strategies are necessary to efficiently identify high-performance configurations. Several search strategies are employed by existing AutoML systems (see, e.g., Alcobaça and de Carvalho, 2025; Hutter et al., 2019; Zöller and Huber, 2021; Baratchi et al., 2024; Salehin et al., 2024). **Random search** samples configurations uniformly from the search space (Bergstra and Bengio, 2012). This search strategy is a good option for large, high-dimensional search spaces with cheap evaluations. Furthermore, random search is a strong baseline against more sophisticated approaches. **Bayesian optimisation** builds a surrogate model to predict the performance of previously unseen configurations and uses an acquisition function to select promising configurations for evaluation (Falkner et al., 2018; Feurer et al., 2022; Baratchi et al., 2024; Alcobaça and de Carvalho, 2025; Shahriari et al., 2016; Wang et al., 2023c; Garnett, 2023). It is most effective when evaluations are expensive, as the surrogate model focuses evaluations on promising regions; its advantage over random search may diminish in very high-dimensional spaces, where the surrogate becomes harder to fit reliably (Shahriari et al., 2016). **Reinforcement learning approaches** learn a policy to select configurations based on the observed performance evaluations, often by training an additional neural network to learn this policy (Zoph et al., 2018). **Evolutionary algorithms** maintain populations of configurations and create or sample new ones using evolutionary operations, such as crossover and mutation (Stanley et al., 2019; Olson et al., 2016). These approaches are well-suited to large, complex, or structured search spaces where gradient information is unavailable, but at the cost of high computational complexity. **Gradient-based methods**, primarily used in neural architecture search (NAS), apply continuous relaxation techniques to transform discrete architectural hyperparameters into differentiable variables that can be optimised via gradient descent along with the network weights (Liu et al., 2018). This approach was introduced in DARTS (Liu et al., 2018), upon which many subsequent NAS systems were built (e.g., Chu et al., 2020; Xu et al., 2020), similar to how ResNet (He et al., 2016) has driven many developments in computer vision.

The computational complexity of a search strategy is influenced by three factors:

1. The first factor is search efficiency: how many evaluations are needed to find high-quality configurations. For instance, grid search is typically less efficient than random search, as it requires more evaluations to achieve comparable performance. Evolutionary algorithms are also evaluation-intensive, as population-based methods require evaluating many candidates per generation over many generations. The search efficiency can be increased by using a warm-starting strategy (see, e.g., Zimmer et al., 2021), which initialises the search with performance of configurations that worked well on similar datasets, similar to using pre-trained weights in deep learning. Warm-starting is closely related to meta-learning (Vanschoren, 2019), which aims to generalise beyond learning from a single dataset, to learn how different datasets are learned. In the context of HPO, this typically works by maintaining a repository of prior experiments recording which configurations performed well on previous datasets. Similar datasets are identified via meta-features (i.e., simple dataset properties such as the number of samples, class imbalance ratio, or feature statistics, Vanschoren, 2019), and the best-performing configurations from those datasets may be used to initialise the search on the new task.

2. The second factor is sampling cost: the overhead of selecting the next candidate configuration. For grid and random search this cost is negligible. However, for reinforcement learning-based search the cost can be substantial, as it involves both training and inference of a separate neural network. Bayesian optimisation lies in between these two extremes, incurring a moderate sampling cost through fitting and querying a surrogate model, though it can make up for this cost by its sampling efficiency (Falkner et al., 2018). Gradient-based search strategies operate in a different manner: instead of sampling discrete candidates, they optimise architectural hyperparameters continuously alongside network weights, making each evaluation expensive but reducing the total number of evaluations needed. However, this comes at a high memory cost that scales with the size of the search space, and the continuous relaxation can introduce biases that affect the quality of the final architecture (Liu et al., 2018).

3. The third factor is cost per evaluation, which is largely determined by the performance evaluation strategy discussed next.

2.1.3 Performance evaluation strategy

The performance evaluation strategy is a procedure that estimates (or predicts) and compares the performance of candidate configurations; the performance values thus obtained are used by the search strategy to help determine which configurations to sample next. Key choices include the evaluation metric and the training budget allocated per evaluation, both of which affect not only the quality of the performance estimate, but also the computational cost of the search. The per-evaluation cost depends largely on the complexity of components in the search space. For instance, deep neural networks are more expensive to evaluate than decision trees. In EO, this cost may be further amplified by large image sizes and high spectral dimensionality (for instance, hyperspectral images may have hundreds of channels). Several strategies exist to reduce this cost. Early stopping terminates training before convergence if a configuration shows poor promise. More generally, multi-fidelity methods, such as Hyperband (Li et al., 2017b), evaluate candidate configurations on reduced budgets (i.e., fewer training iterations, or smaller data subsets) before assigning more resources to the most promising configurations. At the extreme, zero-cost NAS proxies estimate performance from gradient statistics at initialisation, requiring no training at all (Salmani Pour Avval et al., 2025).

Another common approach is to use transfer learning, where model weights are initialised from a backbone pretrained on a large dataset, such as ImageNet (Deng et al., 2009) or BigEarthNet (Sumbul et al., 2019), rather than training from scratch. Fine-tuning a pretrained model on the target dataset typically requires fewer training steps to converge, directly reducing the cost of each evaluation (see, e.g., Hossen et al., 2025). Foundation models, and geospatial foundation models in EO (see, e.g., Jakubik et al., 2023; Cong et al., 2022; Guo et al., 2024) leverage extensive pre-training, often on a combination of datasets, to create general-purpose models that can be fine-tuned for a variety of tasks. However, the benefit of transfer learning depends on the similarity between pretraining and target domains (Wang et al., 2023a).

These cost-reduction strategies involve a tradeoff; cheaper evaluations give less reliable performance estimates, which risks selecting configurations that do not generalise well to

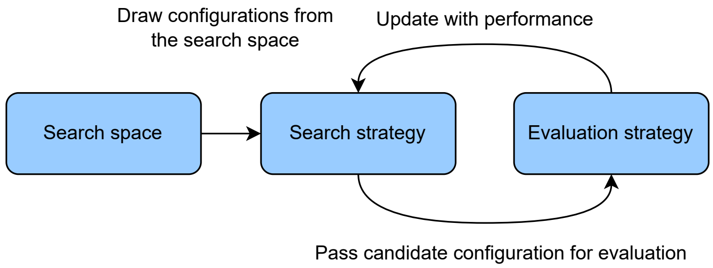

Figure 3: AutoML systems consist of three components that interact to create and configure ML pipelines.

the target task, especially if the configuration is searched on a different dataset than the target data. This tradeoff is particularly relevant in EO, where the gap between proxy performance and real-world generalisation can be large due to data diversity and the frequency of extrapolation to unseen conditions.

2.2 Types of AutoML Systems

AutoML systems can be organised according to three problems they are designed to solve:

- **Algorithm or model selection**: automatically selecting the best model $A^{(j)}$ from a set of candidate algorithms $\mathcal{A} = \{A^{(1)}, \ldots, A^{(N)}\}$, where N is the number of algorithms in $\mathcal{S}$. For instance, choosing between a support vector machine and a random forest, between two network architectures, or even variants of the same algorithm with different hyperparameter settings;

- **Hyperparameter optimisation (HPO)**: automatically determining performance-optimising hyperparameter configuration $\lambda^* \in \Lambda$ for a fixed algorithm A, i.e., $\mathcal{S} = \Lambda$:

$$\lambda^* \in \arg\min_{\lambda \in \Lambda} \mathcal{L}(A_\lambda, \mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}}) \tag{1}$$

- **Combined algorithm selection and hyperparameter optimisation (CASH)** (Thornton et al., 2013): simultaneously selecting the best model or algorithm and hyperparameter settings, i.e., $\mathcal{S} = \bigcup_j \{A^{(j)}\} \times \Lambda^{(j)}$. Formally, let $\mathcal{A} = \{A^{(1)}, \ldots, A^{(N)}\}$ be a set of algorithms, where each algorithm $A^{(j)}$ has an associated hyperparameter space $\Lambda^{(j)}$. Given a dataset $\mathcal{D}$, a validation procedure that yields one or more train/validation splits $\{\mathcal{D}^{(i)}_{\text{train}}, \mathcal{D}^{(i)}_{\text{val}}\}_{i=1}^{K}$ (e.g., a single holdout split, $K = 1$, or K-fold cross-validation, as in Thornton et al., 2013), and a loss function $\mathcal{L}$, the CASH problem is to find the algorithm and hyperparameter configuration that minimise the average validation loss:

$$A^{(j^*)}_{\lambda^*} \in \arg\min_{A^{(j)} \in \mathcal{A},\, \lambda \in \Lambda^{(j)}} \frac{1}{K} \sum_{i=1}^{K} \mathcal{L}\left(A^{(j)}_{\lambda}, \mathcal{D}^{(i)}_{\text{train}}, \mathcal{D}^{(i)}_{\text{val}}\right) \tag{2}$$

Broad-spectrum methods tend to address the CASH problem. NAS can be formulated as an HPO problem, as the search space does not tend to include multiple distinct algorithms, as is the case with broad-spectrum frameworks, but instead consists of architectural and, sometimes, training hyperparameters. NAS search strategies can be generic HPO algorithms, or specific ones designed to optimise neural network architectures. HPO is not just used within NAS, but is pervasive in ML, as it has become standard practice to optimise ML pipelines with HPO (Bischl et al., 2023; Thornton et al., 2013; Olson et al., 2016). Consequently, it is not always explicitly described when studies have used HPO to tune training hyperparameters, making it difficult to systematically cover HPO for EO in those cases. Here, we focus on broad-spectrum frameworks and NAS, including HPO algorithms employed as search strategies, in AutoML4EO. We discuss each type of system and examples of their use for AutoML4EO.

2.2.1 Broad-spectrum frameworks

Broad-spectrum frameworks are AutoML systems that focus on optimising classical ML models and their hyperparameters by addressing the CASH problem. These frameworks are convenient to use and require limited ML expertise; the user often only needs to choose the target performance metric. The search space, search strategy, and evaluation strategy are often pre-determined, but users can usually exclude pipeline steps (such as pre-processing), model classes, or hyperparameters from the search space. The search spaces can cover entire ML pipelines, including data pre-processing, feature selection, model selection, and hyperparameter optimisation. Table 1 provides examples of some commonly used AutoML systems in ML4EO research. Most frameworks consider a search space based on algorithms implemented in existing ML packages, such as scikit-learn (Pedregosa et al., 2011). Some of these AutoML systems have sparked follow-up work, such as Auto-sklearn (Feurer et al., 2022), and improve performance by creating ensembles of multiple optimised ML models.

Auto-sklearn (Feurer et al., 2022) uses Bayesian optimisation as a search strategy to optimise ML pipelines and constructs voting ensembles by greedily selecting previously evaluated pipelines to improve the total performance of the ensemble, applying the greedy ensemble selection algorithm as proposed by Caruana et al. (2004). H2O AutoML (LeDell and Poirier, 2020) uses random search as a search strategy and creates stacking ensembles (van der Laan et al., 2007; Wolpert, 1992), which consist of a set of base models evaluated in earlier iterations, and a meta-learner model that learns to combine the predictions from the base models into a final prediction. AutoGluon (Erickson et al., 2020) constructs ensembles of base models (which may be tuned individually with a specific search strategy), using a combination of bagging and stacking ensembling strategies. In contrast to H2O AutoML, AutoGluon stacks multiple layers, each consisting of multiple models, where each layer takes as input the original features in addition to the predictions of the previous layer, combining them to produce a final prediction. TPOT (Olson et al., 2016) does not create ensembles, but uses genetic programming to evolve tree-structured ML pipelines across generations, returning a single pipeline that balances predictive performance and complexity. Finally, FLAML (Wang et al., 2021a) optimises single models without pre-processing or ensembling.

2.2.2 NAS

In contrast to broad-spectrum frameworks that focus on the configuration of classical machine learning algorithms, neural architecture search (NAS) methods (see, e.g., Salmani Pour Avval et al., 2025; Talbi, 2021; Baymurzina et al., 2022; Elsken et al., 2019; White et al., 2023; Mohan et al., 2023) focus on automatically designing a neural network by optimising architectural hyperparameters such as operations, connections, activation functions, kernel sizes, filters, and the number of layers. Training hyperparameters, such as the learning rate, are typically not tuned by NAS approaches, but NAS-designed networks can be tuned using HPO methods after the search is complete. As a result, NAS is highly related to HPO. While NAS can be designed for any data or task, image classification with natural images is the most prevalent task (Mohan et al., 2023).

Search spaces and search strategies. The search space defines the set of architectures a NAS system can explore, and is typically the component that requires the most domain knowledge to design (White et al., 2023); the choice of search space largely determines which search strategies are feasible. We consider three categories of NAS search spaces (see, e.g., Alcobaça and de Carvalho, 2025; Baratchi et al., 2024; Talbi, 2021; White et al., 2023; Elsken et al., 2019).

Macro-level search spaces optimise either the full architecture (e.g., the number of layers and expansion rate patterns) or only high-level design decisions (e.g., how to connect the layers). These search spaces are common in sample-based approaches (i.e., NAS search strategies that train and evaluate multiple networks before choosing one, for instance, Jin et al., 2019; Zimmer et al., 2021; Balaprakash et al., 2018), and are the most expressive type of search space that are most likely to yield novel architectures, making them well-suited to tasks where little is known about the optimal architecture. However, they are also the most computationally expensive and can be difficult to optimise reliably (White et al., 2023). Macro-level search spaces are often paired with sample-based strategies (e.g., Jin et al., 2019; Zimmer et al., 2021), which train and evaluate multiple architectures independently and select the best one. For EO tasks involving relatively new or understudied problems, such as multi-temporal classification tasks or tasks with limited labelled data, macro search may be appropriate when sufficient data and computational resources are available to support the search.

Micro-level or cell-based search spaces optimise a smaller repeated cell structure rather than the entire network structure, substantially reducing the search cost. This makes them well-suited to supernet strategies, which train a single network encompassing all design choices in the search space. Supernet strategies differ in how they navigate the search space (White et al., 2023): non-differentiable approaches (see, e.g., Cai et al., 2020) use an optimisation algorithm to search for the best sub-network either during or after supernet training, whereas differentiable approaches (see, e.g., Liu et al., 2018; Cai et al., 2019) optimise architecture hyperparameters via gradient descent in tandem with training the supernet. The final network is obtained by keeping the subnetwork of operations with the highest weights and discarding the rest. Gradient-based approaches may also over- or underestimate the performance of the final architecture because the supernet's weight-sharing during training does not perfectly reflect how a standalone architecture would perform when trained independently (White et al., 2023). Given the same search space, supernet approaches are less computationally complex than sample-based approaches but require more memory to contain the supernet. Micro-level search is most appropriate when a reasonable high-level architecture is already known and the goal is to refine performance further. This type of search space is therefore well-suited to established EO tasks such as land cover classification on large standard datasets. A limitation is that micro-level search relies on fixed macro-level design choices and may generalise poorly to tasks that differ significantly from those for which the macro structure was designed (Elsken et al., 2019).

Hierarchical search spaces impose a tree-like structure on the search, allowing joint or sequential optimisation of macro and micro design choices. As a result, hierarchical search spaces offer architectural diversity similar to macro search spaces, while reducing the size of the search space through the tree-like structure. However, the optimisation problem increases in complexity, requiring search strategies capable of handling multi-level optimisation problems.

Several practical considerations should guide search space design. First, the amount of labelled data: expressive search spaces require sufficient data for the search strategy to reliably distinguish good configurations from bad ones, which may be a limiting factor in EO applications where labelled data are expensive to obtain. Second, for dense prediction tasks such as semantic segmentation of high resolution imagery, the optimisation gap is larger than in standard image classification, because architectures and images are larger, and typically fewer labelled samples are available (Mohan et al., 2023). In practice, this problem is often addressed by restricting the search to minor design decisions of known architectures, including training hyperparameters and kernel size or number of filters. In summary, for well-studied EO tasks with large datasets, NAS challenges are similar to those in the natural image domain, with the main differences arising at evaluation, which needs to account for spatiotemporal autocorrelation and differences in class distribution between the training data and deployment scenarios. For newer, less well-studied tasks, expressive search spaces (macro and hierarchical) may help discover strong architectures, but are often not feasible due to limited labelled data, making domain-informed NAS design likely to have the greatest impact.

NAS frameworks. End-to-end NAS frameworks allow users to automatically design architectures for dealing with their data by means of a few lines of code specifying the training and validation data, and a target metric to optimise (Table 1). They offer pre-defined search spaces, and allow users to customise the design choices available in the search space, but rarely natively allow users to switch between types of search spaces. Most NAS frameworks are defined using macro search spaces. For instance, AutoKeras (Jin et al., 2019) is a sample-based learning framework built on Keras for regression and classification; it supports tabular and image data, and offers different search strategies. Auto-PyTorch (Zimmer et al., 2021) similarly searches macro search spaces, and, in addition to optimising the training and macro hyperparameters, also uses warm-starting: a fixed set of configurations automatically run at the start of the search to build the surrogate model, which estimates the performance of a given configuration. This set of configurations is obtained by running a multi-fidelity Bayesian optimisation search strategy (Falkner et al., 2018) independently on a set of meta-training benchmark datasets and collecting the best-performing (incumbent) configuration from each run and greedily adding configurations to the set if they lead to performance improvements. This approach is also called meta-learning for model selection.

Though convenient, the customisation possibilities of these approaches are limited, and changing the search space or search strategy beyond the options implemented in the packages can require significant software engineering effort (Wąsala et al., 2024). DeepHyper (Balaprakash et al., 2018) provides a more flexible interface for HPO and NAS in user-defined macro and micro search spaces, as well as multiple search strategies. When choosing a NAS approach, practitioners should consider whether the convenience of a pre-defined, end-to-end framework outweighs the flexibility of implementing a NAS algorithm directly, the latter offering greater control over the search space, search strategy, and integration with existing pipelines.

2.3 Automated data preparation

EO pipelines are inherently multi-step and complex. Before any model can be trained, raw satellite data must typically be integrated across sensors, cleaned of artefacts such as clouds or missing values, and preprocessed into a form suitable for learning, through steps such as normalisation, resampling or spatial cropping. In ML4EO, these steps are rarely trivial and require domain-specific approaches. Mladenovic et al. (2026) organise data preparation tasks according to four functions: integration, cleaning, preprocessing, and feature engineering (the last of which is less relevant in deep learning). In ML4EO, these translate to the common following tasks:

- **Integration**: Combining multiple datasets into a single tabular dataset, reprocessing satellite images from multiple sensors to make them more similar[1];
- **Cleaning**: filling missing values (Wąsala et al., 2026), filtering data based on metrics like quality assurance or cloud cover percentage, masking clouds (Francis et al., 2022), clipping high values that may be errors;
- **Preprocessing**: resampling the classes to mitigate class imbalance (Chen et al., 2025) increasing the resolution with super-resolution approaches, cropping large image tiles to workable sizes, normalising data, designing data augmentations (Hopkins et al., 2025) or augmenting with synthetic data (Castro Lopes et al., 2025).

Broad-spectrum AutoML frameworks (Section 2.2.1) automate several simple steps as part of a unified pipeline, though typically via relatively simple algorithms such as imputation or basic scaling. NAS, in contrast, often assumes that input data has already been preprocessed; data preparation falls outside of the scope of the architecture search itself. The precise steps required depend strongly on the specific dataset and task, and EO-specific challenges such as cloud masking and spatiotemporal partitioning are not straightforwardly automated.

One preparation step that can be naturally framed as an AutoML problem is automated data augmentation. The choice and composition of augmentation strategies (e.g., rotations, flips, or colour transformations) can be treated as a hyperparameter optimisation problem

1. See, e.g., harmonised Landsat and Sentinel-2 data. https://hls.gsfc.nasa.gov/. Last accessed 16 April 2026.

(see, e.g., AutoAugment Cubuk et al., 2019). Hopkins et al. (2025) proposed an EO-specific automated data augmentation approach inspired by TrivialAugment (Müller and Hutter, 2021), showing that tailoring augmentation search to satellite imagery outperformed the standard algorithm on satellite image classification tasks. Beyond augmentation, automating the broader set of EO preparation steps remains largely unexplored and represents a promising direction for future work.

3 Taxonomy of AutoML for EO

ML4EO research is often categorised into EO applications (e.g., landslide detection, ship detection: Gao et al., 2023; Hoeser et al., 2020; Salcedo-Sanz et al., 2020), ML tasks (e.g., object detection, segmentation: Hoeser et al., 2020), or research focused on a specific modality or sensor type of EO data (e.g., optical, radar: Affek and Szymański, 2024; Han et al., 2023; Ghamisi et al., 2019). Categorisation comes with significant complexity. On the one hand, categories that are too narrow make it more challenging to draw methodological insights that generalise across applications, datasets, or tasks. On the other hand, categories that are too broad make it harder to identify specific methodological solutions transferable to a researcher's work from a pool of loosely related work. Furthermore, an ML pipeline developed for operational use in a specific application may have different generalisation requirements than general-purpose methods developed with benchmark datasets. Finally, many of these categorisations do not transfer to AutoML4EO, where methods often need to generalise across applications, datasets, and sometimes even tasks (see, e.g., Wang et al., 2021b).

We propose a taxonomy of AutoML4EO works (Figure 4) that allows researchers and practitioners to gain insight into common challenges and problem-solving patterns in AutoML4EO. We distinguish AutoML4EO works based on three criteria: (i) the research goal (predictive analytics or ML methods research), (ii) the problem-solving strategy (model- or data-centric strategies), and (iii) the problem statement characteristics (groups of methods to address specific AutoML4EO problems). In the remainder of this section, we will elaborate on the individual components of our taxonomy.

3.1 Predictive EO analytics with AutoML *vs* AutoML4EO methods design

EO researchers and domain experts seek to improve predictive EO analysis pipeline results of a precisely defined study area (e.g., a few croplands or an area susceptible to landslides) with off-the-shelf AutoML frameworks and techniques (Section 2.2.1). In such studies, the aim is not to design new techniques, but to realise and quantify performance improvements with AutoML. On the other hand, many existing studies on AutoML4EO focus solely on designing new methods (Section 5). AutoML4EO methods research aims to improve results of larger problem classes, such as "multispectral land cover classification" instead of a single study area. Using standardised benchmarks, these studies develop new algorithms to address challenges not covered by existing AutoML systems, such as multi-image data fusion. AutoML4EO works can fall into both categories; for instance, when an AutoML4EO pipeline is designed for a specific application group, it can lead to new methodological insights.

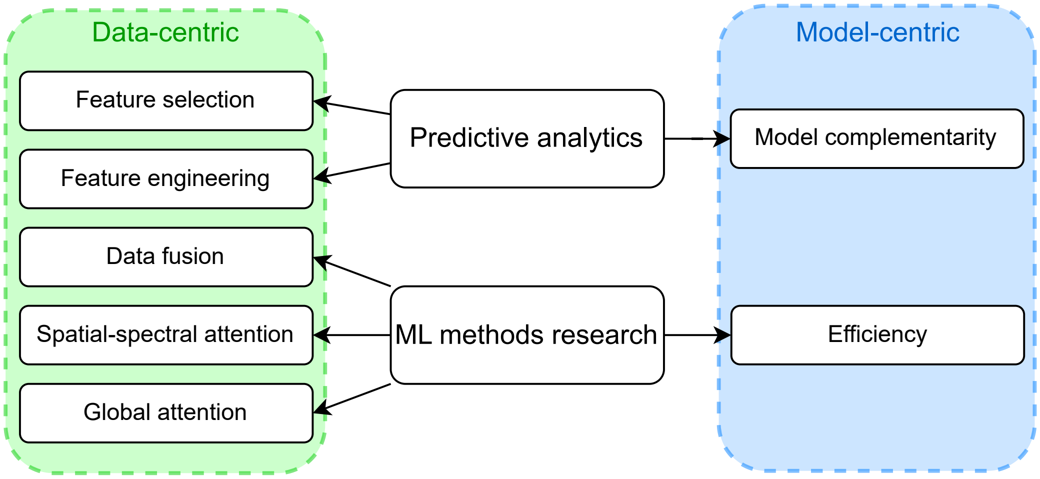

Figure 4: Our taxonomy of AutoML4EO has three levels: the type of research (predictive analytics *vs* ML methods research), the problem-solving strategy (data- *vs* model-centric), and problem statement characteristics (common methodological challenges, such as feature selection or feature engineering).

3.2 Data-centric *vs* model-centric perspective

Data-centric machine learning focuses on addressing dataset-specific properties, while improving performance by creating improved features for a given application (Oala et al., 2024; Zha et al., 2025). Zha et al. (2025) classify data-centric machine learning techniques into three categories: improving the training data, inference or evaluation data, and data maintenance (of training data and operational systems). In deep learning, data-centric techniques go beyond pre-processing, feature engineering, and data partitioning, as neural networks automatically extract features from data, blurring the line between data engineering and model design. Therefore, we expand our definition of data-centric methods to include methods designed to extract better features using domain knowledge about the data. Many studies on predictive analytics with AutoML use domain knowledge about the data to select and craft informative features (Sections 4.1.1 and 4.1.2). Insights about the data can also be leveraged to design new ML techniques, such as spatial-spectral and global attention (Sections 5.1.1 and 5.1.2). Other AutoML4EO challenges are more model-centric, emphasising the development of new ML methods and generally applicable algorithms rather than improving data quality. For instance, improving the computational efficiency of AutoML4EO frameworks regardless of input data is an algorithmic problem (Section 5.2). Furthermore, the model-centric perspective leads to new predictive analysis pipelines through algorithmic advances in combining AutoML frameworks with traditional modelling techniques, such as physics models (Section 4.2). Real-world ML4EO problems sometimes require solutions that combine model- and data-centric approaches. Similarly, AutoML4EO methods can combine components from both perspectives rather than adhering to a single paradigm. In those cases, we categorise these components separately, with some works discussed across multiple subsections to address their relevant contributions.

4 Predictive EO analytics with AutoML

Researchers create data-driven predictive analysis pipelines by applying AutoML frameworks to EO tasks in domains from soil to atmosphere, examples include: crop yield prediction (Darra et al., 2023, 2024, 2025; Kheir et al., 2024; Li et al., 2022b; Ramzan et al., 2023; Lai et al., 2025), landslide susceptibility assessment (Bruzón et al., 2021; Ma et al., 2023; Tang et al., 2023; Renza et al., 2021), earthquake prediction (Yusof et al., 2024), weather forecasting (Maulik et al., 2020; Kiedanski et al., 2025), plant phenotyping (Koh et al., 2021), and many more tasks. We selected published papers that applied AutoML frameworks without significant modifications to their search space, search strategy, or performance evaluation strategy. These approaches focus on developing and improving prediction pipelines for EO tasks. In the remainder of this section, we discuss how AutoML can be used to improve predictive EO analysis pipelines using data-centric (Section 4.1) and model-centric (Section 4.2) techniques.

4.1 AutoML frameworks: tools for data understanding

In ML4EO, training instances are created by transforming EO data into pixels, image patches, or entire tiles to train ML models. Real-world datasets, which are often small, are frequently represented as tabular data, where each row corresponds to a single pixel, while each feature represents a spectral band – specific ranges of wavelengths in the electromagnetic spectrum. The number of bands depends on the sensor used. Multispectral imagery captures data in a few broad bands (e.g., 13 in the case of Sentinel-2). Hyperspectral imagery, on the other hand, captures data in many contiguous bands (e.g., 224 for EnMAP). Typically, the more bands, the narrower each of them is, and the more detailed the spectral information becomes. For classical machine learning-based AutoML frameworks that lack native image support, pixel data must be converted into a tabular format (see, e.g., Babaeian et al., 2021; Darra et al., 2023, 2024; Li et al., 2022b; Kheir et al., 2024). Table 1 lists the most-used frameworks and functionality (a full list showing which publications use which frameworks is presented in Table 4). This table is not intended as a comprehensive or current comparison of AutoML frameworks; rather, it summarises the frameworks as they were used in the EO studies reviewed in this survey; given the pace of development in AutoML, some details will inevitably have changed since publication. The list includes broad-spectrum frameworks, simple AutoML functions, and NAS frameworks. All frameworks support regression, and most support classification, while a few also support time series. We discuss two ways in which AutoML frameworks improve data understanding in EO: by producing insights into feature importances and by constructing informative features.

4.1.1 Feature importance & selection

Feature importance quantifies how much each input feature contributes to the predictions obtained from a given model. Feature importances can serve two purposes: post-hoc analysis, to understand which data elements are most important for modelling a specific phenomenon, and feature selection during development, to remove noisy features and reduce overfitting (Li et al., 2017a).

Framework	Year	Tasks	Im.	FS	PP	MeL	Code	Last update
H2O AutoML (LeDell and Poirier, 2020)	2020	Regression, classification	✗	✗	✓	✗	Link	15-04-2026
Auto-Sklearn (Feurer et al., 2022)	2015	Regression, classification	✗	✗	✓	✓	Link	18-04-2023
AutoKeras (Jin et al., 2019)	2018	Regression, classification	✓	✗	✓	✗	Link	25-11-2025
TPOT (Olson et al., 2016)	2016	Regression, classification	✓	✓	✓	✗	Link	11-09-2025
FLAML (Wang et al., 2021a)	2019	Regression, classification, time series forecasting and classification, ranking	✗	✗	✓	✗	Link	14-04-2026
AutoGluon-Tabular & Timeseries (Erickson et al., 2020; Tang et al., 2024)	2020	Regression, classification, time series	✗	✓	✓	✗	Link	19-04-2026
MATLAB (`fitrauto`)	2020	Regression	✗	✗	✗	✗	N.A.	N.A.
PyCaret (Ali, 2020)	2020	Regression, classification, time series forecasting, anomaly detection, clustering	✗	✓	✓	✗	Link	06-03-2025
MLBox (Axel de Romblay, 2019)	2017	Classification, Regression	✗	✓	✓	✗	Link	25-08-2020
Auto-PyTorch (Zimmer et al., 2021)	2021	Regression, classification, time series forecasting	✗	✗	✓	✓	Link	23-08-2022
DeepHyper (Balaprakash et al., 2018)	2018	Regression, classification, forecasting	✓	✗	✗	✗	Link	12-01-2026

Table 1: AutoML frameworks used for EO applications covered in this survey. The table shows the year the corresponding work was published, if available, and the year the framework was made available. The tasks column shows which tasks are supported. New functionality like support for image data (Im. column) has been added to updated versions of frameworks (e.g., AutoGluon), but those updated frameworks had not yet been applied to EO tasks at the time of writing. Feature selection (FS) is the automated selection of a subset of features to train the ML models. Pre-processing (PP) encompasses filling in missing values, one-hot encoding, and normalisation. Meta-learning (MeL), refers to the option of using ensembles of configurations that worked well on other datasets for model selection. Additionally, the table lists the link to the code repository and the date of the last update of the repository (as of April 2026). H2O AutoML, Auto-sklearn, TPOT, PyCaret, AutoGluon-Tabular, FLAML and DeepHyper are still actively maintained.

H2O AutoML (LeDell and Poirier, 2020), the most-used framework for EO tasks (see, e.g., Adams et al., 2020; Babaeian et al., 2021; Chen et al., 2023; Kheir et al., 2024; Lee et al., 2023; Ma et al., 2023; Mangalath Ravindran et al., 2022; Mubarak and Koeshidayatullah, 2023; Naik et al., 2023; Šandera and Štych, 2024; Sun et al., 2021; Zhang et al., 2024; Bodini, 2024; Ngoc Tu et al., 2025) supports feature importance. Other AutoML systems, such as AutoGluon (Erickson et al., 2020), PyCaret (Ali, 2020), and TPOT (Olson et al., 2016), also support automated feature selection next to feature importance (see Table 5). Though convenient, automated feature selection can harm interpretability by obscuring how or why features are selected. Critically, features important to a subset of the data may be discarded when the impact on overall accuracy is small, potentially causing misclassification of subgroups and introducing model bias that emerges only during subset analysis. To address these interpretability concerns, practitioners often prefer a semi-automated approach.

One such approach involves running AutoML frameworks on different feature sets to identify essential features for specific EO-related problems, a procedure analogous to grid search applied to input data rather than hyperparameters (César de Sá et al., 2022). This allows users to examine performance across different feature combinations and decide which features to retain. For instance, comparing Auto-sklearn performance across different feature sets has been used in various studies as an easy-to-obtain proxy for feature importance (Li et al., 2022b; Darra et al., 2024, 2025; Bodini, 2024; Wang et al., 2024b). However, this approach still does not address the subgroup bias problem, as subgroups cannot be specified in current AutoML frameworks.

Beyond the subgroup performance issue, there is a broader challenge in interpreting feature importances and AutoML-designed pipelines, related to the evaluation gap. The features and pipeline components selected by AutoML frameworks reflect predictive utility for a specific dataset, but not necessarily a physical or causal association. Multiple factors can distort feature importances: small datasets can lead to overfitting, spurious correlations can become learning shortcuts for ML models, and sampling biases can misrepresent the underlying task. Additionally, algorithmic bias can amplify focus on specific features over others, particularly in the presence of class imbalance, which is prevalent in many EO problems, such as those related to anomaly detection (see, e.g., McGovern et al., 2024). Previous work in explainable AI (xAI) demonstrates a human tendency to over-interpret numerical explanations, even when there is a lack of full understanding of what these mean (Ehsan et al., 2024). Therefore, data-driven explanations should serve as exploratory starting points requiring validation through domain knowledge and physical understanding, rather than definitive conclusions about causal structures.

4.1.2 Constructing informative features

Feature engineering is the process of combining or processing raw input features to improve the final ML model. Feature engineering can reduce noise by combining correlated features, for instance, using domain knowledge about a downstream task. Despite the success of end-to-end deep learning approaches, feature engineering remains useful when limited access to labelled data causes deep neural networks to overfit. The feature engineering functions utilised in AutoML frameworks are often based on operations such as principal component analysis (PCA), which obscure the contributions of individual input features (Figure 5).

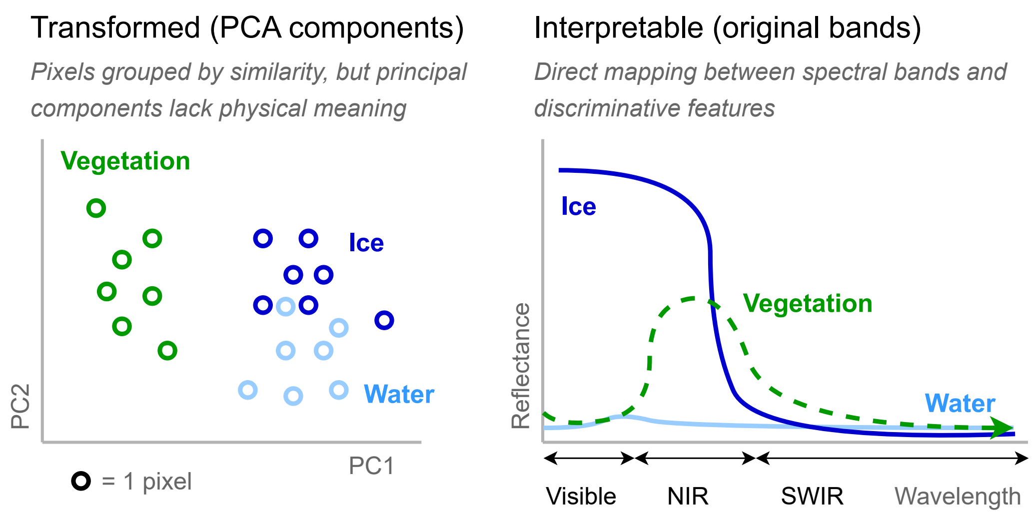

Figure 5: There is a tradeoff between interpretability and performance in feature engineering. Automatic feature extraction algorithms such as principal component analysis (principal components PC1 and PC2 on the **left**) can improve class separation and algorithm performance in tasks such as land cover classification (César de Sá et al., 2022), but obscure which original bands drive class differences. In contrast, spectral signatures (**right**) reveal interpretable patterns that directly link to physical properties: ice shows high reflectance in the visible spectrum, dropping sharply in the near infrared (NIR); vegetation has high NIR but low visible and short-wave infrared (SWIR) reflectance; and water maintains consistently low reflectance across all wavelengths. These interpretable patterns identify which satellite bands are most discriminative, enabling domain insights that automated feature extraction cannot provide.

Many EO practitioners therefore prefer other, more transparent alternatives, such as meaningful combinations of spectral features, such as vegetation indices. Vegetation indices are ratios of bands that help visually monitor variables such as vegetation health in multispectral data and serve as proxies for the joint spectral contributions of bands to a task. Darra et al. (2024) and Lai et al. (2025) further included temporal information by defining features combining vegetation indices from multiple timestamps to improve predictions in two land cover tracking tasks. Das et al. (2025) extracted features from satellite image time series using a recurrent neural network (RNN) with long short-term memory (LSTM) modules to model soil carbon with H2O and PyCaret. While most practitioners use domain knowledge to manually select band combinations (Babaeian et al., 2021; Darra et al., 2023, 2024, 2025; Li et al., 2022b; Kheir et al., 2024; Lai et al., 2025), AutoML systems can be leveraged to construct such spectral features. For instance, Naik et al. (2023) selected bands and functions to combine them using a tree-structured Parzen estimator (TPE). Darra et al. (2025) reported that features extracted using feature engineering algorithms such as PCA, which is available in many AutoML systems, can yield more accurate predictions than vegetation-index-based features for tomato yield prediction, suggesting a tradeoff between performance and interpretability. To satisfy the need for both high performance and interpretability in EO applications, AutoML systems need to include feature engineering algorithms that are similarly traceable to the input as vegetation indices are.

4.2 Model-centric AutoML in EO: limitations and complementarity

While many EO problems can be approached as regression or classification using AutoML tools, standard AutoML tools may not support configuring the most powerful models. For instance, Arp et al. (2024) applied Auto-sklearn to cloud inpainting by reframing it as a pixel-wise temporal autoregressive problem, using a cloud-free image from a different timestamp as input alongside the cloudy image to predict missing pixel values. The authors showed that more specialised algorithms are needed to capture spatial as well as temporal relationships, while implementing a regression model based on these strategies is not possible with an off-the-shelf AutoML tool since the number of neighbouring pixels available as features varies per sample depending on local cloud cover, violating the fixed-width feature matrix assumed by standard regression frameworks. Without spatial neighbourhood information, the AutoML-created regressors converged to predict the mean of the input features, a statistically reasonable outcome given the inconsistent relationship between the cloudy and cloud-free inputs, but one that yielded physically unrealistic reconstructions of cloud-covered areas. They proposed an algorithm for interpolating values through value propagation and showed how the hyperparameters of the algorithm can be tuned to make the algorithm self-adaptive to any scene. Their analysis further showed the complementarity of the AutoML-configured model and the specialised spatial interpolation method.

Chen et al. (2023) faced related challenges when applying H2O AutoML to geoscientific model ensembling for mapping global soil water retention parameters and estimating evapotranspiration. AutoML-generated models alone violated physical conservation laws, leading to unreliable results when extrapolating beyond the training distribution, which displays a clear instance of the evaluation gap, where in-distribution metric performance was not predictive of out-of-distribution results. They proposed to address the problem by reframing the role of AutoML: rather than predicting geophysical variables directly, AutoML was used to learn per-instance ensemble weights over a portfolio of physics models, each of which individually respects physical constraints. This approach yielded more accurate and physically consistent predictions, though it struggled with uneven data availability and other challenges associated with the employed physics models (Chen et al., 2023). Together, these examples suggest that generic AutoML systems tend to fail in EO tasks that require extrapolation because they optimise for in-distribution metric performance without any mechanism to enforce the structural or physical relationships that govern EO data. This behaviour is to be expected, as the problem is inherent to ML models in general and not only specific to generic AutoML models. However, it does reflect general requirements of EO applications that are not met by standard (Auto)ML systems alone. Therefore, rather than replacing domain knowledge, AutoML may be most effective for EO applications requiring extrapolation when combined with EO approaches where domain knowledge is explicitly encoded.

5 AutoML4EO methods research

We now turn to the second research goal in AutoML4EO: developing general methods with a broader scope than predictive analysis pipelines. Although AutoML frameworks are used to improve many predictive EO analytics pipelines, these frameworks still fail to address the requirements of image-based EO tasks, which involve spatial processing and complex data structures beyond traditional tabular prediction. Tabular frameworks such as Auto-sklearn (Feurer et al., 2022) and H2O AutoML (LeDell and Poirier, 2020) do not conserve spatial information from the images. NAS frameworks such as AutoKeras (Jin et al., 2019) and Auto-PyTorch (Zimmer et al., 2021) do not directly support common ML4EO tasks like object detection and segmentation, and are not designed to deal with heterogeneous data obtained from different sensors or stored in different formats. AutoML4EO methods research addresses these gaps, either by extending existing frameworks (Palacios Salinas et al., 2021; Wąsala et al., 2024) or by proposing new algorithms (see, e.g., Chen et al., 2019b; Zhang et al., 2022a; Li et al., 2022c), focusing on deep learning (Section 2.2.2). Neural networks are a natural choice for ML4EO because their flexible design theoretically allows for processing any data format. While AutoML for neural networks is not just restricted to NAS, HPO of training parameters has become best practice in deep learning, and is often not positioned prominently in the corresponding papers. Therefore, we focus on NAS algorithms proposed for or adapted to EO, organised in three data-centric and one model-centric research themes, and conclude with an analysis of the search spaces and search strategies used by the proposed NAS algorithms. Figure 6 shows the architectural components associated with each research theme.

5.1 Using data understanding to design AutoML4EO

Feature engineering tends to be primarily presented as a pre-processing task. However, in neural networks, a significant part — if not all — of feature extraction takes place inside the model. As a result, incorporating domain knowledge of the data in the search space of NAS can yield significant performance improvements. We discuss three data-

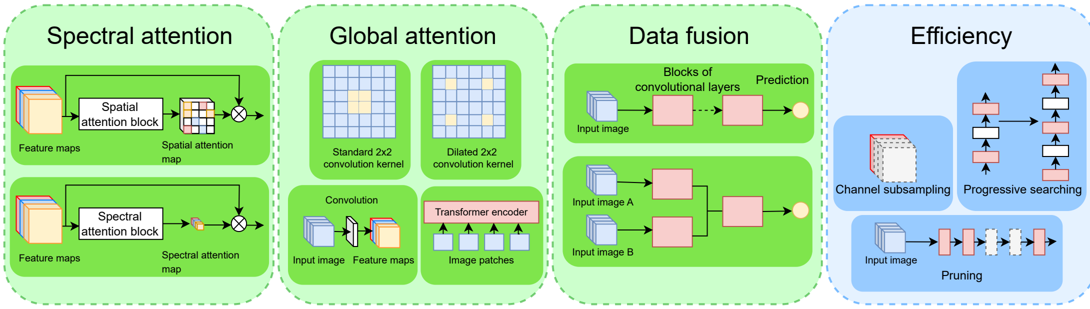

Figure 6: Much of AutoML4EO research revolves around four themes: three data-centric research directions (spectral attention, global attention, and data fusion) that use knowledge about EO data to design new architectural components, and one model-centric research theme on improving computational efficiency.

centric AutoML4EO research themes: incorporation of spectral attention, modelling global features, and multimodal data fusion.

5.1.1 Spatial-spectral attention

Spatial attention modules have long been used in natural image recognition tasks to help the model focus on the most important spatial features. In EO problems, spectral features can also be of high importance. For instance, in hyperspectral land cover classification, land classes can be differentiated based on spectral features alone, as has been shown in several studies applying broad-spectrum frameworks to land use/land cover (LULC) problems, where each training instance comprises the different bands of a single pixel, and spatial information is lost (Azedou et al., 2023; Adams et al., 2020; Zhang et al., 2021). Therefore, many AutoML4EO papers combine spatial and spectral attention and focus on designing or incorporating these modules into deep learning architectures. Zhang et al. (2022a) proposed 3-D-ANAS: a hierarchical NAS for LULC that incorporates spatial and spectral information. They introduced a new decomposed convolution that treats the spectral and spatial dimensions separately. Xue et al. (2022b), Zhong et al. (2022) and Shen et al. (2022) extended 3-D-ANAS with spatial and spectral attention blocks to improve the spatial and spectral features extracted by the decomposed convolution. Combining spatial and spectral feature extraction techniques can lead to more discriminative features. AutoML4EO studies in predictive analytics suggest that, beyond spectral information, including temporal information can further improve downstream performance. Still, satellite image time series and their temporal information remain relatively underexplored in AutoML4EO.

5.1.2 Global attention

In EO images, the global features provide context about the background or region, though they may lead to bias (e.g., geographic bias, where the model performs better in some geographic areas than others) if the context is not relevant to the task. The background provides context on the surroundings, for instance, whether the pictured area is from a coastal or desert region. Global context in CNNs can be extracted by increasing the receptive field, for example, by increasing the number of convolutional layers or by using dilated convolutions (Luo et al., 2016). Chen et al. (2022) and Wang et al. (2021b) included dilated convolutions in their NAS search space and found that they are selected more often in final networks than standard convolutions. Zhang and Ma (2024) also found that increasing kernel sizes and using dilated convolutions help extract global features and contextual information that distinguishes true garbage dump sites from lookalikes. Another approach is to use multi-scale feature fusion, where features from different depths in the network are merged and used for the final prediction (Shen et al., 2022). A third strategy is to use transformers to capture long-range spatial dependencies in satellite data (Naseer et al., 2021); for instance, Xue et al. (2022a) improved LULC performance by adding a transformer module to the 3-D-ANAS (Zhang et al., 2022a) classification head to model the global features, while Zhong et al. (2022) opted for a full transformer-based search space.

5.1.3 Data fusion

Features extracted from additional data sources can improve performance in many EO tasks. Combining data from different sources in a single deep learning pipeline to improve the quality of model predictions is also called data fusion. Data fusion leverages complementary information in the fused data sources to extract features that are more informative than features extracted from a single data source. Data fusion is connected to representation learning, as the biggest challenge lies in obtaining a joint representation of the heterogeneous input data with the correct information for a downstream task. Deep-learning-based data fusion in EO mainly focuses on fusing two data sources, such as optical data with SAR or hyperspectral data with LiDAR (Li et al., 2022a). Fusing more data sources can lead to performance gains (Hu et al., 2023), but these usually come at the cost of increased computational complexity.

AutoML4EO data fusion aims to find the optimal strategy to fuse the multimodal features, focusing on the design of the fusion module. Simple fusion strategies, such as concatenation or addition, may lead to information loss or inefficient representations that create a larger set of features rather than a joint representation mixing features from the input modalities. However, approaches to obtain representative joint representations require considerably more effort and domain knowledge to design (e.g., about how data sources complement each other), and it may not yet be possible to fully automate these procedures.

Li et al. (2021) presented a NAS approach for SAR-optical data fusion that only optimises module architecture, fusing the multimodal features and fixing the rest of the architecture for land cover classification. In later work on SAR-optical fusion, Li et al. (2022c) and Feng et al. (2025) introduced a NAS approach for data fusion in which each input modality is processed by a separate network branch. These modality branches are optimised independently to extract features specific to their respective modalities. Zhang et al. (2022d) proposed AutoMF for multimodal weather forecasting. They use hierarchical search to optimise the architecture on multiple levels – first, on the sub-network level, fusing temperature, humidity, wind, precipitation, and terrain type, and then at the level of the spatio-temporal sub-network, using fused features to produce weather forecasts.

Wąsala et al. (2025) extended the multi-branch idea to data fusion of any number of modalities with AutoMergeNet, which automatically generates multi-branch networks for data fusion and optimises three key architectural hyperparameters: the backbone, the depth at which multimodal features are fused, and the fusion strategy. Applied to methane plume detection, AutoMergeNet revealed two concrete instances of the limitations introduced in Section 1. First, despite the search space being well-matched to the problem structure, the network struggled to learn the natural hierarchy between the primary and auxiliary modalities during training, leading to visually empty images sometimes being misclassified as plumes because the auxiliary data layers contributed too strongly to the predictions. This was addressed by adding a unimodal classifier, which exclusively used the primary modality, thus explicitly enforcing the hierarchy. This problem can be framed as part of the optimisation gap: even when the architecture is reasonable, heterogeneous data with complex inter-modal relationships can produce a training signal that is too weak or inconsistent for search and training to reliably converge on well-performing fusion strategies without additional structural constraints. Unlike unimodal tasks where the training signal more directly reflects feature quality, multimodal fusion introduces ambiguity about which modality contributes what information and at which level, making it harder for the search to distinguish informative architectures from ones that happen to perform well on the available data.

5.2 Model-centric AutoML4EO via search and model efficiency

Not all AutoML4EO challenges can be solved by integrating domain knowledge about the data; some challenges are independent of unique data characteristics and require a model-centric approach. Improving the search and model efficiency is an important theme in AutoML4EO. The computational demands of AutoML methods vary considerably. The user-defined runtime of broad-spectrum frameworks typically ranges from minutes to hours for the EO applications discussed in Section 4, likely scaling with dataset size, though this is not reported explicitly. TPOT (Olson et al., 2016) is a notable exception: its evolutionary search strategy (see Section 2.1.2) is evaluation-intensive by design, and reported runtimes sometimes exceed 70 hours on tabular EO data (Choi et al., 2025; Kiala et al., 2020). Once NAS is introduced, and especially when moving to image data, compute times grow substantially, ranging from several GPU hours to multiple GPU days. The majority of NAS approaches surveyed here are based on DARTS (Liu et al., 2018), which uses gradient-based search and is generally more efficient than earlier methods. This stands in contrast to NASNet (Zoph et al., 2018), which relies on reinforcement learning-based search and required 2000 GPU hours: an order of magnitude higher in computational cost, largely due to the cost associated with its search strategy.

These moderate computational requirements may reflect the reality that most AutoML4EO research is developed under constrained compute budgets, often runnable on a single workstation with a single GPU. Nevertheless, computational cost remains a practical concern. AutoML4EO often further increases the computational demands of NAS methods. First, satellite images can be large (for instance, when the spatial or spectral resolution is high, or when large images need to be used in order to provide sufficient contextual information for accurate classification). Second, tasks such as segmentation and object detection are among the more computationally heavy ML tasks because networks for these tasks are large, use large amounts of memory, and need more epochs to converge than image classification networks (Mohan et al., 2023). When many experiments are needed, e.g., for hyperparameter importance analysis, cross-dataset validation, or ablation studies, per-run costs may accumulate quickly. Precisely attributing runtime differences to specific causes (e.g., dataset size, image dimensions, size of search space, search strategy, or stopping criterion) is difficult in practice, because the surveyed works vary substantially in experimental design and do not always include ablations or comparisons to the original unmodified algorithms. This makes computational efficiency a hard dimension to evaluate systematically, and a valuable direction for future work. Here, we discuss how AutoML4EO researchers have addressed the computational complexity through more efficient search procedures and lightweight architectures.

Efficient NAS. One-shot NAS approaches tend to be more computationally efficient than sample-based ones (Section 2.2.2). As a result, many NAS for EO approaches combine a gradient-based search strategy with a cell-based search space. However, keeping the full supernet in memory results in high memory consumption (Liu et al., 2018). Zhang et al. (2022b) and Jing et al. (2020) reduced the memory consumption of DARTS in ship detection and satellite image classification, respectively, by following PC-DARTS (Xu et al., 2020) in taking a random sample of the feature maps at each layer instead of calculating gradients based on the full set of feature maps.

In LULC, Wang et al. (2022b) applied progressive searching to decouple the search into multiple phases, fixing the best hyperparameters found in previous stages and optimising a new set at each stage. This strategy reduces the memory consumption during the search because not all hyperparameter value paths have to be kept in memory simultaneously. Xue et al. (2022a) used network path pruning to remove architecture choices with bad performance in SAR-based height estimation from the search space and path binarisation (as introduced by Cai et al., 2019) during the search to reduce memory consumption by decreasing the number of bits needed to store the network. Furthermore, they employ gradient-based search using the differentiable architecture sampler (GDAS), which subsamples the supernet during the search using gradient descent to reduce search complexity (Dong and Yang, 2019a).

Another strategy is to reduce the number of operations in the search space to limit memory consumption (Zhang et al., 2020). Finally, while generalised NAS approaches tend to focus on optimising full network architectures (see, e.g., Liu et al., 2018), the size of the search space, and therefore, the computational complexity of the search can be severely reduced by fixing most of the architecture and optimising only a few critical layers, such as the classification head or fusion modules in data fusion.

Efficient architectures. Edge computing — running ML algorithms on satellites — can avoid expensive downlinking of cloudy images (Giuffrida et al., 2022). However, the cost of sending hardware to space and the dependence on solar power for their energy supply limit the in-orbit computational power of satellites. Furthermore, the process can result in the loss of valuable data. NAS can address this problem by designing neural networks to optimise accuracy and minimise the number of parameters, and thus to reduce inference cost using multi-objective optimisation. Du et al. (2023) and Ao et al. (2023) used multi-objective optimisation to minimise the number of parameters in the creation of deep learning models for edge computing. Traore et al. (2021) used transformable architecture search (TAS) to

Search space	Methods
Single module	Wang et al., 2023b; Li et al., 2021, 2020
Macro	Dong et al., 2020; Polonskaia et al., 2021; Han et al., 2022; Xue et al., 2022a; de Paulo et al., 2022; Peng et al., 2023; Wąsala et al., 2025
Micro	Du et al., 2023; Chen et al., 2021; Broni-Bediako et al., 2022; Chen et al., 2022; Xiao et al., 2024; Zhang et al., 2020; Gudzius et al., 2023; Zhang et al., 2022e; Wang et al., 2022a; Chen et al., 2019b; Traore et al., 2021; Peng et al., 2021; Shen et al., 2022; Li et al., 2022c; Wan et al., 2022; Liu et al., 2021a,b; Chen et al., 2019a
Hierarchical	Wang et al., 2024a; Zhang et al., 2022c; Wang et al., 2021b; Zhong et al., 2022; Zhang et al., 2022a; Ao et al., 2023; Wang et al., 2022b; Zhang et al., 2022d, 2023

Table 2: Types of search spaces used in AutoML4EO design.

optimise the size of a given network (Dong and Yang, 2019b), which leads to significantly smaller architectures than the baselines while maintaining comparable performance. Liang et al. (2020) reduced the number of parameters of the final aerial image classification network by pruning candidate networks during the search. Finally, Zhu et al. (2022) reduced the number of network parameters by converting floating-point convolutional layers to binary convolutions when decoding the best network from the supernet.

5.3 Optimisation gap in AutoML4EO

While the preceding sections reviewed AutoML4EO methods along data- and model-centric axes, examining them through the lens of the search space and strategy taxonomy introduced in Section 2.2.2 reveals additional patterns. Rather than designing full networks from scratch, AutoML4EO methods cover a broad spectrum, from approaches that optimise a single module, such as a feature pyramid network (FPN) (Lin et al., 2017) or a data fusion module (Wang et al., 2023b; Li et al., 2021, 2020), to approaches that jointly optimise macro- and micro-level design choices in hierarchical search spaces (see Table 2).

Despite the expressiveness of macro and hierarchical search spaces, the majority of AutoML4EO works covered in this survey use cell-based search spaces, commonly with differentiable search strategies such as DARTS (see, e.g., Du et al., 2023; Chen et al., 2022; Xiao et al., 2024; Zhang et al., 2020; Gudzius et al., 2023; Zhang et al., 2022e; Wang et al., 2022a; Chen et al., 2019b; Traore et al., 2021; Li et al., 2022c; Peng et al., 2021; Shen et al., 2022). In practice, some works further relax the constraint that every cell must be identical (de Paulo et al., 2022; Peng et al., 2023), yielding search spaces that are somewhat more expressive than canonical micro-level search, but still far from the full architectural diversity that macro or hierarchical search spaces would allow. This pattern of constraining the optimisation problem is further reinforced by the use of pre-trained weights: several works initialise the search from pre-trained blocks obtained from satellite image datasets (Palacios Salinas et al., 2021; Wąsala et al., 2024). This goes a step beyond the practice noted in Section 2.2.2 of restricting the search spaces of NAS for dense image tasks to minor design decisions of known architectures by embedding prior knowledge directly into the initialisation. Furthermore, even when hierarchical search spaces are employed, the available design choices are often constrained, particularly in the macro layer, with some as simple as choosing between a few types of blocks (Zhang et al., 2022a; Zhong et al., 2022). Other works, such as LoveNAS (Wang et al., 2024a), apply their hierarchical NAS algorithm only to a section of the network, while fixing the rest. Together, these choices reflect the data and compute constraints inherent to EO applications, which can make it more difficult to effectively search large and complex search spaces for truly novel solutions. As a result, many AutoML4EO algorithms do not step far beyond fine-tuning networks within established architectural families such as ResNet or U-Net, yielding better predictive pipelines but possibly not significantly advancing the field.

Current AutoML4EO works mostly focus on the design of the search space, which makes sense, because this is where domain knowledge of a problem can be most directly encoded, for instance by incorporating task-appropriate backbones or data fusion modules. Yet, the works surveyed also employ a wide variety of search strategies beyond DARTS, including random search (Chen et al., 2021; Broni-Bediako et al., 2022), evolutionary algorithms (Wan et al., 2022), particle swarm optimisation (Liu et al., 2021b), and weight-based mini-batch updates (Liu et al., 2021a). It is difficult to draw conclusions from this diversity, however, because any modifications to search strategies are minor, and the resulting search strategies are not shown to be effective in exploring more expressive search spaces that could lead to novel architectures. Therefore, there appears to be a gap between the complexity of the EO problems researchers and domain experts are interested in solving, and the complexity of the optimisation problems current AutoML systems are able to solve (i.e., the optimisation gap introduced in Section 1). Addressing this gap is key to creating AutoML4EO systems that can discover novel architectures for new or understudied applications.

6 Outlook

In this paper, we analysed how AutoML frameworks enhance EO task performance and surveyed a large body of research on the design of AutoML4EO algorithms (Appendix A). Much of AutoML4EO focuses on data-centric techniques, such as feature selection, with relatively little attention to more model-centric approaches, such as leveraging the complementarity between AutoML and physics models. In this section, we present four key challenges in advancing AutoML4EO frameworks: evaluation gap, standardised evaluation protocols, interpretability in the context of bias, and learning from unlabelled data. We discuss research opportunities to address each challenge.

6.1 Integrate best practices from the geospatial domain to address the evaluation gap

Current AutoML4EO systems face significant challenges in evaluation, which may harm replicability (Kapoor et al., 2024) and the resulting pipeline's ability to generalise to previously unseen data. These systems adapt empirical design practices from traditional ML research, which typically ignores practices specific to geospatial data. However, AutoML systems rely on accurate performance estimates to guide the search for good models, hyperparameters, and pipeline configurations. As a result, when the evaluation strategy is flawed,

Section	Perspective	Tasks	References
4.1.1 Feature selection	Data	LULC Crop yield prediction, landslide susceptibility mapping, reservoir characterisation, estimating biophysical parameters, weather forecasting, cloud detection	Bai et al., 2022; Bodini, 2024; Babaeian et al., 2021; Bruzón et al., 2021; César de Sá et al., 2022; Darra et al., 2024, 2025, 2023; Kasimati et al., 2022; Kheir et al., 2024; Kiala et al., 2020; Kiedanski et al., 2025; Lai et al., 2025; Lee et al., 2023, 2024; Li et al., 2022b; Liu et al., 2025; Ma et al., 2023; Mangalath Ravindran et al., 2022; Mubarak and Koeshidayatullah, 2023; Paluba et al., 2025; Renza et al., 2021; Šandera and Štych, 2024; Singh et al., 2021; Sun et al., 2021; Tang et al., 2023; Xia et al., 2023; Zhang et al., 2021, 2025, 2024; Zheng et al., 2023; Su et al., 2024; Liang et al., 2024
4.1.2 Feature engineering	Data	Crop yield prediction, anomalous ship detection, weather forecasting	Naik et al., 2023; Darra et al., 2024; Lai et al., 2025; Kurchaba et al., 2023; Maulik et al., 2020
4.2 Model complementarity	Model	Cloud inpainting, mapping soil parameters	Arp et al., 2024; Chen et al., 2023
5.2 Efficiency	Model	LULC, ship detection, image classification	Zhang et al., 2022b; Jing et al., 2020; Wang et al., 2022b; Xue et al., 2022a; Zhang et al., 2020; Ju et al., 2025; Ao et al., 2023; Traore et al., 2021; Liang et al., 2020; Zhu et al., 2022
5.1.1 Spatial-spectral attention	Data	LULC	Zhang et al., 2022a; Xue et al., 2022b; Zhong et al., 2022; Shen et al., 2022
5.1.2 Global attention	Data	LULC, Height estimation, image classification	Chen et al., 2022; Wang et al., 2021b; Zhang and Ma, 2024; Shen et al., 2022; Zhang et al., 2022a; Xue et al., 2022b; Zhong et al., 2022
5.1.3 Data fusion	Data	LULC, Atmospheric plume detection, Weather forecasting	Wąsala et al., 2025; Li et al., 2021, 2022c; Feng et al., 2025; Zhang et al., 2022d

Table 3: Categorisation of AutoML4EO summarising the EO tasks and showing the number of references in each category (Section 3). Feature selection literature is the best represented, while few works explore the complementarity between AutoML and physics models. All works in the spatial-spectral attention section exclusively focus on land use and land cover (LULC) classification, suggesting spatial-spectral features are especially useful for this task.

promising configurations may be wrongly discarded during search. Geospatial best practices (Maxwell et al., 2021a,b, 2022; Li et al., 2024; Koldasbayeva et al., 2024; Jemeljanova et al., 2024; Kedron et al., 2021) emphasise avoiding information leakage from training to testing data (see, e.g., Kapoor and Narayanan, 2023) in the empirical design to obtain accurate and generalisable performance estimates. Biases in ML4EO data, such as spatiotemporal autocorrelation, can lead to overestimated performance if not accounted for in the evaluation, an issue raised or confirmed in many studies (Li et al., 2024; Koldasbayeva et al., 2024; Maxwell et al., 2021b, 2022; Jemeljanova et al., 2024; Roberts et al., 2017; Karasiak et al., 2022; Oliveira et al., 2021). Despite these known issues, many AutoML4EO systems do not fully consider these factors in their evaluation strategies. We discuss two opportunities to mitigate the effects of data leakage at two stages of AutoML4EO: implementing spatiotemporal data partitioning schemes during search and evaluating final pipelines on use case applications.

Opportunity: Spatiotemporal cross-validation

It is common practice in ML to split labelled datasets into training, validation, and testing partitions, using stratified sampling to ensure equal class distributions in each partition. Geospatial data presents additional challenges in creating such partitions due to spatiotemporal autocorrelation, where nearby pixels in space or time tend to share similar characteristics. When not addressed properly, spatiotemporal autocorrelation can lead to high correlation between the training and testing partitions. Studies found a performance discrepancy of as much as 40% between random and spatial or block sampling, showing that randomly partitioning data can severely overestimate performance on testing data (Roberts et al., 2017; Kedron et al., 2021; Traore et al., 2021). As a result, AutoML frameworks may select and configure models based on bloated performance metrics if spatiotemporal autocorrelation is not addressed. The impact of cross-validation strategies on AutoML pipeline performance remains underexplored in AutoML4EO. Implementing spatiotemporal cross-validation could be a low-hanging fruit for creating more robust pipelines, particularly for small datasets highly susceptible to spatiotemporal autocorrelation (as often encountered in predictive analytics, Section 4). Comparing results across different data partitioning schemes can also reveal the sensitivity of an AutoML4EO system to variations in given training data.

Opportunity: Real-world applications to diagnose data leakage

Leakage can also occur when the distribution of the labelled testing set significantly differs from that of the target population or the data to which the model would be applied in practice (Kapoor and Narayanan, 2023). When creating class-balanced data partitions to address class-imbalanced problems, a new problem can easily arise: a mismatch between the class distribution in the testing data and data encountered in operational scenarios. ML4EO works address this problem with real-world applications, but these analyses require significant domain expertise and are hard, if not impossible, to automate (see, e.g., Tang et al., 2023; Li et al., 2022b). Real-world application analyses differ from testing set evaluation, as they typically involve unlabelled regions and larger geographic scales. While time-intensive, these analyses provide crucial insights into model generalisation for practical applications. Establishing real-world applications as standard practice in AutoML4EO will require collaboration between AutoML researchers and domain experts, since domain expertise is necessary to conduct meaningful analyses of unlabelled regions.

6.2 Standardised evaluation with benchmark datasets

Standardised evaluation is the cornerstone of ML methods research. However, establishing consistent evaluation in AutoML4EO presents unique challenges due to the sheer diversity of EO tasks and datasets. This fragmentation makes it difficult to fairly compare AutoML4EO algorithms and measure improvements in the ML pipelines thus obtained. A critical component of this evaluation challenge involves baseline comparisons; meta-analyses of ML applications in other disciplines have shown that weak baselines are a persistent problem and lead to overly optimistic research findings (McGreivy and Hakim, 2024; DeMasi et al., 2017). However, fair baseline comparison in AutoML4EO is complicated by inconsistent evaluation practices. Many studies build upon common foundation algorithms. Yet, comparing performance across different implementations and datasets remains difficult due to varying setups of experiments and evaluation protocols (e.g., data partitioning). For instance, cross-study comparisons between competitive solutions based on the same underlying algorithm become challenging without standardised benchmarking procedures. We discuss two opportunities to standardise the evaluation of AutoML4EO datasets.

Opportunity: Reusing foundation model benchmark infrastructure

Benchmarks play a crucial role in AutoML research by enabling fair comparisons and speeding up development through improved access to data and the ability to evaluate the performance of candidate ML pipeline configurations. Benchmark datasets, for instance, represent fundamental data properties shared by similar datasets. The rise in popularity of geospatial foundation models has led to the development of evaluation benchmarks that include many different tasks and sensors, such as PhilEO Bench (Fibaek et al., 2024) and PANGAEA (Marsocci et al., 2024). These benchmarks collect hundreds of gigabytes of labelled EO data and provide the infrastructure to use the data and evaluate resulting models. The recent assembly of these foundation model benchmarks enables more standardised evaluation of AutoML4EO methods.

Opportunity: Expanding NAS benchmarks

Another opportunity lies in the development of NAS benchmarks for EO. NAS benchmarks (e.g., see Ying et al., 2019; Bansal et al., 2022) collect the performance of all possible networks within common NAS search spaces on widely-used datasets like CIFAR-10 (Krizhevsky, 2009). Search space benchmarks facilitate the development of new search strategies, because they make the performance of candidate networks available without the need for costly evaluations. Therefore, the expansion of NAS search space benchmarks with results on EO datasets, such as NWPU-RESISC

(Cheng et al., 2017) and EuroSAT (Helber et al., 2019), represents an opportunity to address the challenge of computational cost of AutoML4EO (Section 5.2).

6.3 Mitigating bias

Observational data, such as remote sensing data, may seem inherently objective, because sensors indiscriminately record information. However, bias can occur in EO data in many different ways (McGovern et al., 2022, 2024). Some bias is inherent — for instance, sensors are limited to detecting signals within their sensitivity range. Other biases arise during dataset creation, when samples are selected from vast volumes of EO data. For example, oversampling rare phenomena, such as methane plumes, to create balanced datasets introduces sampling bias (Mehrabi et al., 2021), which can lead to many false positives when propagated to the model (McGovern et al., 2024; Wąsala et al., 2025). Similarly, representation bias (Mehrabi et al., 2021) in datasets lacking diversity may produce confounders that lead to shortcut learning (Xiong et al., 2022), where models learn relationships that exist only in the training data, but not in the real world. For instance, Wąsala et al. (2026) found that missing pixels can lead to representation bias and that this issue can remain undetected when using standard performance metrics such as precision, accuracy, and recall. The reduced human oversight typically associated with the use of AutoML techniques can amplify the risk of incurring such biases. While geospatial data formats, such as NetCDF and GeoTIFF, contain rich metadata for identifying subgroups (e.g., time stamp, location, and auxiliary data such as cloud masks), current AutoML frameworks cannot easily access this information for subgroup analysis. Instead, evaluations rely on model-centric metrics, such as overall accuracy, which fail to reveal poor performance on important data subsets. We highlight a research opportunity to better leverage metadata in AutoML4EO to mitigate bias.

Opportunity: Multi-objective optimisation for subgroup fairness

While geospatial metadata can provide a solid basis for model evaluation, it could also be leveraged earlier in the AutoML4EO pipeline to create less biased models. The concepts of bias and fairness in ML are usually applied to tasks with direct impact on humans, ensuring equal performance across subgroups defined by demographic attributes (e.g., equal performance regardless of gender, Mehrabi et al., 2021). Many metrics exist to assess different aspects of fairness, such as differences in error rates between data subgroups. This concept of group fairness could be extended to EO problems using geospatial metadata attributes to define subgroups. Such an analysis could, for instance, enable measurement of the geographic bias of a given model. Fairness metrics can then be used in multi-objective AutoML systems to create ML pipelines that are both accurate and fair, similar to how efficiency-focused AutoML4EO optimises both accuracy and computational efficiency (Section 5.2). The potential of such multi-objective, fairness-aware AutoML4EO systems lies in greater control over the final pipelines in predictive analytics. However, a significant challenge remains in identifying important subgroups that should have similar performance, which requires extensive domain knowledge.

6.4 Learning from unlabelled data

Unlabelled data collected by satellites represents an underexplored resource in AutoML4EO. Satellites collect much more data than can realistically be labelled. For example, the original pair of Sentinel-2 satellites collects 3 TB of data every day, [2] corresponding to 20 times the size of the ImageNet dataset (Deng et al., 2009). Including unlabelled data in ML pipelines may improve performance (Van Engelen and Hoos, 2020); unlabelled data could also mitigate sampling or representation bias present in labelled data by increasing dataset diversity without the cost of additional labelling. Furthermore, enriching the training data with unlabelled samples can align the training data distribution with the real world, reducing data leakage. ML techniques that leverage unlabelled data prominently include semi- and self-supervised learning. We discuss an opportunity to include unlabelled data in AutoML4EO.

Opportunity: AutoML4EO with self-supervised pre-training

Recently, foundation models (FMs) have emerged as strong alternatives to traditional broad-spectrum AutoML frameworks across multiple domains. In tabular and time-series data, models such as TabPFN (Hollmann et al., 2025) and Chronos (Ansari et al., 2024) achieve high performance through zero-shot inference in a single pass, without requiring fine-tuning. A similar shift has taken place in computer vision, with self-supervised geospatial FMs such as Prithvi (Jakubik et al., 2023) and SatMAE (Cong et al., 2022), and multimodal FMs are beginning to emerge (see, e.g., Guo et al., 2024; Wang et al., 2025; Han et al., 2024).

However, the applicability of FMs in EO remains constrained along two dimensions. First, geospatial FMs are limited by the datasets available for pre-training (primarily Sentinel-1 and Sentinel-2 data), and tasks requiring unconventional architectures, such as unconventional output heads, specialised loss functions, and non-standard combinations of modalities (see, e.g., Hanna et al., 2023; Wąsala et al., 2025), remain outside their scope. Second, even for tasks within their applicability domain, the diversity of EO modalities, geographies, and tasks means that the pre-training distribution frequently does not cover the target task, and task-specific ML4EO models still outperform task-agnostic geospatial FMs (Mai et al., 2024).

AutoML could bridge these gaps in two complementary ways. Next-generation AutoML4EO systems could adaptively design new networks for any combination of modalities and task requirements, including specialised heads for FM backbones. Additionally, NAS approaches that leverage unlabelled data could expand the applicability of AutoML4EO, given that satellites collect far more data than can realistically be labelled. However, no semi-supervised AutoML or NAS system currently exists for image recognition. More broadly, AutoML with unlabelled data alone may be challenging: without labels, evaluation must rely on self-supervised or proxy metrics, which risks widening the generalisation gap between AutoML-optimised ML pipelines and real-world applications.

2. https://www.eoportal.org/satellite-missions/copernicus-sentinel-2 Last accessed 16 June 2026

7 Conclusions

As a result of the rapid increase in available computational resources and satellite data, the research areas of AutoML and ML4EO are growing fast. However, ML is not fully leveraged for many EO data sources due to the lack of suitable automatic feature extraction techniques. Moreover, evaluation frameworks are often not designed to account for biases in EO data, reflecting a broader challenge of ensuring that models generalise to real-world deployment conditions that differ from training data. We believe the primary goal of AutoML4EO research should be to bridge this generalisation gap, so that we can automatically generate ML pipelines that provide a strong, validated starting point for operational deployment, while acknowledging that ML and domain expertise remain essential for final validation and deployment decisions. Achieving this requires rethinking the design and evaluation of AutoML4EO systems. Libraries such as Auto-sklearn and Auto-Keras accelerate the development of ML models by automating technical design choices, yet do not incorporate domain knowledge during ML pipeline optimisation, which can lead to automatically created pipelines that underperform in real-world use cases compared to expert-designed approaches. The gap between expert-designed and automatically designed systems extends beyond methodological design choices (such as the network architecture) to evaluation. Experts can better identify mismatches between training data and deployment conditions, and design evaluation frameworks that expose these differences. In the near term, significant advances in AutoML4EO will require collaboration between EO practitioners and ML researchers to develop application-aware AutoML4EO systems with robust evaluation frameworks. Over the long term, such systems could contribute to AI for Science by enabling EO experts to efficiently and reliably develop ML pipelines for novel problems, while reducing the manual effort required to analyse large volumes of satellite data.

Broader Impact Statement

This survey paper aims to give a broad overview of the field of Automated Machine Learning for Earth Observation (AutoML4EO). AutoML4EO systems that are robust to deployment conditions including class imbalance, noise, and bias, can increase the efficiency and reproducibility of developing data-driven EO analysis pipelines, stimulating scientific insights and breakthroughs in EO. However, there is always a risk in automation and removing human oversight. Therefore, AutoML4EO systems and the resulting ML4EO pipelines should always be thoroughly evaluated before the ML4EO pipeline is deployed. Furthermore, human experts should always oversee operational ML systems and frequently evaluate them for biases and performance degradation.

Acknowledgments and Disclosure of Funding

Each application of ML4EO has its own unique characteristics, and this survey could not have been written without the input and perspectives from practitioners. We thank Patrick Sebel from the ESA Φ-Lab for the discussions about (hyperspectral) land cover classifications. We extend further thanks to Laurens Arp for sharing his insights about the role of the lack of ground truth in EO applications, and Abygail Stegenga for sharing her insights about inter- and extrapolation. Parts of this research have been supported by: the "Physics-aware Spatio-temporal Machine Learning for Earth Observation Data" project (project number OCENW.KLEIN.425) of the Open Competition ENW research programme, which is partly financed by the Dutch Research Council (NWO); the Open Space Innovation Platform[3] as a Co-Sponsored Research Agreement and carried out under the Discovery programme of, and funded by, the European Space Agency (contract number 4000136204/21/NL/GLC/my); and an Alexander von Humboldt Professorship in Artificial Intelligence awarded to Holger Hoos.

References

Matthew D. Adams, Felix Massey, Karl Chastko, and Calvin Cupini. Spatial modelling of particulate matter air pollution sensor measurements collected by community scientists while cycling, land use regression with spatial cross-validation, and applications of machine learning for data correction. *Atmospheric Environment*, 230:117479, June 2020. ISSN 1352-2310. doi: 10.1016/j.atmosenv.2020.117479.

Michał Affek and Julian Szymański. A survey on the datasets and algorithms for satellite data applications. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 17:1–23, 2024. ISSN 2151-1535. doi: 10.1109/JSTARS.2024.3424954.

Edesio Alcobaça and André C. P. L. F. de Carvalho. A Literature Review on Automated Machine Learning. *Artificial Intelligence Review*, 59(1):5, November 2025. ISSN 1573-7462. doi: 10.1007/s10462-025-11397-2.

Moez Ali. *PyCaret: An open source, low-code machine learning library in Python*. PyCaret, April 2020. URL `https://www.pycaret.org`. PyCaret version 1.0.

Abdul Fatir Ansari, Lorenzo Stella, Ali Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Bernie Wang. Chronos: Learning the Language of Time Series. *Transactions on Machine Learning Research*, May 2024. ISSN 2835-8856.

Lei Ao, Kaiyuan Feng, Kai Sheng, Hongyu Zhao, Xin He, and Zigang Chen. TPE-NAS: A Two-Phase Evolutionary Neural Architecture Search for Remote Sensing Image Classification. *Remote Sensing*, 15(8):2212, January 2023. ISSN 2072-4292. doi: 10.3390/rs15082212.

Laurens Arp, Holger Hoos, Peter van Bodegom, Alistair Francis, James Wheeler, Dean van Laar, and Mitra Baratchi. Training-Free Thick Cloud Removal for Sentinel-2 Imagery Using Value Propagation Interpolation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 216:168–184, October 2024. ISSN 0924-2716. doi: 10.1016/j.isprsjprs.2024.07.030.

3. https://ideas.esa.int

Axel de Romblay. Mlbox, 2019. URL https://mlbox.readthedocs.io/en/latest/.

Ali Azedou, Aouatif Amine, Isaya Kisekka, Said Lahssini, Youness Bouziani, and Said Moukrim. Enhancing Land Cover/Land Use (LCLU) classification through a comparative analysis of hyperparameters optimization approaches for deep neural network (DNN). *Ecological Informatics*, 78:102333, December 2023. ISSN 1574-9541. doi: 10.1016/j.ecoinf.2023.102333.

Ebrahim Babaeian, Sidike Paheding, Nahian Siddique, Vijay K. Devabhaktuni, and Markus Tuller. Estimation of root zone soil moisture from ground and remotely sensed soil information with multisensor data fusion and automated machine learning. *Remote Sensing of Environment*, 260:112434, July 2021. ISSN 0034-4257. doi: 10.1016/j.rse.2021.112434.

Zhigang Bai, Qimeng Liu, and Yu Liu. Groundwater Potential Mapping in Hubei Region of China Using Machine Learning, Ensemble Learning, Deep Learning and AutoML Methods. *Natural Resources Research*, 31(5):2549–2569, October 2022. ISSN 1573-8981. doi: 10.1007/s11053-022-10100-4.

Prasanna Balaprakash, Michael Salim, Thomas D. Uram, Venkat Vishwanath, and Stefan M. Wild. DeepHyper: Asynchronous Hyperparameter Search for Deep Neural Networks. In *2018 IEEE 25th International Conference on High Performance Computing (HiPC)*, pages 42–51, Piscataway, New Jersey, United States, December 2018. IEEE. doi: 10.1109/HiPC.2018.00014.

Archit Bansal, Danny Stoll, Maciej Janowski, Arber Zela, and Frank Hutter. JAHS-Bench-201: A Foundation For Research On Joint Architecture And Hyperparameter Search. *Advances in Neural Information Processing Systems*, 35:38788–38802, December 2022.

Mitra Baratchi, Can Wang, Steffen Limmer, Jan N. van Rijn, Holger Hoos, Thomas Bäck, and Markus Olhofer. Automated machine learning: Past, present and future. *Artificial Intelligence Review*, 57(5):122, April 2024. ISSN 1573-7462. doi: 10.1007/s10462-024-10726-1.

Dilyara Baymurzina, Eugene Golikov, and Mikhail Burtsev. A Review of Neural Architecture Search. *Neurocomputing*, 474:82–93, February 2022. ISSN 0925-2312. doi: 10.1016/j.neucom.2021.12.014.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(10):281–305, 2012.

Bernd Bischl, Martin Binder, Michel Lang, Tobias Pielok, Jakob Richter, Stefan Coors, Janek Thomas, Theresa Ullmann, Marc Becker, Anne-Laure Boulesteix, Difan Deng, and Marius Lindauer. Hyperparameter Optimization: Foundations, Algorithms, Best Practices, and Open Challenges. *WIREs Data Mining and Knowledge Discovery*, 13(2): e1484, 2023. ISSN 1942-4795. doi: 10.1002/widm.1484.

Matteo Bodini. Daily Streamflow Forecasting Using AutoML and Remote-Sensing-Estimated Rainfall Datasets in the Amazon Biomes. *Signals*, 5(4):659–689, December 2024. ISSN 2624-6120. doi: 10.3390/signals5040037.

Clifford Broni-Bediako, Yuki Murata, Luiz H. B. Mormille, and Masayasu Atsumi. Searching for CNN Architectures for Remote Sensing Scene Classification. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–13, 2022. ISSN 1558-0644. doi: 10.1109/TGRS.2021.3097938.

Adrián G. Bruzón, Patricia Arrogante-Funes, Fátima Arrogante-Funes, Fidel Martín-González, Carlos J. Novillo, Rubén R. Fernández, René Vázquez-Jiménez, Antonio Alarcón-Paredes, Gustavo A. Alonso-Silverio, Claudia A. Cantu-Ramirez, and Rocío N. Ramos-Bernal. Landslide Susceptibility Assessment Using an AutoML Framework. *International Journal of Environmental Research and Public Health*, 18(20):10971, January 2021. ISSN 1660-4601. doi: 10.3390/ijerph182010971.

Han Cai, Ligeng Zhu, and Song Han. Proxylessnas: Direct neural architecture search on target task and hardware. In *7th International Conference on Learning Representations, ICLR 2019*, New Orleans, Louisiana, USA, 2019. OpenReview.net. URL `https://openreview.net/forum?id=HylVB3AqYm`.

Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once for all: Train one network and specialize it for efficient deployment. In *Proceedings of the 8th International Conference on Learning Representations, ICLR 2020*, Addis Ababa, Ethiopia, 2020. OpenReview.net. URL `https://openreview.net/forum?id=HylxE1HKwS`.

Rich Caruana, Alexandru Niculescu-Mizil, Geoff Crew, and Alex Ksikes. Ensemble Selection from Libraries of Models. In *Twenty-First International Conference on Machine Learning - ICML '04*, page 18, Banff, Alberta, Canada, 2004. ACM Press. doi: 10.1145/1015330.1015432.

Julio Castro Lopes, Jose Luis Oliveira, and Rui Pedro Lopes. A comprehensive review of synthetic image generation methods in remote sensing. *International Journal of Remote Sensing*, 46(15):5773–5801, August 2025. ISSN 0143-1161, 1366-5901. doi: 10.1080/01431161.2025.2527373.

Hao Chen, Tiejun Wang, Yonggen Zhang, Yun Bai, and Xi Chen. Dynamically weighted ensemble of geoscientific models via automated machine-learning-based classification. *Geoscientific Model Development*, 16(19):5685–5701, October 2023. ISSN 1991-959X. doi: 10.5194/gmd-16-5685-2023.

Jie Chen, Haozhe Huang, Jian Peng, Jiawei Zhu, Li Chen, Chao Tao, and Haifeng Li. Contextual Information-Preserved Architecture Learning for Remote-Sensing Scene Classification. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2022. doi: 10.1109/TGRS.2021.3063287.

Pengdi Chen, Yuanrui Ren, Baoan Zhang, and Yuan Zhao. Class Imbalance in the Automatic Interpretation of Remote Sensing Images: A Review. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 18:9483–9508, 2025. ISSN 2151-1535. doi: 10.1109/JSTARS.2025.3555567.

Xin Chen, Lingxi Xie, Jun Wu, and Qi Tian. Progressive Differentiable Architecture Search: Bridging the Depth Gap Between Search and Evaluation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1294–1303, Seoul, Korea (South), October 2019a. IEEE. doi: 10.1109/ICCV.2019.00138.

Yuling Chen, Wentao Teng, Zhen Li, Qiqi Zhu, and Qingfeng Guan. Cross-Domain Scene Classification Based on a Spatial Generalized Neural Architecture Search for High Spatial Resolution Remote Sensing Images. *Remote Sensing*, 13(17):3460, January 2021. ISSN 2072-4292. doi: 10.3390/rs13173460.

Yushi Chen, Kaiqiang Zhu, Lin Zhu, Xin He, Pedram Ghamisi, and Jón Atli Benediktsson. Automatic Design of Convolutional Neural Network for Hyperspectral Image Classification. *IEEE Transactions on Geoscience and Remote Sensing*, 57(9):7048–7066, September 2019b. ISSN 1558-0644. doi: 10.1109/TGRS.2019.2910603.

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote Sensing Image Scene Classification: Benchmark and State of the Art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017. doi: 10.1109/JPROC.2017.2675998.

Sungwon Choi, Jeon Woojin, Jung Imgook, Jung Daeseong, Sim Suyoung, Woo Jongho, Kim Honghee, Park Sungwoo, Kim Seungwon, Lee Seungkyoo, and Kyung-Soo Han. Integrated Detection of Ships and Their Wakes Using Sentinel-2A Imagery and Automated Machine Learning Approaches. *Marine Geodesy*, 0(0):1–23, 2025. ISSN 0149-0419. doi: 10.1080/01490419.2025.2486149.

Xiangxiang Chu, Tianbao Zhou, Bo Zhang, and Jixiang Li. Fair DARTS: Eliminating Unfair Advantages in Differentiable Architecture Search. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV*, page 465–480, Berlin, Heidelberg, 2020. Springer-Verlag. doi: 10.1007/978-3-030-58555-6_28.

Yezhen Cong, Samar Khanna, Chenlin Meng, Patrick Liu, Erik Rozi, Yutong He, Marshall Burke, David B. Lobell, and Stefano Ermon. SatMAE: Pre-training Transformers for Temporal and Multi-Spectral Satellite Imagery. In *Advances in Neural Information Processing Systems*, 2022.

Ekin D. Cubuk, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le. AutoAugment: Learning Augmentation Strategies From Data. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 113–123, June 2019. doi: 10.1109/CVPR.2019.00020.

Nuno César de Sá, Mitra Baratchi, Vincent Buitenhuis, Perry Cornelissen, and Peter van Bodegom. AutoML for Estimating Grass Height from ETM+/OLI Data from Field Measurements at a Nature Reserve. *GIScience & Remote Sensing*, 59(1):2164–2183, 2022. ISSN 1548-1603. doi: 10.1080/15481603.2022.2152304.

Nicoleta Darra, Borja Espejo-Garcia, Aikaterini Kasimati, Olga Kriezi, Emmanouil Psomiadis, and Spyros Fountas. Can Satellites Predict Yield? Ensemble Machine Learning and Statistical Analysis of Sentinel-2 Imagery for Processing Tomato Yield Prediction. *Sensors*, 23(5):2586, January 2023. ISSN 1424-8220. doi: 10.3390/s23052586.

Nicoleta Darra, Borja Espejo Garcia, Emmanouil Psomiadis, and Spyros Fountas. Optimizing Crop Yield Prediction: Inter-Dated Vegetation Indices & Automl Ensembles in Processing Tomato Crop. In *IGARSS 2024 - 2024 IEEE International Geoscience and Remote Sensing Symposium*, pages 4320–4324, Athens, Greece, July 2024. IEEE. doi: 10.1109/IGARSS53475.2024.10641757.

Nicoleta Darra, Borja Espejo-Garcia, Vassilis Psiroukis, Emmanouil Psomiadis, and Spyros Fountas. Spectral bands vs. vegetation indices: An AutoML approach for processing tomato yield predictions based on Sentinel-2 imagery. *Smart Agricultural Technology*, 10: 100805, March 2025. ISSN 2772-3755. doi: 10.1016/j.atech.2025.100805.

Ayan Das, Manoj K. Mishra, Somsubhra Chakraborty, Bimal K. Bhattacharya, Rucha Dave, Dileep Kumar, Khushvadan Patel, Raj Setia, and David C. Weindorf. Deep Carbon: A Multiscale Feature-Time Fusion Approach for Field Level Digital Soil Organic Carbon Mapping. *European Journal of Soil Science*, 76(4):e70161, 2025. ISSN 1365-2389. doi: 10.1111/ejss.70161.

M. C. M. de Paulo, J. N. Turnes, P. N. Happ, M. P. Ferreira, H. A. Marques, and R. Q. Feitosa. How Far Should I Look? A Neural Architecture Search Strategy for Semantic Segmentation of Remote Sensing Images. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, V-3-2022:17–23, May 2022. ISSN 2194-9042. doi: 10.5194/isprs-annals-V-3-2022-17-2022.

Orianna DeMasi, Konrad Kording, and Benjamin Recht. Meaningless comparisons lead to false optimism in medical machine learning. *PLoS ONE*, 12(9):e0184604, September 2017. ISSN 1932-6203. doi: 10.1371/journal.pone.0184604.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, Miami, Florida, USA, June 2009. Computer Vision Foundation / IEEE. doi: 10.1109/CVPR.2009.5206848.

Hongwei Dong, Bin Zou, Lamei Zhang, and Siyu Zhang. Automatic Design of CNNs via Differentiable Neural Architecture Search for PolSAR Image Classification. *IEEE Transactions on Geoscience and Remote Sensing*, 58(9):6362–6375, September 2020. ISSN 1558-0644. doi: 10.1109/TGRS.2020.2976694.

Xuanyi Dong and Yi Yang. Searching for a robust neural architecture in four GPU hours. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1761–1770, Long Beach, California, USA, June 2019a. Computer Vision Foundation / IEEE. doi: 10.1109/CVPR.2019.00186.

Xuanyi Dong and Yi Yang. Network pruning via transformable architecture search. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 759–770, Red Hook, NY, USA, 2019b. Curran Associates Inc.

Wentian Du, Jie Chen, Chaochen Zhang, Po Zhao, Huiyao Wan, Zheng Zhou, Yice Cao, Zhixiang Huang, Yingsong Li, and Bocai Wu. SARNas: A Hardware-Aware SAR Target Detection Algorithm via Multiobjective Neural Architecture Search. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–23, 2023. ISSN 1558-0644. doi: 10.1109/TGRS.2023.3292618.

Upol Ehsan, Samir Passi, Q. Vera Liao, Larry Chan, I-Hsiang Lee, Michael Muller, and Mark O Riedl. The Who in XAI: How AI Background Shapes Perceptions of AI Explanations. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–32, Honolulu HI USA, May 2024. ACM. doi: 10.1145/3613904.3642474.

Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural Architecture Search: A Survey. *Journal of Machine Learning Research*, 20:1–21, 2019.

Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander Smola. AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data. In *7th ICML Workshop on Automated Machine Learning (AutoML 2020)*, 2020.

Stefan Falkner, Aaron Klein, and Frank Hutter. BOHB: robust and efficient hyperparameter optimization at scale. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1436–1445, Stockholm, Sweden, 2018. PMLR.

Shiyang Feng, Zhaowei Li, Bo Zhang, Tao Chen, and Bin Wang. DSF2-NAS: Dual-Stage Feature Fusion via Network Architecture Search for Classification of Multimodal Remote Sensing Images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, pages 1–14, 2025. ISSN 2151-1535. doi: 10.1109/JSTARS.2025.3545831.

Matthias Feurer, Katharina Eggensperger, Stefan Falkner, Marius Lindauer, and Frank Hutter. Auto-sklearn 2.0: hands-free AutoML via meta-learning. *Journal of Machine Learning Research*, 23(1), January 2022. ISSN 1532-4435.

Casper Fibaek, Luke Camilleri, Andreas Luyts, Nikolaos Dionelis, and Bertrand Le Saux. PhilEO Bench: Evaluating Geo-Spatial Foundation Models. In *IGARSS 2024 - 2024 IEEE International Geoscience and Remote Sensing Symposium*, pages 2739–2744, Athens, Greece, July 2024. IEEE. doi: 10.1109/IGARSS53475.2024.10642780.

Alistair Michael Francis, John Mrziglod, Panagiotis Sidiropoulos, and Jan-Peter Muller. SEnSeI: A Deep Learning Module for Creating Sensor Independent Cloud Masks. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–21, 2022. ISSN 0196-2892, 1558-0644. doi: 10.1109/TGRS.2021.3128280.

Song Gao, Yingjie Hu, and Wenwen Li. *Handbook of Geospatial Artificial Intelligence*. CRC Press, Boca Raton, 1 edition, December 2023. ISBN 978-1-00-330842-3. doi: 10.1201/9781003308423.

Roman Garnett. *Bayesian Optimization*. Cambridge University Press, January 2023. ISBN 9781108425780. doi: 10.1017/9781108348973.

Caroline M. Gevaert. Explainable AI for earth observation: A review including societal and regulatory perspectives. *International Journal of Applied Earth Observation and Geoinformation*, 112:102869, August 2022. ISSN 15698432. doi: 10.1016/j.jag.2022.102869.

Pedram Ghamisi, Behnood Rasti, Naoto Yokoya, Qunming Wang, Bernhard Hofle, Lorenzo Bruzzone, Francesca Bovolo, Mingmin Chi, Katharina Anders, Richard Gloaguen, Peter M. Atkinson, and Jon Atli Benediktsson. Multisource and multitemporal data fusion in remote sensing: A comprehensive review of the state of the art. *IEEE Geoscience and Remote Sensing Magazine*, 7(1):6–39, March 2019. doi: 10.1109/MGRS.2018.2890023.

Pedram Ghamisi, Weikang Yu, Andrea Marinoni, Caroline M. Gevaert, Claudio Persello, Sivasakthy Selvakumaran, Manuela Girotto, Benjamin P. Horton, Philippe Rufin, Patrick Hostert, Fabio Pacifici, and Peter M. Atkinson. Responsible AI for Earth Observation. *arXiv preprint arXiv:2405.20868*, May 2024. doi: 10.48550/arXiv.2405.20868.

Gianluca Giuffrida, Luca Fanucci, Gabriele Meoni, Matej Batič, Léonie Buckley, Aubrey Dunne, Chris van Dijk, Marco Esposito, John Hefele, Nathan Vercruyssen, Gianluca Furano, Massimiliano Pastena, and Josef Aschbacher. The Φ-sat-1 mission: The first on-board deep neural network demonstrator for satellite earth observation. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2022. doi: 10.1109/TGRS.2021.3125567.

Povilas Gudzius, Olga Kurasova, Vytenis Darulis, and Ernestas Filatovas. AutoML-Based Neural Architecture Search for Object Recognition in Satellite Imagery. *Remote Sensing*, 15(1):91, 2023. ISSN 2072-4292. doi: 10.3390/RS15010091.

Xin Guo, Jiangwei Lao, Bo Dang, Yingying Zhang, Lei Yu, Lixiang Ru, Liheng Zhong, Ziyuan Huang, Kang Wu, Dingxiang Hu, Huimei He, Jian Wang, Jingdong Chen, Ming Yang, Yongjun Zhang, and Yansheng Li. SkySense: A Multi-Modal Remote Sensing Foundation Model Towards Universal Interpretation for Earth Observation Imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27672–27683, Seattle, Washington, USA, 2024. Computer Vision Foundation / IEEE. doi: 10.1109/CVPR52733.2024.02613.

Boran Han, Shuai Zhang, Xingjian Shi, and Markus Reichstein. Bridging Remote Sensors with Multisensor Geospatial Foundation Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27852–27862, Seattle, Washington, USA, 2024. Computer Vision Foundation /IEEE.

Wei Han, Xiaohan Zhang, Yi Wang, Lizhe Wang, Xiaohui Huang, Jun Li, Sheng Wang, Weitao Chen, Xianju Li, Ruyi Feng, Runyu Fan, Xinyu Zhang, and Yuewei Wang. A Survey of Machine Learning and Deep Learning in Remote Sensing of Geological Environment: Challenges, Advances, and Opportunities. *ISPRS Journal of Photogrammetry and Remote Sensing*, 202:87–113, August 2023. ISSN 0924-2716. doi: 10.1016/j.isprsjprs.2023.05.032.

Zhu Han, Danfeng Hong, Lianru Gao, Bing Zhang, Min Huang, and Jocelyn Chanussot. AutoNAS: Automatic Neural Architecture Search for Hyperspectral Unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2022. ISSN 1558-0644. doi: 10.1109/TGRS.2022.3186480.

Joëlle Hanna, Damian Borth, and Michael Mommert. Physics-Guided Multitask Learning for Estimating Power Generation and CO2 Emissions From Satellite Imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–12, 2023. ISSN 1558-0644. doi: 10.1109/TGRS.2023.3286444.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778, Las Vegas, Nevada, USA, 2016. IEEE Computer Society. doi: 10.1109/CVPR.2016.90.

Katharina Hechinger, Christoph Schweden, Xiao Xiang Zhu, and Göran Kauermann. Human-in-the-loop: Towards label embeddings for assessing classification difficulty. *Statistical Modelling*, page 1471082X251371796, October 2025. ISSN 1471-082X. doi: 10.1177/1471082X251371796.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12 (7):2217–2226, 2019. doi: 10.1109/JSTARS.2019.2918242.

Alexandre Heuillet, Ahmad Nasser, Hichem Arioui, and Hedi Tabia. Efficient Automation of Neural Network Design: A Survey on Differentiable Neural Architecture Search. *ACM Computing Surveys*, 56(11):1–36, June 2024. ISSN 0360-0300, 1557-7341. doi: 10.1145/3665138.

Thorsten Hoeser and Claudia Kuenzer. Object Detection and Image Segmentation with Deep Learning on Earth Observation Data: A Review-Part I: Evolution and Recent Trends. *Remote Sensing*, 12(10):1667, January 2020. ISSN 2072-4292. doi: 10.3390/rs12101667.

Thorsten Hoeser, Felix Bachofer, and Claudia Kuenzer. Object Detection and Image Segmentation with Deep Learning on Earth Observation Data: A Review—Part II: Applications. *Remote Sensing*, 12(18):3053, January 2020. ISSN 2072-4292. doi: 10.3390/rs12183053.

Adrian Höhl, Ivica Obadic, Miguel-Ángel Fernández-Torres, Dario Oliveira, and Xiao Xiang Zhu. Recent Trends, Challenges and Limitations of Explainable AI in Remote Sensing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8199–8205, Seattle, Washington, USA, 2024. Computer Vision Foundation / IEEE.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, January 2025. ISSN 1476-4687. doi: 10.1038/s41586-024-08328-6.

Laurel M. Hopkins, Weng-Keen Wong, Hannah Kerner, Fuxin Li, and Rebecca A. Hutchinson. Data Augmentation Approaches for Satellite Imagery. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(27):28097–28105, April 2025. ISSN 2374-3468. doi: 10.1609/aaai.v39i27.35028.

Md Ismail Hossen, Mohammad Awrangjeb, Shirui Pan, and Abdullah Al Mamun. Transfer learning in agriculture: A review. *Artificial Intelligence Review*, 58(4):97, January 2025. ISSN 1573-7462. doi: 10.1007/s10462-024-11081-x.

Jingliang Hu, Rong Liu, Danfeng Hong, Andrés Camero, Jing Yao, Mathias Schneider, Franz Kurz, Karl Segl, and Xiao Xiang Zhu. MDAS: a new multimodal benchmark dataset for remote sensing. *Earth System Science Data*, 15(1):113–131, January 2023. ISSN 1866-3508. doi: 10.5194/essd-15-113-2023.

Frank Hutter, Holger Hoos, and Kevin Leyton-Brown. An Efficient Approach for Assessing Hyperparameter Importance. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 754–762, Bejing, China, 22–24 Jun 2014. PMLR.

Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren, editors. *Automated Machine Learning: Methods, Systems, Challenges*. The Springer Series on Challenges in Machine Learning. Springer International Publishing, 2019. ISBN 9783030053185. doi: 10.1007/978-3-030-05318-5.

A. Inness, A.-M. Blechschmidt, I. Bouarar, S. Chabrillat, M. Crepulja, R. J. Engelen, H. Eskes, J. Flemming, A. Gaudel, F. Hendrick, V. Huijnen, L. Jones, J. Kapsomenakis, E. Katragkou, A. Keppens, B. Langerock, M. de Mazière, D. Melas, M. Parrington, V. H. Peuch, M. Razinger, A. Richter, M. G. Schultz, M. Suttie, V. Thouret, M. Vrekoussis, A. Wagner, and C. Zerefos. Data assimilation of satellite-retrieved ozone, carbon monoxide and nitrogen dioxide with ECMWF's Composition-IFS. *Atmospheric Chemistry and Physics*, 15(9):5275–5303, May 2015. ISSN 1680-7316. doi: 10.5194/acp-15-5275-2015.

Johannes Jakubik, Sujit Roy, C. E. Phillips, Paolo Fraccaro, Denys Godwin, Bianca Zadrozny, Daniela Szwarcman, Carlos Gomes, Gabby Nyirjesy, Blair Edwards, Daiki Kimura, Naomi Simumba, Linsong Chu, S. Karthik Mukkavilli, Devyani Lambhate, Kamal Das, Ranjini Bangalore, Dario Oliveira, Michal Muszynski, Kumar Ankur, Muthukumaran Ramasubramanian, Iksha Gurung, Sam Khallaghi, Hanxi (Steve) Li, Michael Cecil, Maryam Ahmadi, Fatemeh Kordi, Hamed Alemohammad, Manil Maskey, Raghu Ganti, Kommy Weldemariam, and Rahul Ramachandran. Foundation models for generalist geospatial artificial intelligence. *arXiv preprint arXiv:2310.18660*, 2023. doi: 10.48550/arXiv.2310.18660.

Marta Jemeljanova, Alexander Kmoch, and Evelyn Uuemaa. Adapting machine learning for environmental spatial data - A review. *Ecological Informatics*, 81:102634, July 2024. ISSN 1574-9541. doi: 10.1016/j.ecoinf.2024.102634.

Haifeng Jin, Qingquan Song, and Xia Hu. Auto-Keras: An Efficient Neural Architecture Search System. In *Proceedings of the 25th ACM SIGKDD International Conference on*

Knowledge Discovery & Data Mining, pages 1946–1956, Anchorage, AK, USA, 2019. ACM. doi: 10.1145/3292500.3330648.

Weipeng Jing, Quanlin Ren, Jun Zhou, and Houbing Song. AutoRSISC: Automatic design of neural architecture for remote sensing image scene classification. *Pattern Recognition Letters*, 140:186–192, December 2020. ISSN 0167-8655. doi: 10.1016/j.patrec.2020.09.034.

Moran Ju, Buniu Niu, Mulin Li, Tengkai Mao, and Si-nian Jin. Toward Better Accuracy-Efficiency Trade-Offs for Oriented SAR Ship Object Detection. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 18:9666–9681, 2025. ISSN 2151-1535. doi: 10.1109/JSTARS.2025.3555330.

Sayash Kapoor and Arvind Narayanan. Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, 4(9):100804, September 2023. ISSN 2666-3899. doi: 10.1016/j.patter.2023.100804.

Sayash Kapoor, Emily M. Cantrell, Kenny Peng, Thanh Hien Pham, Christopher A. Bail, Odd Erik Gundersen, Jake M. Hofman, Jessica Hullman, Michael A. Lones, Momin M. Malik, Priyanka Nanayakkara, Russell A. Poldrack, Inioluwa Deborah Raji, Michael Roberts, Matthew J. Salganik, Marta Serra-Garcia, Brandon M. Stewart, Gilles Vandewiele, and Arvind Narayanan. REFORMS: Consensus-based Recommendations for Machine-learning-based Science. *Science Advances*, 10(18):eadk3452, May 2024. doi: 10.1126/sciadv.adk3452.

N. Karasiak, J.-F. Dejoux, C. Monteil, and D. Sheeren. Spatial dependence between training and test sets: Another pitfall of classification accuracy assessment in remote sensing. *Machine Learning*, 111(7):2715–2740, July 2022. ISSN 1573-0565. doi: 10.1007/s10994-021-05972-1.

Anuj Karpatne, Imme Ebert-Uphoff, Sai Ravela, Hassan Ali Babaie, and Vipin Kumar. Machine Learning for the Geosciences: Challenges and Opportunities. *IEEE Transactions on Knowledge and Data Engineering*, 31(8):1544–1554, August 2019. ISSN 1558-2191. doi: 10.1109/TKDE.2018.2861006.

Aikaterini Kasimati, Borja Espejo-García, Nicoleta Darra, and Spyros Fountas. Predicting Grape Sugar Content under Quality Attributes Using Normalized Difference Vegetation Index Data and Automated Machine Learning. *Sensors*, 22(9):3249, January 2022. ISSN 1424-8220. doi: 10.3390/s22093249.

Peter Kedron, Wenwen Li, Stewart Fotheringham, and Michael Goodchild. Reproducibility and replicability: Opportunities and challenges for geospatial research. *International Journal of Geographical Information Science*, 35(3):427–445, March 2021. ISSN 1365-8816. doi: 10.1080/13658816.2020.1802032.

Ahmed Kheir, Vinay Nangia, Abdelrazek Elnashar, Mina Devakota, Mohie Omar, Til Feike, and Ajit Govind. Developing automated machine learning approach for fast and robust crop yield prediction using a fusion of remote sensing, soil, and weather dataset. *Environmental Research Communications*, 6:041005, 2024. ISSN 2515-7620. doi: 10.1088/2515-7620/ad2d02.

Zolo Kiala, Onisimo Mutanga, John Odindi, Kabir Y Peerbhay, and Rob Slotow. Automated classification of a tropical landscape infested by Parthenium weed (Parthenium hyterophorus). *International Journal of Remote Sensing*, 41(22):8497–8519, November 2020. ISSN 0143-1161. doi: 10.1080/01431161.2020.1779375.

Diego Kiedanski, Pablo Rodríguez-Bocca, and Gerardo Rubino. Seasonal Average Temperature Forecast with the AutoGluonTS Modern AutoML Tool. In *Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 458–468, Cham, 2025. Springer Nature Switzerland. doi: 10.1007/978-3-031-74633-8_34.

Dohyung Kim, Jahwan Koo, and Ung-Mo Kim. A survey on automated machine learning: Problems, methods and frameworks. In *Human-Computer Interaction. Theoretical Approaches and Design Methods*, pages 57–70, Cham, 2022. Springer International Publishing.

Joshua C.O. Koh, German Spangenberg, and Surya Kant. Automated Machine Learning for High-Throughput Image-Based Plant Phenotyping. *Remote Sensing 2021, Vol. 13, Page 858*, 13(5):858, February 2021. ISSN 2072-4292. doi: 10.3390/RS13050858.

Diana Koldasbayeva, Polina Tregubova, Mikhail Gasanov, Alexey Zaytsev, Anna Petrovskaia, and Evgeny Burnaev. Challenges in data-driven geospatial modeling for environmental research and practice. *Nature Communications*, 15(1):10700, December 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-55240-8.

Alex Krizhevsky. Learning multiple layers of features from tiny images, 2009. URL `https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf`.

Solomiia Kurchaba, Jasper van Vliet, Fons J. Verbeek, and Cor J. Veenman. Anomalous NO2 emitting ship detection with TROPOMI satellite data and machine learning. *Remote Sensing of Environment*, 297:113761, November 2023. ISSN 0034-4257. doi: 10.1016/j.rse.2023.113761.

J. Lai, Y. Zhang, A. Wang, W. Fei, Y. Diao, R. Li, and J. Wu. FLAML version 2.3.3 model-based assessment of gross primary productivity at forest, grassland, and cropland ecosystem sites. *Geoscientific Model Development*, 18(16):5115–5142, 2025. doi: 10.5194/gmd-18-5115-2025.

E. LeDell and S. Poirier. H2O AutoML: Scalable Automatic Machine Learning. In *7th ICML Workshop on Automated Machine Learning (AutoML 2020)*, 2020.

Doi Lee, Sanghun Son, Jaegu Bae, Soryeon Park, Jeongmin Seo, Dongju Seo, Yangwon Lee, and Jinsoo Kim. Single-Temporal Sentinel-2 for Analyzing Burned Area Detection Methods: A Study of 14 Cases in Republic of Korea Considering Land Cover. *Remote Sensing*, 16(5):884, January 2024. ISSN 2072-4292. doi: 10.3390/rs16050884.

Soo-Jin Lee, Eunha Sohn, Mija Kim, Ki-Hong Park, Kyungwon Park, and Yangwon Lee. Real-Time Retrieval of Daily Soil Moisture Using IMERG and GK2A Satellite Images with NWP and Topographic Data: A Machine Learning Approach for South Korea. *Remote Sensing*, 15(17):4168, January 2023. ISSN 2072-4292. doi: 10.3390/rs15174168.

Jiaxin Li, Danfeng Hong, Lianru Gao, Jing Yao, Ke Zheng, Bing Zhang, and Jocelyn Chanussot. Deep learning in multimodal remote sensing data fusion: A comprehensive review. *International Journal of Applied Earth Observation and Geoinformation*, 112: 102926, August 2022a. ISSN 1569-8432. doi: 10.1016/j.jag.2022.102926.

Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P. Trevino, Jiliang Tang, and Huan Liu. Feature Selection: A Data Perspective. *ACM Computing Surveys*, 50(6): 94:1–94:45, December 2017a. ISSN 0360-0300. doi: 10.1145/3136625.

Kai-Yun Li, Raul Sampaio de Lima, Niall G. Burnside, Ele Vahtmäe, Tiit Kutser, Karli Sepp, Victor Henrique Cabral Pinheiro, Ming-Der Yang, Ants Vain, and Kalev Sepp. Toward Automated Machine Learning-Based Hyperspectral Image Analysis in Crop Yield and Biomass Estimation. *Remote Sensing*, 14(5):1114, January 2022b. ISSN 2072-4292. doi: 10.3390/rs14051114.

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. *Journal of Machine Learning Research*, 18:6765–6816, 2017b.

Wenwen Li, Chia-Yu Hsu, Sizhe Wang, and Peter Kedron. GeoAI Reproducibility and Replicability: A Computational and Spatial Perspective. *Annals of the American Association of Geographers*, October 2024. ISSN 2469-4452.

Xiao Li, Lin Lei, and Gangyao Kuang. Multi-Modal Fusion Architecture Search for Land Cover Classification Using Heterogeneous Remote Sensing Images. In *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, pages 5997–6000, Brussels, Belgium, July 2021. IEEE. doi: 10.1109/IGARSS47720.2021.9555029.

Xiao Li, Lin Lei, Caiguang Zhang, and Gangyao Kuang. Multimodal Semantic Consistency-Based Fusion Architecture Search for Land Cover Classification. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2022c. ISSN 1558-0644. doi: 10.1109/TGRS.2022.3193273.

Z. Li, L. Chen, X. Huang, Z. Zhang, and H. Shi. Differentiable Neural Architecture Search for SAR Image Ship Object Detection. In *IET International Radar Conference (IET IRC 2020)*, volume 2020, pages 950–954, Online, November 2020. IEEE. doi: 10.1049/icp.2021.0794.

Peng Liang, Cheng-Zhi Qin, and A.-Xing Zhu. Using Automated Machine Learning for Spatial Prediction—The Heshan Soil Subgroups Case Study. *Land*, 13(4):551, April 2024. ISSN 2073-445X. doi: 10.3390/land13040551.

Wei Liang, Jihao Li, Wenhui Diao, Xian Sun, Kun Fu, and Yirong Wu. FGATR-Net: Automatic Network Architecture Design for Fine-Grained Aircraft Type Recognition in Remote Sensing Images. *Remote Sensing*, 12(24):4187, January 2020. ISSN 2072-4292. doi: 10.3390/rs12244187.

Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. In *Proceedings of the*

IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 936–944, Honolulu, Hawaii, USA, July 2017. Computer Vision Foundation/IEEE. doi: 10.1109/CVPR.2017.106.

Bochun Liu, Jinming Ge, Qingyu Mu, Chi Zhang, Xiaoyu Hu, Jiajing Du, Yanyan Wu, Bo Wang, Xiang Li, Jianping Huang, and Qiang Fu. Unbiased High-Precision Cloud Detection for Advanced Himawari Imager using Automatic Machine Learning. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, pages 1–18, 2025. ISSN 2151-1535. doi: 10.1109/JSTARS.2025.3535762.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. In *Proceedings of the 6th International Conference on Learning Representations, ICLR 2018*, Vancouver, Canada, 2018. OpenReview.net. URL `https://openreview.net/forum?id=S1eYHoC5FX`.

Hongying Liu, Derong Xu, Tianwen Zhu, Fanhua Shang, Yuanyuan Liu, Jianhua Lu, and Ri Yang. Graph Convolutional Networks by Architecture Search for PolSAR Image Classification. *Remote Sensing*, 13(7):1404, January 2021a. ISSN 2072-4292. doi: 10.3390/rs13071404.

Xiaobo Liu, Chaochao Zhang, Zhihua Cai, Jianfeng Yang, Zhilang Zhou, and Xin Gong. Continuous Particle Swarm Optimization-Based Deep Learning Architecture Search for Hyperspectral Image Classification. *Remote Sensing*, 13(6):1082, 2021b. doi: 10.3390/rs13061082.

Scott M Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the Effective Receptive Field in Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, volume 29, pages 4898–4906, Barcelona, Spain, 2016. Curran Associates, Inc.

Junwei Ma, Dongze Lei, Zhiyuan Ren, Chunhai Tan, Ding Xia, and Haixiang Guo. Automated Machine Learning-Based Landslide Susceptibility Mapping for the Three Gorges Reservoir Area, China. *Mathematical Geosciences*, 56:975–1010, November 2023. ISSN 1874-8953. doi: 10.1007/s11004-023-10116-3.

Gengchen Mai, Weiming Huang, Jin Sun, Suhang Song, Deepak Mishra, Ninghao Liu, Song Gao, Tianming Liu, Gao Cong, Yingjie Hu, Chris Cundy, Ziyuan Li, Rui Zhu, and Ni Lao. On the Opportunities and Challenges of Foundation Models for GeoAI (Vision Paper). *ACM Trans. Spatial Algorithms Syst.*, 10(2):11:1–11:46, July 2024. ISSN 2374-0353. doi: 10.1145/3653070.

Gengchen Mai, Yiqun Xie, Xiaowei Jia, Ni Lao, Jinmeng Rao, Qing Zhu, Zeping Liu, Yao-Yi Chiang, and Junfeng Jiao. Towards the next generation of Geospatial Artificial Intelligence. *International Journal of Applied Earth Observation and Geoinformation*, 136:104368, February 2025. ISSN 1569-8432. doi: 10.1016/j.jag.2025.104368.

Sowmya Mangalath Ravindran, Santosh Kumar Moorakkal Bhaskaran, Sooraj K. Ambat, Kannan Balakrishnan, and Manoj Manguttathil Gopalakrishnan. An automated machine learning methodology for the improved prediction of reference evapotranspiration using minimal input parameters. *Hydrological Processes*, 36(5):e14571, 2022. ISSN 1099-1085. doi: 10.1002/hyp.14571.

Valerio Marsocci, Yuru Jia, Georges Le Bellier, David Kerekes, Liang Zeng, Sebastian Hafner, Sebastian Gerard, Eric Brune, Ritu Yadav, Ali Shibli, Heng Fang, Yifang Ban, Maarten Vergauwen, Nicolas Audebert, and Andrea Nascetti. PANGAEA: A Global and Inclusive Benchmark for Geospatial Foundation Models. *arXiv preprint arXiv:2412.04204*, December 2024. doi: 10.48550/arXiv.2412.04204.

Romit Maulik, Romain Egele, Bethany Lusch, and Prasanna Balaprakash. Recurrent Neural Network Architecture Search for Geophysical Emulation. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, page 8, Online, November 2020. IEEE/ACM. doi: 10.1109/SC41405.2020.00012.

Aaron E. Maxwell, Timothy A. Warner, and Luis Andrés Guillén. Accuracy Assessment in Convolutional Neural Network-Based Deep Learning Remote Sensing Studies—Part 1: Literature Review. *Remote Sensing*, 13(13):2450, January 2021a. ISSN 2072-4292. doi: 10.3390/rs13132450.

Aaron E. Maxwell, Timothy A. Warner, and Luis Andrés Guillén. Accuracy Assessment in Convolutional Neural Network-Based Deep Learning Remote Sensing Studies—Part 2: Recommendations and Best Practices. *Remote Sensing*, 13(13):2591, January 2021b. ISSN 2072-4292. doi: 10.3390/rs13132591.

Aaron E. Maxwell, Michelle S. Bester, and Christopher A. Ramezan. Enhancing Reproducibility and Replicability in Remote Sensing Deep Learning Research and Practice. *Remote Sensing*, 14(22):5760, January 2022. ISSN 2072-4292. doi: 10.3390/rs14225760.

Amy McGovern, Imme Ebert-Uphoff, II Gagne, David John, and Ann Bostrom. Why we need to focus on developing ethical, responsible, and trustworthy artificial intelligence approaches for environmental science. *Environmental Data Science*, 1:e6, January 2022. ISSN 2634-4602. doi: 10.1017/eds.2022.5.

Amy McGovern, Ann Bostrom, Marie McGraw, Randy J. Chase, II Gagne, David John, Imme Ebert-Uphoff, Kate D. Musgrave, and Andrea Schumacher. Identifying and Categorizing Bias in AI/ML for Earth Sciences. *Bulletin of the American Meteorological Society*, March 2024. doi: 10.1175/BAMS-D-23-0196.1.

Nick McGreivy and Ammar Hakim. Weak Baselines and Reporting Biases Lead to Overoptimism in Machine Learning for Fluid-Related Partial Differential Equations. *Nature Machine Intelligence*, 6(10):1256–1269, October 2024. ISSN 2522-5839. doi: 10.1038/s42256-024-00897-5.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A Survey on Bias and Fairness in Machine Learning. *ACM Computing Surveys*, 54(6): 115:1–115:35, July 2021. ISSN 0360-0300. doi: 10.1145/3457607.

Saša Mladenovic, Marius Lindauer, and Carola Doerr. Automated Data Preparation for Machine Learning: A Survey. *Journal of Data-centric Machine Learning Research*, February 2026.

Rohit Mohan, Thomas Elsken, Arber Zela, Jan Hendrik Metzen, Benedikt Staffler, Thomas Brox, Abhinav Valada, and Frank Hutter. Neural Architecture Search for Dense Prediction Tasks in Computer Vision. *International Journal of Computer Vision*, 131(7): 1784–1807, July 2023. ISSN 1573-1405. doi: 10.1007/s11263-023-01785-y.

Yousef Mubarak and Ardiansyah Koeshidayatullah. Hierarchical automated machine learning (AutoML) for advanced unconventional reservoir characterization. *Scientific Reports*, 13(1):13812, August 2023. ISSN 2045-2322. doi: 10.1038/s41598-023-40904-0.

Samuel G. Müller and Frank Hutter. Trivialaugment: Tuning-free yet state-of-the-art data augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 774–782, October 2021.

Parth Naik, Michele Dalponte, and Lorenzo Bruzzone. Automated Machine Learning Driven Stacked Ensemble Modeling for Forest Aboveground Biomass Prediction Using Multitemporal Sentinel-2 Data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 16:3442–3454, 2023. ISSN 2151-1535. doi: 10.1109/JSTARS.2022.3232583.

Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing Properties of Vision Transformers. In *Advances in Neural Information Processing Systems*, volume 34, pages 23296–23308, Online, 2021. Curran Associates, Inc.

Nguyen Ngoc Tu, Haishen Lü, Wei He, Peipei Xu, Mengyao Zhao, Shuai Liu, Yonghua Zhu, and Xinhui Lei. Automated machine learning integrating multi-source satellite observations to predict gross and net CO2 fluxes of coastal wetlands in China. *Environmental Research Letters*, 20(8):084011, July 2025. ISSN 1748-9326. doi: 10.1088/1748-9326/ade731.

Luis Oala, Manil Maskey, Lilith Bat-Leah, Alicia Parrish, Nezihe Merve Gürel, Tzu-Sheng Kuo, Yang Liu, Rotem Dror, Danilo Brajovic, Xiaozhe Yao, Max Bartolo, William A Gaviria Rojas, Ryan Hileman, Rainier Aliment, Michael W. Mahoney, Meg Risdal, Matthew Lease, Wojciech Samek, Debojyoti Dutta, Curtis G Northcutt, Cody Coleman, Braden Hancock, Bernard Koch, Girmaw Abebe Tadesse, Bojan Karlaš, Ahmed Alaa, Adji Bousso Dieng, Natasha Noy, Vijay Janapa Reddi, James Zou, Praveen Paritosh, Mihaela van der Schaar, Kurt Bollacker, Lora Aroyo, Ce Zhang, Joaquin Vanschoren, Isabelle Guyon, and Peter Mattson. DMLR: Data-centric machine learning research - past, present and future. *Journal of Data-centric Machine Learning Research*, 1(5):1–27, 2024.

Mariana Oliveira, Luís Torgo, and Vítor Santos Costa. Evaluation Procedures for Forecasting with Spatiotemporal Data. *Mathematics*, 9(6):691, January 2021. ISSN 2227-7390. doi: 10.3390/math9060691.

Randal S. Olson, Nathan Bartley, Ryan J. Urbanowicz, and Jason H. Moore. Evaluation of a tree-based pipeline optimization tool for automating data science. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, GECCO '16, pages 485–492, New York, NY, USA, 2016. ACM. doi: 10.1145/2908812.2908918.

Nelly Rosaura Palacios Salinas, Mitra Baratchi, Jan N. van Rijn, and Andreas Vollrath. Automated Machine Learning for Satellite Data: Integrating Remote Sensing Pre-trained Models into AutoML Systems. In *Machine Learning and Knowledge Discovery in Databases. Applied Data Science Track*, volume 12979 of *Lecture Notes in Computer Science*, pages 447–462, Cham, 2021. Springer International Publishing. doi: 10.1007/978-3-030-86517-7_28.

Daniel Paluba, Bertrand Le Saux, Francesco Sarti, and Přemysl Štych. Estimating vegetation indices and biophysical parameters for Central European temperate forests with Sentinel-1 SAR data and machine learning. *Big Earth Data*, 9(2):155–186, 2025. ISSN 2096-4471. doi: 10.1080/20964471.2025.2459300.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Cheng Peng, Yangyang Li, Licheng Jiao, and Ronghua Shang. Efficient Convolutional Neural Architecture Search for Remote Sensing Image Scene Classification. *IEEE Transactions on Geoscience and Remote Sensing*, 59(7):6092–6105, July 2021. ISSN 0196-2892, 1558-0644. doi: 10.1109/TGRS.2020.3020424.

Cheng Peng, Yangyang Li, Ronghua Shang, and Licheng Jiao. RSBNet: One-shot neural architecture search for a backbone network in remote sensing image recognition. *Neurocomputing*, 537:110–127, June 2023. ISSN 0925-2312. doi: 10.1016/j.neucom.2023.03.046.

Iana S. Polonskaia, Ilya R. Aliev, and Nikolay O. Nikitin. Automated Evolutionary Design of CNN Classifiers for Object Recognition on Satellite Images. *Procedia Comput. Sci.*, 193(C):210–219, January 2021. ISSN 1877-0509. doi: 10.1016/j.procs.2021.10.021.

Zeeshan Ramzan, H. M. Shahzad Asif, Irfan Yousuf, and Muhammad Shahbaz. A Multimodal Data Fusion and Deep Neural Networks Based Technique for Tea Yield Estimation in Pakistan Using Satellite Imagery. *IEEE Access*, 11:42578–42594, 2023. ISSN 2169-3536. doi: 10.1109/ACCESS.2023.3271410.

Markus Reichstein, Gustau Camps-Valls, Bjorn Stevens, Martin Jung, Joachim Denzler, Nuno Carvalhais, and Prabhat. Deep learning and process understanding for data-driven Earth system science. *Nature 2019 566:7743*, 566(7743):195–204, February 2019. ISSN 1476-4687. doi: 10.1038/s41586-019-0912-1.

Diego Renza, Elsa Adriana Cárdenas, Carlos Marcelo Jaramillo, Serena Sarah Weber, and Estibaliz Martinez. Landslide Susceptibility Model by Means of Remote Sensing Images and AutoML. *Communications in Computer and Information Science*, 1431 CCIS:25–37, 2021. ISSN 18650937. doi: 10.1007/978-3-030-86702-7_3/FIGURES/4.

David R. Roberts, Volker Bahn, Simone Ciuti, Mark S. Boyce, Jane Elith, Gurutzeta Guillera-Arroita, Severin Hauenstein, José J. Lahoz-Monfort, Boris Schröder, Wilfried Thuiller, David I. Warton, Brendan A. Wintle, Florian Hartig, and Carsten F. Dormann. Cross-Validation Strategies for Data with Temporal, Spatial, Hierarchical, or Phylogenetic Structure. *Ecography*, 40(8):913–929, 2017. ISSN 1600-0587. doi: 10.1111/ecog.02881.

David Rolnick, Priya L. Donti, Lynn H. Kaack, Kelly Kochanski, Alexandre Lacoste, Kris Sankaran, Andrew Slavin Ross, Nikola Milojevic-Dupont, Natasha Jaques, Anna Waldman-Brown, Alexandra Sasha Luccioni, Tegan Maharaj, Evan D. Sherwin, S. Karthik Mukkavilli, Konrad P. Kording, Carla P. Gomes, Andrew Y. Ng, Demis Hassabis, John C. Platt, Felix Creutzig, Jennifer Chayes, and Yoshua Bengio. Tackling Climate Change with Machine Learning. *ACM Computing Surveys*, 55(2):42:1–42:96, February 2022. ISSN 0360-0300. doi: 10.1145/3485128.

Ribana Roscher, Marc Russwurm, Caroline Gevaert, Michael Kampffmeyer, Jefersson A. Dos Santos, Maria Vakalopoulou, Ronny Hänsch, Stine Hansen, Keiller Nogueira, Jonathan Prexl, and Devis Tuia. Better, Not Just More: Data-Centric Machine Learning for Earth Observation. *IEEE Geoscience and Remote Sensing Magazine*, 12(4):335–355, 2024. doi: 10.1109/MGRS.2024.3470986.

S. Salcedo-Sanz, P. Ghamisi, M. Piles, M. Werner, L. Cuadra, A. Moreno-Martínez, E. Izquierdo-Verdiguier, J. Muñoz-Marí, Amirhosein Mosavi, and G. Camps-Valls. Machine learning information fusion in Earth observation: A comprehensive review of methods, applications and data sources. *Information Fusion*, 63:256–272, November 2020. ISSN 1566-2535. doi: 10.1016/j.inffus.2020.07.004.

Imrus Salehin, Md. Shamiul Islam, Pritom Saha, S. M. Noman, Azra Tuni, Md. Mehedi Hasan, and Md. Abu Baten. AutoML: A systematic review on automated machine learning with neural architecture search. *Journal of Information and Intelligence*, 2(1):52–81, January 2024. ISSN 2949-7159. doi: 10.1016/j.jiixd.2023.10.002.

Sasan Salmani Pour Avval, Nathan D. Eskue, Roger M. Groves, and Vahid Yaghoubi. Systematic Review on Neural Architecture Search. *Artificial Intelligence Review*, 58(3): 73, January 2025. ISSN 1573-7462. doi: 10.1007/s10462-024-11058-w.

Jiří Šandera and Přemysl Štych. Mapping changes of grassland to arable land using automatic machine learning of stacked ensembles and H2O library. *European Journal of Remote Sensing*, 57(1):2294127, December 2024. ISSN null. doi: 10.1080/22797254.2023.2294127.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016. doi: 10.1109/JPROC.2015.2494218.

Junge Shen, Bin Cao, Chi Zhang, Ruxin Wang, and Qi Wang. Remote Sensing Scene Classification Based on Attention-Enabled Progressively Searching. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–13, 2022. ISSN 1558-0644. doi: 10.1109/TGRS.2022.3186588.

Abhilash Singh, Kumar Gaurav, Atul Kumar Rai, and Zafar Beg. Machine Learning to Estimate Surface Roughness from Satellite Images. *Remote Sensing*, 13(19):3794, January 2021. ISSN 2072-4292. doi: 10.3390/rs13193794.

Kenneth O. Stanley, Jeff Clune, Joel Lehman, and Risto Miikkulainen. Designing neural networks through neuroevolution. *Nature Machine Intelligence*, 1(1):24–35, January 2019. ISSN 2522-5839. doi: 10.1038/s42256-018-0006-z.

Ye Su, Longlong Zhao, Hongzhong Li, Xiaoli Li, Jinsong Chen, and Yuankai Ge. An Efficient Task Implementation Modeling Framework with Multi-Stage Feature Selection and AutoML: A Case Study in Forest Fire Risk Prediction. *Remote Sensing*, 16(17):3190, January 2024. ISSN 2072-4292. doi: 10.3390/rs16173190.

Gencer Sumbul, Marcela Charfuelan, Begüm Demir, and Volker Markl. Bigearthnet: A Large-Scale Benchmark Archive for Remote Sensing Image Understanding. In *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 5901–5904, July 2019. doi: 10.1109/IGARSS.2019.8900532.

Alexander Y. Sun, Bridget R. Scanlon, Himanshu Save, and Ashraf Rateb. Reconstruction of GRACE Total Water Storage Through Automated Machine Learning. *Water Resources Research*, 57(2):e2020WR028666, 2021. ISSN 1944-7973. doi: 10.1029/2020WR028666.

El-Ghazali Talbi. Automated Design of Deep Neural Networks: A Survey and Unified Taxonomy. *ACM Computing Surveys*, 54(2), March 2021. ISSN 0360-0300. doi: 10.1145/3439730.

Guixi Tang, Zhice Fang, and Yi Wang. Global Landslide Susceptibility Prediction Based on the Automated Machine Learning (AutoML) Framework. *Geocarto International*, 38(1):2236576, December 2023. ISSN 1010-6049. doi: 10.1080/10106049.2023.2236576.

Zhiqiang Tang, Haoyang Fang, Su Zhou, Taojiannan Yang, Zihan Zhong, Cuixiong Hu, Katrin Kirchhoff, and George Karypis. Autogluon-multimodal (automm): Supercharging multimodal automl with foundation models. In *International Conference on Automated Machine Learning*, volume 256 of *Proceedings of Machine Learning Research*, pages 15/1–35, Paris, France, 2024. PMLR.

Chris Thornton, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. Auto-WEKA: combined selection and hyperparameter optimization of classification algorithms. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, page 847–855, New York, NY, USA, 2013. Association for Computing Machinery. doi: 10.1145/2487575.2487629.

Rene Traore, Andrés Camero, and Xiaoxiang Zhu. Compact Neural Architecture Search for Local Climate Zones Classification. In *ESANN 2021 Proceedings*, pages 393–398, Online event (Bruges, Belgium), 2021. Ciaco - i6doc.com. doi: 10.14428/esann/2021.ES2021-55.

Mark J. van der Laan, Eric C Polley, and Alan E. Hubbard. Super Learner. *Statistical Applications in Genetics and Molecular Biology*, 6(1):20, 2007. doi: 10.2202/1544-6115.1309.

Jesper E. Van Engelen and Holger H. Hoos. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440, February 2020. ISSN 0885-6125, 1573-0565. doi: 10.1007/s10994-019-05855-6.

Joaquin Vanschoren. Meta-Learning. In *Automated Machine Learning: Methods, Systems, Challenges*, pages 35–61. Springer International Publishing, Cham, 2019. ISBN 978-3-030-05318-5. doi: 10.1007/978-3-030-05318-5_2.

Jorge Vicent Servera, Luca Martino, Jochem Verrelst, and Gustau Camps-Valls. Multifidelity Gaussian Process Emulation for Atmospheric Radiative Transfer Models. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–10, 2023. doi: 10.1109/TGRS.2023.3300460.

Yuting Wan, Yanfei Zhong, Ailong Ma, Junjue Wang, and Liangpei Zhang. E2SCNet: Efficient Multiobjective Evolutionary Automatic Search for Remote Sensing Image Scene Classification Network Architecture. *IEEE Transactions on Neural Networks and Learning Systems*, 35:1–15, 2022. ISSN 2162-2388. doi: 10.1109/TNNLS.2022.3220699.

Aili Wang, Dong Xue, Haibin Wu, and Yanfeng Gu. Efficient Convolutional Neural Architecture Search for LiDAR DSM Classification. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–17, 2022a. ISSN 1558-0644. doi: 10.1109/TGRS.2022.3171520.

Chi Wang, Qingyun Wu, Markus Weimer, and Erkang Zhu. FLAML: A Fast and Lightweight AutoML Library. In *Proceedings of Machine Learning and Systems*, volume 3, pages 434–447, 2021a.

Di Wang, Jing Zhang, Bo Du, Gui-Song Xia, and Dacheng Tao. An Empirical Study of Remote Sensing Pretraining. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–20, 2023a. ISSN 1558-0644. doi: 10.1109/TGRS.2022.3176603.

Hao Wang, Dezhi Han, Zhongdai Wu, Junxiang Wang, Yuan Fan, and Yachao Zhou. NAS-YOLOX: Ship detection based on improved YOLOX for SAR imagery. In *2023 IEEE 10th International Conference on Cyber Security and Cloud Computing (CSCloud)/2023 IEEE 9th International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, pages 126–131, Xiangtan, China, July 2023b. IEEE. doi: 10.1109/CSCloud-EdgeCom58631.2023.00030.

Junjue Wang, Yanfei Zhong, Zhuo Zheng, Ailong Ma, and Liangpei Zhang. RSNet: The Search for Remote Sensing Deep Neural Networks in Recognition Tasks. *IEEE Transactions on Geoscience and Remote Sensing*, 59(3):2520–2534, March 2021b. ISSN 0196-2892, 1558-0644. doi: 10.1109/TGRS.2020.3001401.

Junjue Wang, Yanfei Zhong, Ailong Ma, Zhuo Zheng, Yuting Wan, and Liangpei Zhang. LoveNAS: Towards multi-scene land-cover mapping via hierarchical searching adaptive network. *ISPRS Journal of Photogrammetry and Remote Sensing*, 209:265–278, March 2024a. ISSN 0924-2716. doi: 10.1016/j.isprsjprs.2024.01.011.

Xilu Wang, Yaochu Jin, Sebastian Schmitt, and Markus Olhofer. Recent Advances in Bayesian Optimization. *ACM Computing Surveys*, 55(13s), 2023c. ISSN 0360-0300. doi: 10.1145/3582078.

Yi Wang, Zhitong Xiong, Chenying Liu, Adam J. Stewart, Thomas Dujardin, Nikolaos Ioannis Bountos, Angelos Zavras, Franziska Gerken, Ioannis Papoutsis, Laura Leal-Taixé, and Xiao Xiang Zhu. Towards a unified copernicus foundation model for earth vision. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9888–9899, 2025. doi: 10.1109/ICCV51701.2025.00922.

Yu Wang, Yansheng Li, Wei Chen, Yunzhou Li, and Bo Dang. DNAS: Decoupling Neural Architecture Search for High-Resolution Remote Sensing Image Semantic Segmentation. *Remote Sensing*, 14(16):3864, January 2022b. ISSN 2072-4292. doi: 10.3390/rs14163864.

Zitao Wang, Jianping Wang, and Mengling Li. Spatial predictions of groundwater potential using automated machine learning (AutoML): A comparative study of feature selection and training sample size in Qinghai Province, China. *Environmental Science and Pollution Research*, 31(1):1127–1145, January 2024b. ISSN 1614-7499. doi: 10.1007/s11356-023-31262-5.

Julia Wąsala, Suzanne Marselis, Laurens Arp, Holger Hoos, Nicolas Longépé, and Mitra Baratchi. AutoSR4EO: An AutoML Approach to Super-Resolution for Earth Observation Images. *Remote Sensing*, 16(3):443, January 2024. ISSN 2072-4292. doi: 10.3390/rs16030443.

Julia Wąsala, Joannes D. Maasakkers, Berend J. Schuit, Gijs Leguijt, Ilse Aben, Rochelle Schneider, Holger Hoos, and Mitra Baratchi. AutoMergeNet: AutoML-based M-Source Satellite Data Fusion Evaluated with Atmospheric Case Studies. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2025. doi: 10.1109/JSTARS.2025.3621068.

Julia Wąsala, Joannes D. Maasakkers, Ilse Aben, Rochelle Schneider, Holger Hoos, and Mitra Baratchi. Mitigating representation bias caused by missing pixels in methane plume detection. In *Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 325–334, Cham, 2026. Springer Nature Switzerland.

Richard Wen and Songnian Li. Spatial Decision Support Systems with Automated Machine Learning: A Review. *ISPRS International Journal of Geo-Information*, 12(1):12, December 2022. ISSN 2220-9964. doi: 10.3390/ijgi12010012.

Colin White, Mahmoud Safari, Rhea Sukthanker, Binxin Ru, Thomas Elsken, Arber Zela, Debadeepta Dey, and Frank Hutter. Neural Architecture Search: Insights from 1000 Papers. *arXiv preprint arXiv:2301.08727*, January 2023. doi: 10.48550/arXiv.2301.08727.

David H. Wolpert. Stacked Generalization. *Neural Networks*, 5(2):241–259, January 1992. ISSN 0893-6080. doi: 10.1016/S0893-6080(05)80023-1.

D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997. doi: 10.1109/4235.585893.

Xinran Xia, Disong Fu, Wei Shao, Rubin Jiang, Shengli Wu, Peng Zhang, Dazhi Yang, and Xiangao Xia. Retrieving Precipitable Water Vapor Over Land From Satellite Passive Microwave Radiometer Measurements Using Automated Machine Learning. *Geo-*

physical Research Letters, 50(22):e2023GL105197, 2023. ISSN 1944-8007. doi: 10.1029/2023GL105197.

Fen Xiao, Han Xiang, Chunhong Cao, and Xieping Gao. Neural Architecture Search-Based Few-Shot Learning for Hyperspectral Image Classification. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–15, 2024. ISSN 1558-0644. doi: 10.1109/TGRS.2024.3385478.

Wei Xiong, Zhenyu Xiong, and Yaqi Cui. A Confounder-Free Fusion Network for Aerial Image Scene Feature Representation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15:5440–5454, 2022. ISSN 2151-1535. doi: 10.1109/JSTARS.2022.3189052.

Yuhui Xu, Lingxi Xie, Xiaopeng Zhang, Xin Chen, Guo-Jun Qi, Qi Tian, and Hongkai Xiong. PC-DARTS: Partial Channel Connections for Memory-Efficient Architecture Search. In *Proceedings of the 8th International Conference on Learning Representations, ICLR 2020*, Addis Ababa, Ethiopia, 2020. OpenReview.net. URL `https://openreview.net/forum?id=BJlS634tPr&noteId=BJlS634tPr`.

Minglong Xue, Jian Li, Zheng Zhao, and Qingli Luo. SAR2HEIGHT: Height Estimation from a Single SAR Image in Mountain Areas via Sparse Height and Proxyless Depth-Aware Penalty Neural Architecture Search for Unet. *Remote Sensing*, 14(21):5392, January 2022a. ISSN 2072-4292. doi: 10.3390/rs14215392.

Xizhe Xue, Haokui Zhang, Bei Fang, Zongwen Bai, and Ying Li. Grafting Transformer on Automatically Designed Convolutional Neural Network for Hyperspectral Image Classification. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–16, 2022b. ISSN 1558-0644. doi: 10.1109/TGRS.2022.3180685.

Chris Ying, Aaron Klein, Eric Christiansen, Esteban Real, Kevin Murphy, and Frank Hutter. NAS-bench-101: Towards reproducible neural architecture search. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7105–7114, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Manzhu Yu, Shiyan Zhang, Kai Zhang, Junjun Yin, Matthew Varela, and Jiheng Miao. Developing high-resolution PM2.5 exposure models by integrating low-cost sensors, automated machine learning, and big human mobility data. *Frontiers in Environmental Science*, 11, 2023. ISSN 2296-665X.

Khairul Adib Yusof, Syamsiah Mashohor, Mardina Abdullah, Mohd Amiruddin Abd Rahman, Nurul Shazana Abdul Hamid, Kasyful Qaedi, Khamirul Amin Matori, and Masashi Hayakawa. Earthquake Prediction Model Based on Geomagnetic Field Data Using Automated Machine Learning. *IEEE Geoscience and Remote Sensing Letters*, 21:1–1, 2024. ISSN 1558-0571. doi: 10.1109/LGRS.2024.3354954.

Daochen Zha, Zaid Pervaiz Bhat, Kwei-Herng Lai, Fan Yang, Zhimeng Jiang, Shaochen Zhong, and Xia Hu. Data-centric Artificial Intelligence: A Survey. *ACM Computing Surveys*, 57(5):129:1–129:42, January 2025. ISSN 0360-0300. doi: 10.1145/3711118.

Bo Zhang, Li Zhang, Yong Pang, Peter North, Min Yan, Hongge Ren, Linlin Ruan, Zhenyu Yang, and Bowei Chen. Improved Forest Signal Detection for Space-Borne Photon-Counting LiDAR Using Automatic Machine Learning. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 17:1–13, 2024. ISSN 2151-1535. doi: 10.1109/JSTARS.2023.3290680.

Bo Zhang, Li Zhang, Min Yan, Jian Zuo, Yuqi Dong, and Bowei Chen. High-resolution mapping of forest parameters in tropical rainforests through AutoML integration of GEDI with Sentinel-1/2, Landsat 8 and ALOS-2 data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 18:9084–9118, 2025. ISSN 2151-1535. doi: 10.1109/JSTARS.2025.3550878.

Haokui Zhang, Chengrong Gong, Yunpeng Bai, Zongwen Bai, and Ying Li. 3-D-ANAS: 3-D Asymmetric Neural Architecture Search for Fast Hyperspectral Image Classification. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–19, 2022a. ISSN 1558-0644. doi: 10.1109/TGRS.2021.3079123.

Mingwei Zhang, Weipeng Jing, Jingbo Lin, Nengzhen Fang, Wei Wei, Marcin Woźniak, and Robertas Damaševičius. NAS-HRIS: Automatic Design and Architecture Search of Neural Network for Semantic Segmentation in Remote Sensing Images. *Sensors*, 20(18): 5292, January 2020. ISSN 1424-8220. doi: 10.3390/s20185292.

Peng Zhang, Jinsong Tang, Heping Zhong, Mingqiang Ning, Dandan Liu, and Ke Wu. Self-Trained Target Detection of Radar and Sonar Images Using Automatic Deep Learning. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2022b. ISSN 1558-0644. doi: 10.1109/TGRS.2021.3096011.

Qi Zhang, Yuanxi Peng, Zhiwen Zhang, and Teng Li. Semantic Segmentation of Spectral LiDAR Point Clouds Based on Neural Architecture Search. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–11, 2023. ISSN 1558-0644. doi: 10.1109/TGRS.2023.3284995.

Siwei Zhang and Jun Ma. CascadeDumpNet: Enhancing open dumpsite detection through deep learning and AutoML integrated dual-stage approach using high-resolution satellite imagery. *Remote Sensing of Environment*, 313:114349, November 2024. ISSN 0034-4257. doi: 10.1016/j.rse.2024.114349.

Tianxiang Zhang, Jinya Su, Zhiyong Xu, Yulin Luo, and Jiangyun Li. Sentinel-2 Satellite Imagery for Urban Land Cover Classification by Optimized Random Forest Classifier. *Applied Sciences*, 11(2):543, January 2021. ISSN 2076-3417. doi: 10.3390/app11020543.

Xinbang Zhang, Qizhao Jin, Shiming Xiang, and Chunhong Pan. MFNet: The Spatio-Temporal Network for Meteorological Forecasting With Architecture Search. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2022c. ISSN 1558-0571. doi: 10.1109/LGRS.2022.3213618.

Xinbang Zhang, Qizhao Jin, Shiming Xiang, and Chunhong Pan. AutoMF: Spatio-temporal Architecture Search for The Meteorological Forecasting Task. In *2022 26th International*

Conference on Pattern Recognition (ICPR), pages 4708–4714, Montréal, Canada, August 2022d. IEEE. doi: 10.1109/ICPR56361.2022.9956271.

Zhen Zhang, Shanghao Liu, Yang Zhang, and Wenbo Chen. RS-DARTS: A Convolutional Neural Architecture Search for Remote Sensing Image Scene Classification. *Remote Sensing*, 14(1):141, 2022e. ISSN 2072-4292. doi: 10.3390/rs14010141.

Zhonghua Zheng, Arlene M. Fiore, Daniel M. Westervelt, George P. Milly, Jeff Goldsmith, Alexandra Karambelas, Gabriele Curci, Cynthia A. Randles, Antonio R. Paiva, Chi Wang, Qingyun Wu, and Sagnik Dey. Automated Machine Learning to Evaluate the Information Content of Tropospheric Trace Gas Columns for Fine Particle Estimates Over India: A Modeling Testbed. *Journal of Advances in Modeling Earth Systems*, 15 (3):e2022MS003099, 2023. ISSN 1942-2466. doi: 10.1029/2022MS003099.

Zilong Zhong, Ying Li, Lingfei Ma, Jonathan Li, and Wei-Shi Zheng. Spectral–Spatial Transformer Network for Hyperspectral Image Classification: A Factorized Architecture Search Framework. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–15, 2022. ISSN 1558-0644. doi: 10.1109/TGRS.2021.3115699.

Hairui Zhu, Shanhong Guo, Weixing Sheng, and Lei Xiao. SBNN: A Searched Binary Neural Network for SAR Ship Classification. *Applied Sciences*, 12(14):6866, January 2022. ISSN 2076-3417. doi: 10.3390/app12146866.

Lucas Zimmer, Marius Lindauer, and Frank Hutter. Auto-Pytorch: Multi-Fidelity MetaLearning for Efficient and Robust AutoDL. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(9):3079–3090, September 2021. ISSN 0162-8828, 2160-9292, 1939-3539. doi: 10.1109/TPAMI.2021.3067763.

Marc-André Zöller and Marco F. Huber. Benchmark and Survey of Automated Machine Learning Frameworks. *Journal of Artificial Intelligence Research*, 70:409–472, January 2021. ISSN 1076-9757. doi: 10.1613/jair.1.11854.

Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8697–8710, salt Lake City, Utah, USA, 2018. Computer Vision Foundation/IEEE. doi: 10.1109/CVPR.2018.00907.

Appendix A. Applications of AutoML frameworks

Table 4 gives an overview of which works have used which frameworks in their research.

Framework	Ref	Applications
H2O AutoML	LeDell and Poirier, 2020	Adams et al., 2020; Babaeian et al., 2021; Bodini, 2024; Chen et al., 2023; Kheir et al., 2024; Lee et al., 2023; Ma et al., 2023; Mangalath Ravindran et al., 2022; Mubarak and Koeshidayatullah, 2023; Naik et al., 2023; Šandera and Štych, 2024; Sun et al., 2021; Zhang et al., 2024, 2025; Ngoc Tu et al., 2025; Das et al., 2025
Auto-Sklearn	Feurer et al., 2022	César de Sá et al., 2022; Darra et al., 2023, 2024, 2025; Li et al., 2022b; Kasimati et al., 2022; Paluba et al., 2025; Arp et al., 2024; Bodini, 2024; Yu et al., 2023; Su et al., 2024; Liang et al., 2024
AutoKeras	Jin et al., 2019	Palacios Salinas et al., 2021; Koh et al., 2021; Wąsala et al., 2024; Bodini, 2024; Renza et al., 2021
TPOT	Olson et al., 2016	Kurchaba et al., 2023; Tang et al., 2023; Kiala et al., 2020; Choi et al., 2025; Bodini, 2024
FLAML	Wang et al., 2021a	Xia et al., 2023; Zheng et al., 2023; Lai et al., 2025; Liu et al., 2025; Wang et al., 2024b; Ngoc Tu et al., 2025
AutoGluon-Tabular & Time-series	Erickson et al., 2020; Tang et al., 2024	César de Sá et al., 2022; Kiedanski et al., 2025; Bai et al., 2022
MATLAB	`fitrauto`	Yusof et al., 2024; Singh et al., 2021; Zhang et al., 2021
PyCaret	Ali, 2020	Bruzón et al., 2021; Lee et al., 2024; Das et al., 2025
MLBox	Axel de Romblay, 2019	Bodini, 2024
Auto-PyTorch	Zimmer et al., 2021	Polonskaia et al., 2021; Tang et al., 2023; Zhang and Ma, 2024
DeepHyper	Balaprakash et al., 2018	Maulik et al., 2020

Table 4: AutoML frameworks used for EO applications.

Appendix B. Explainability functions implemented in AutoML frameworks

Table 5 gives an overview of explainability functions directly implemented in the AutoML frameworks applied to EO problems (at the time of writing). When not directly available, users can often access explainability functions through other Python packages, such as `sklearn`.

Framework	Feature importance	SHAP	Confusion matrix	Partial dependence	Inter-model correlation
H2O AutoML LeDell and Poirier, 2020	✓	✓	✓	✓	✓
Auto-sklearn Feurer et al., 2022	✗	✗	✗	✗	✗
AutoGluon-Tabular Erickson et al., 2020	✓	✓	✗	✗	✗
Auto-PyTorch Zimmer et al., 2021	✗	✗	✗	✗	✗
AutoKeras Jin et al., 2019	✗	✗	✗	✗	✗
PyCaret Ali, 2020	✓	✓	✗	✗	✗
MATLAB `fitrauto`	✗	✗	✗	✗	✗
TPOT Olson et al., 2016	✓	✗	✗	✗	✗
DeepHyper Balaprakash et al., 2018	✓*	✗	✗	✗	✗

Table 5: AutoML frameworks have limited post-hoc explainability functionality. H2O AutoML has the most: feature importance, SHapley Additive exPlanations (SHAP, Lundberg and Lee, 2017), confusion matrices, partial dependence plots, and inter-model correlation. *Only for some models.