Automated Machine Learning: Foundations, Advancements, Applications, and Future Directions

SurveyForge

Abstract— Automated Machine Learning (AutoML) stands as a transformative force in machine learning by automating the complex processes of model development, deployment, and optimization. This comprehensive survey explores AutoML's scope, detailing methodologies in hyperparameter tuning, neural architecture search, and pipeline optimization, highlighting advancements in interpretability, fairness, and resource-aware automation. Pivotal advances include the integration of meta-learning and transfer learning to expedite adaptation across tasks, as well as leveraging foundation models like large language models for enhanced semantic analysis. While AutoML offers improved accessibility, challenges such as scalability, computational efficiency, and bias remain critical. Recent trends emphasize Green AutoML's commitment to sustainability and ethical standards, and the expansion of multimodal capabilities to integrate diverse data types is underway. The survey concludes that future research should focus on refining scalability, enhancing interpretability, and formalizing ethical frameworks to ensure that AutoML evolves into an equitable and responsible technology that empowers human expertise while maintaining societal trust.

Index Terms—Automated Machine Learning, Neural Architecture Search, Multimodal AutoML

1 Introduction

Automated Machine Learning (AutoML) has emerged as one of the most transformative fields within artificial intelligence, aiming to democratize machine learning by automating the end-to-end process of building, optimizing, and deploying predictive models. This subsection introduces the foundational understanding of AutoML, tracing its historical evolution, exploring key motivations, and situating its advancements within the broader machine learning landscape. It also outlines the objectives and scope of this survey while emphasizing the importance of AutoML as a unifying framework for fostering innovation in both theoretical and applied domains.

The concept of automating aspects of machine learning has seen significant advancements since its inception. Early efforts sought to alleviate the manual burden of hyperparameter tuning and algorithm selection, a process widely considered a bottleneck in traditional machine learning workflows. These foundational ideas were materialized in tools like Auto-WEKA, which introduced the Combined Algorithm Selection and Hyperparameter Optimization (CASH) framework, leveraging Bayesian optimization to explore algorithm-hyperparameter configurations jointly [1]. Over the last decade, this vision has expanded to encompass the entire machine learning pipeline, encompassing automated preprocessing, feature engineering, model selection, hyperparameter optimization, and evaluation strategies [2], [3]. Notably, recent advancements, such as neural architecture search (NAS), have underscored the potential of AutoML to automate deep learning system design, eliminating the need for manual architectural engineering [4], [5].

The motivation behind AutoML stems from the increasing volume and complexity of machine learning tasks in a data-rich world where domain expertise is not always readily available. With organizations aspiring to derive actionable insights from big data, the demand for tools that lower the barrier to entry for non-experts has intensified [3]. AutoML frameworks address this need, enabling domain specialists to utilize machine learning without requiring deep statistical knowledge [2]. For instance, in time-critical industrial settings, AutoML systems have demonstrated exceptional utility by rapidly producing robust pipelines that dynamically adapt to real-world constraints, as shown in domains like predictive maintenance and dynamic advertising systems [6], [7]. Furthermore, achieving scalability and reducing computational costs remain pivotal, especially as AutoML is increasingly employed in resource-constrained environments such as edge computing and mobile applications [8].

While AutoML's agenda aligns with advancing machine learning by rendering it faster, simpler, and more scalable, its significance transcends operational efficiency. AutoML serves as a complementary paradigm to human-driven machine learning, augmenting human ingenuity by automating repetitive, computation-heavy tasks [9]. Certain areas, however, continue to require thoughtful human oversight, such as designing optimization strategies for fairness and incorporating domain-specific biases to ensure equitable decision-making [10]. This symbiosis exemplifies the duality of AutoML as both a catalyst for innovation and a tool for bridging the gap between human interpretability and algorithmic precision.

This comprehensive survey delves into the core methodologies, applications, and challenges of AutoML, guided by four key objectives: to consolidate existing knowledge, identify gaps in automation pipelines, evaluate emerging techniques, and propose future research directions. Topics explored include the automation of distinct pipeline components such as feature engineering and hyperparameter tuning [11], [12], as well as recent synergies with large language models to enhance feature engineering and semantic awareness of pipelines [13], [13]. Additionally, we critically assess ethical and environmental concerns, particularly in scalability and resource sustainability, to establish a balanced perspective on AutoML's broader impact [14].

The structure of this paper reflects these interwoven objectives. Following this introduction, the survey systematically examines AutoML's core components and workflow, detailing advances in data preprocessing, model selection, hyperparameter tuning, and evaluation. It then transitions to discuss innovative methodologies, practical applications across diverse domains, and the ethical challenges intrinsic to automation. Finally, the survey evaluates emerging research trends, emphasizing fields such as multimodal AutoML, few-shot learning, and lifelong model adaptation [7].

As AutoML continues to expand its boundaries—bringing together advanced neural architecture search, automated pipeline optimization, and scalable computation frameworks—it is evolving into a cornerstone of machine learning innovation. This progress highlights the exciting possibilities and persistent challenges of designing robust, trustworthy AI systems. Ultimately, advancing AutoML requires interdisciplinary collaboration, sustained research into optimization algorithms, and a strong commitment to embedding fairness, interpretability, and sustainability into automated workflows.

2 Core Components and Workflow of Automated Machine Learning

2.1 Data Preprocessing Automation

Data preprocessing is a foundational step in the machine learning pipeline, directly influencing the performance, robustness, and generalizability of downstream models. Automating data preprocessing tasks is a central challenge in Automated Machine Learning (AutoML) systems, as these tasks encompass a wide range of operations, including handling missing data, detecting and mitigating outliers, encoding categorical variables, and ensuring overall dataset quality and consistency. This subsection explores current automated techniques, highlights their methodological advancements, evaluates practical trade-offs, and discusses challenges and future directions for data preprocessing automation.

A critical component of automation in data preprocessing is the imputation of missing data, an issue prevalent in real-world datasets. Traditional methods such as mean, median, or mode imputation are computationally inexpensive but fail to capture complex data dependencies. Automated approaches now increasingly draw on machine learning algorithms and probabilistic models to enhance accuracy and robustness. Techniques such as k-nearest neighbors (kNN) imputation, iterative imputations using chained equations, and deep generative models like variational autoencoders (VAEs) have been integrated into automated systems for better predictions of missing values [15]. Moreover, multi-fidelity imputation frameworks, which tailor the imputation strategy based on the reliability and density of available data, are emerging as promising developments. The flexibility of such frameworks enables AutoML systems to adaptively choose methods based on dataset-specific characteristics and task requirements, but these often entail computational overhead compared to simpler statistical methods.

Outlier detection and treatment are similarly critical for ensuring data integrity. Automated tools now leverage advanced methodologies like isolation forests, which isolate data points with anomalous behavior through recursive partitioning, and robust clustering methods that identify outliers based on distance metrics and density estimations [16]. A notable advancement in this domain is the incorporation of deep learning techniques, such as autoencoder-based anomaly detection, which is particularly effective for high-dimensional data. However, while automated outlier detection algorithms often demonstrate high sensitivity, they sometimes struggle with interpretability and may inadvertently remove critical domain-specific signals. This underscores the importance of incorporating domain knowledge to strike a balance between exclusion of irrelevant noise and retention of informative, potentially extreme, values.

The transformation of categorical data into numerical representations is another crucial preprocessing task, traditionally achieved using one-hot encoding or label encoding. Automated systems now include advanced techniques such as target encoding, frequency encoding, and high-cardinality encoding methods like embeddings derived from language models [12]. These approaches are particularly beneficial in handling large-scale categoricals with numerous distinct levels, reducing memory consumption while improving representation fidelity. However, challenges remain, especially in ensuring robustness against data leakage during target encoding or when auto-generated embeddings depend heavily on labeled data.

General data cleaning and quality assessment is a broader aspect supervised by AutoML systems. Automated pipelines increasingly detect and resolve inconsistencies in data types, redundant entries, and duplicate records while assessing overall data quality through statistical and heuristic metrics [17], [18]. Sophisticated approaches integrate data profiling tools to identify heterogeneous patterns in feature distributions, enabling AutoML algorithms to flag anomalies proactively. Yet, the lack of universally accepted benchmarks for quantifying data quality challenges the standardization of such tools, leaving room for further innovation.

While automated preprocessing techniques have significantly matured, a major point of contention remains their scalability and ability to generalize across heterogeneous datasets. Emerging trends focus on context-aware preprocessing techniques facilitated by advances in large language models (LLMs) [13]. Here, LLMs dynamically interpret dataset semantics to recommend preprocessing transformations, but the integration of these models into AutoML systems is still in early stages, with concerns about computational resource requirements.

In conclusion, automated data preprocessing in AutoML demonstrates a paradigm shift towards leveraging machine learning models as not only the end-product but also a medium for solving data preparation challenges. However, achieving context sensitivity, computational efficiency, and adaptability remains critical for real-world adoption. Addressing these challenges will require innovative cross-disciplinary approaches, combining advancements in both AutoML and domain-aware systems. The field is primed for further research on frameworks that adapt dynamically to dataset variability while incorporating interpretability and transparency in preprocessing decisions.

2.2 Feature Engineering and Representation Learning

Feature engineering and representation learning are pivotal stages in Automated Machine Learning (AutoML), directly impacting the performance, interpretability, and generalizability of resulting models. The automation of these processes aims to identify, extract, and synthesize informative features from raw data while reducing human intervention, ensuring efficiency across datasets with diverse properties. This subsection explores the state-of-the-art approaches to automated feature engineering and the integration of representation learning techniques to abstract complex feature sets, effectively bridging preprocessing dynamics and downstream model performance discussed earlier.

Automated feature engineering streamlines the traditionally labor-intensive process of selecting, transforming, and generating feature sets, allowing AutoML frameworks to dynamically adapt to underlying dataset characteristics. Methods such as recursive feature elimination (RFE) and model-based filtering evaluate the importance of features systematically within a given dataset. Tools like TPOT [16] and OpenFE [19] employ evolutionary algorithms and feature-boosting techniques to optimize feature combinations, ensuring a balance between simplicity and predictive power. For example, OpenFE's two-stage pruning algorithm stands out for its ability to efficiently assess feature utility, achieving state-of-the-art results in benchmarks and real-world contexts. However, despite achieving high performance, these methods often falter in semantic interpretability, which is a key consideration in domains like healthcare and finance, where understanding feature relevance is essential.

In addition to feature selection, automated feature synthesis enhances the scope of feature generation by creating new derived features through operations such as arithmetic combinations, polynomial expansions, and logarithmic transformations. Genetic programming approaches have demonstrated significant promise, particularly in capturing non-linear interactions [20]. Reinforcement learning has further advanced automated synthesis, where it navigates expansive transformation spaces adaptively to identify optimal feature sets [21]. Nevertheless, computational efficiency remains an ongoing challenge, as high-dimensional datasets amplify overheads. Addressing this tradeoff is critical for ensuring that synthesis methods remain scalable and practicable for large-scale applications.

Representation learning, heavily influenced by advancements in deep learning, complements automated feature engineering by enabling the extraction of high-quality latent data representations. Techniques like autoencoders compress data into compact, abstract forms while retaining critical structural information [12]. These methods are particularly advantageous for unstructured data types, including text, images, and time-series data, where traditional engineered features may underperform. Multimodal AutoML frameworks, for instance, unify disparate data types using embeddings for integrated analysis [22]. However, while these latent embeddings frequently surpass handcrafted features in predictive power, the abstraction comes at the cost of interpretability. Striking a balance between representation depth and explainability remains a pressing issue, particularly in sensitive applications involving decision transparency.

An emerging trend in feature engineering automation is the integration of domain knowledge, a critical step towards enhancing interpretability and relevance. Tools like CAAFE [13] leverage Large Language Models (LLMs) to generate context-aware features, combining the advantages of domain expertise with machine intelligence. By introducing semantically meaningful transformations, these hybrid approaches effectively address the gap between algorithmic rigor and practical applicability. Such methods not only improve feature richness but also ensure that the generated features align more closely with specific problem contexts, minimizing the need for manual input.

Scalability poses another significant challenge for automated feature engineering frameworks, especially as they are deployed on increasingly large datasets. Multi-fidelity feature evaluation techniques and distributed computation frameworks have started to alleviate resource constraints, enabling effective processing at scale [18]. Industrially viable solutions like SAFE balance interpretability and computational efficiency, a crucial factor as AutoML systems are applied in production environments [12].

Future advancements in automated feature engineering and representation learning should aim to harmonize interpretability, domain relevance, and computational feasibility. Progress in hybrid models combining domain expertise with machine intelligence, as well as efforts to bridge symbolic and neural representations, hold promise for the next generation of AutoML systems. Moreover, addressing the tradeoffs between abstraction and explainability will be critical to fostering wider adoption, particularly in regulated industries. Frameworks like OpenFE and CAAFE exemplify the potential of innovation in this domain, pointing towards an AutoML ecosystem that integrates efficiency, scalability, and domain alignment seamlessly with the broader goals of model selection and generation.

2.3 Model Selection and Generation Automation

Model selection and model generation lie at the heart of Automated Machine Learning (AutoML) systems, dictating their ability to efficiently tailor machine learning pipelines to the unique characteristics of specific datasets. This subsection explores the methodological advancements and challenges associated with automating these two critical tasks. While traditional static model selection strategies have been the cornerstone of AutoML, recent efforts in dynamic model generation aim to create more adaptable, resource-efficient solutions.

The traditional approach to model selection within AutoML frameworks generally involves searching across a predefined set of algorithms. Techniques such as Combined Algorithm Selection and Hyperparameter Optimization (CASH) automate both the choice of model and the tuning of its hyperparameters by defining a joint search space. Bayesian optimization has been extensively employed to explore this space efficiently, leveraging probabilistic models to balance exploration and exploitation [1], [23]. This meta-learning-based paradigm often incorporates dataset meta-features—summaries of the dataset's statistical or structural properties—to match datasets with algorithm configurations that have performed well in similar prior settings [3]. Despite its widespread adoption, the CASH framework is computationally intensive, particularly when evaluated against high-dimensional search spaces or large-scale datasets.

Dynamic model selection, a more adaptive alternative, leverages meta-learning and dataset embeddings to select optimal algorithms in real time. Meta-knowledge extraction enables AutoML systems to refine selections by continuously learning from prior model performance [3], [24]. Additionally, multi-fidelity optimization techniques, which evaluate partial configurations or subsets of data at an early stage, have proven effective in reducing computational costs while maintaining selection accuracy [25].

Beyond selection, dynamic model generation introduces another layer of adaptability by automating the creation of novel machine learning architectures, particularly in deep learning contexts. Neural Architecture Search (NAS) has emerged as a dominant approach for designing neural network architectures. Early NAS methods employed reinforcement learning or evolutionary algorithms to explore architecture search spaces [5], [26]. However, these methods were computationally prohibitive, often requiring thousands of GPU hours to converge on optimal architectures. Recent advancements such as Differentiable Neural Architecture Search (DARTS) have dramatically improved efficiency by enabling gradient-based optimization directly over the architecture space, circumventing the need for discrete search operations [27].

Transfer learning and pre-trained models are pivotal for reducing the computational burden and enhancing flexibility in dynamic model generation. By capitalizing on pre-trained architectures, automated systems fine-tune models for downstream tasks, thereby circumventing the need for substantial computational resources required for training from scratch [28]. This approach, when combined with search strategies, allows AutoML pipelines to adapt to domain-specific challenges with minimal manual intervention [29].

Emerging methods also address critical challenges in resource-constrained environments. Resource-efficient NAS designs lightweight architectures for edge devices by integrating multi-objective optimization that balances accuracy, latency, and memory consumption [5]. Such advances in efficiency are particularly crucial as AutoML systems become integral to large-scale and real-time applications.

Despite these advancements, open questions remain. Optimal integration of selection and generation remains underexplored. Hybrid approaches that dynamically transition between model selection and generation, depending on the problem complexity or computational budget, represent a promising area for development. Moreover, incorporating fairness, interpretability, and resource constraints as core objectives in the automation process presents both a technical and ethical challenge for the field [2].

In conclusion, model selection and generation automation continue to benefit from advancements in optimization algorithms, meta-learning, and scalable neural architecture design. As AutoML moves towards greater sophistication, future directions include the development of holistic frameworks capable of unifying selection and generation processes, ensuring adaptability, efficiency, and equity across diverse applications. These innovations will play a pivotal role in fulfilling AutoML's vision of democratizing machine learning while driving its adoption in real-world scenarios.

2.4 Hyperparameter Optimization Strategies

Hyperparameter optimization (HPO) serves as a foundational pillar within the AutoML workflow, playing a defining role in enhancing model performance by determining the optimal configuration of parameters that govern machine learning algorithm behavior and training dynamics. This subsection provides a cohesive exploration of traditional HPO methods, contemporary advancements, and emerging trends, while examining their respective tradeoffs, efficiencies, and underlying principles. Positioned between discussions on model selection and evaluation methodologies, HPO acts as a critical link that directly impacts both the efficiency of generated models and the rigor of their validation processes.

Traditional approaches to hyperparameter tuning, such as grid search and random search, remain staple techniques in AutoML workflows. Grid search systematically explores all possible combinations of predefined hyperparameter values, ensuring comprehensiveness but incurring steep computational costs, particularly in high-dimensional search spaces or when applied to complex models [2], [3]. Random search alleviates this burden by stochastically sampling configurations, focusing on the most influential hyperparameters in high-dimensional spaces, and often achieving similar or superior results with fewer evaluations [30]. However, while random search avoids the rigidity of exhaustive exploration, it fails to leverage information about the structure of the search space, which limits its potential efficiency gains.

Bayesian optimization has emerged as a more refined approach to these limitations by introducing probabilistic surrogate models, such as Gaussian Processes (GPs) and Tree-structured Parzen Estimators (TPEs), to guide the search process. These models iteratively identify promising hyperparameter configurations by balancing the tradeoff between exploration of uncertain regions and exploitation of known high-performing candidates through acquisition functions. This strategy supports resource-efficient optimization, making it a mainstay in many modern AutoML platforms [31], [32]. Despite these advantages, Bayesian methods encounter challenges in scaling to high-dimensional or highly categorical search spaces, where methods like dimensionality reduction or hierarchical tuning often become necessary workarounds [2], [33].

Evolutionary algorithms offer a compelling alternative, inspired by processes of natural selection. These methods iteratively evolve populations of candidate configurations using operations like mutation, crossover, and selection to converge to performant solutions, proving particularly effective in rugged or multimodal search landscapes [5], [26]. Evolutionary approaches are also well-suited for distributed environments, enhancing their appeal for large-scale tasks. Notably, strategies like aging evolution have demonstrated their robustness by jointly optimizing both neural architectures and associated hyperparameters, achieving strong results on diverse benchmarks [34]. Nevertheless, their reliance on numerous evaluations often limits their efficiency, especially when evaluating resource-intensive fitness objectives.

Contemporary advancements in hierarchical and adaptive optimization address the need for efficiency and scalability by dynamically refining search strategies. Multi-fidelity optimization, for instance, leverages approximations—such as reduced dataset sizes or fewer training iterations—to estimate model performance early, thus enabling cost-efficient decision-making without compromising overall quality [35], [36]. Hierarchical methods decompose complex search spaces into multiple levels, allowing broader hyperparameters to be set before fine-tuning more granular configurations, effectively reducing complexity in large-scale scenarios like Neural Architecture Search (NAS) [37]. Such mechanisms closely align with the workflow efficiencies achieved in dynamic model selection and generation, reinforcing the benefits of adaptability across the AutoML pipeline.

Emerging approaches in meta-learning and transfer learning further accelerate HPO by leveraging prior knowledge from previously optimized tasks to narrow search domains and initialize "warm-start" optimizations. These strategies utilize meta-learned surrogate models that predict promising hyperparameter regions based on dataset characteristics, bypassing the need for exhaustive search during initial stages [28], [38]. While this transfer of knowledge offers significant computational savings, ensuring generalizability across varied tasks remains a key challenge, reflecting similar complexities in dynamic model selection and generative methodologies.

As concerns about resource consumption and environmental impact grow, researchers are increasingly advocating for energy-efficient and cost-aware HPO methods. These approaches explicitly factor energy usage into optimization metrics or integrate lightweight, low-resource frameworks, driving progress towards "Green AutoML" [32], [39]. Simultaneously, advances in automating constraint-aware optimization—addressing fairness, interpretability, or domain-specific requirements—are essential for extending AutoML's utility in complex and high-stakes applications.

In conclusion, hyperparameter optimization not only serves as a critical enabler for cutting-edge AutoML workflows but also provides a vital touchpoint for advancing the balance between effectiveness, efficiency, and responsibility in machine learning automation. Continued innovation in surrogate modeling, adaptive tuning, and sustainability-oriented techniques will be integral to bridging upstream model generation processes with downstream evaluation and validation, ensuring AutoML systems evolve coherently and ethically to address diverse real-world demands.

2.5 Evaluation and Validation in AutoML Workflows

Evaluation and validation are critical components in ensuring the reliability and performance of models produced by Automated Machine Learning (AutoML) workflows. These components encompass the automated selection and measurement of evaluation metrics and the rigorous design of validation strategies, both of which are pivotal for unbiased and effective AutoML system outcomes. In this subsection, we explore the state-of-the-art techniques, discuss their strengths and limitations, and identify the emerging trends and challenges in automating robust evaluation and validation within AutoML frameworks.

One of the foundational techniques for model evaluation in AutoML is cross-validation, particularly nested cross-validation, which is frequently employed to ensure unbiased estimates of generalization performance and hyperparameter tuning. Traditional nested cross-validation methods, while robust, often impose significant computational burdens. Lin et al. [40] demonstrate that for many practical scenarios, simpler methods, such as flat cross-validation, can produce comparable results with drastically reduced computational cost, provided the hyperparameter space is appropriately constrained. This highlights a trade-off between computational feasibility and statistical rigor, which remains an active area of exploration in AutoML.

Multi-objective evaluation strategies are gaining attention as AutoML systems strive to optimize not just predictive accuracy but also aspects like model interpretability, fairness, and resource efficiency. Pareto-based optimization methods [41] provide a systematic approach to balancing these often-conflicting objectives, enabling the discovery of Pareto-optimal solutions that address diverse stakeholder requirements. Despite their promise, these techniques face scalability challenges when applied to high-dimensional hyperparameter and model spaces, necessitating innovations such as surrogate-based multi-objective Bayesian optimization [42].

Synthetic data, which can augment or replace real-world validation datasets in certain scenarios, holds potential to reduce the cost and time required for AutoML validation. Frameworks such as Fabolas [42] optimize configurations on smaller data subsets to extrapolate performance on larger, more complex datasets, significantly accelerating the evaluation process. Similarly, active learning paradigms have been incorporated to dynamically sample critical evaluation points, thereby improving validation efficiency without sacrificing accuracy or robustness [43].

Automated error analysis and model debugging have emerged as complementary practices within validation workflows. By leveraging interpretability tools like symbolic regression [44], AutoML pipelines can automate the identification of error regions within the input space, providing actionable insights for improving model robustness. However, these automated diagnostic techniques often inherit the limitations of the underlying interpretable models, requiring additional research to handle more sophisticated model architectures such as neural networks.

A key emerging trend within evaluation is the integration of energy-efficiency metrics, as the environmental impact of AutoML continues to grow. Recent efforts in constrained Bayesian optimization explicitly consider energy consumption as part of the evaluation framework, balancing it with predictive accuracy [45]. These innovations align computational practices with broader sustainability goals, advancing the field towards "Green AutoML."

Challenges persist, particularly in automating the evaluation of fairness and bias in AutoML workflows. Despite advances in fairness-aware learning [41], reconciling fairness objectives with performance metrics remains a complex, multi-faceted problem. Furthermore, as AutoML pipelines handle increasingly heterogeneous datasets, ensuring equitable evaluation across data subgroups introduces additional complications.

Future directions in evaluation and validation for AutoML include enhancing the scalability of nested validation techniques, integrating domain-specific metrics into general-purpose AutoML frameworks, and developing more adaptive validation strategies for non-stationary environments. The ongoing fusion of surrogate-based modeling and dynamic evaluation methods promises to reduce computational cost and improve reliability without compromising rigor. As AutoML systems become more autonomous, the development of standardized, fully automated evaluation benchmarks for fairness, robustness, and efficiency will be paramount for advancing research and ensuring widespread adoption.

3 Methods and Algorithms Driving Automated Machine Learning

3.1 Search Space Design and Optimization Techniques

Search space design and optimization techniques form the cornerstone of Automated Machine Learning (AutoML) systems by ensuring efficient exploration of the possible configurations for machine learning tasks. The search space encompasses all candidate solutions, including machine learning models, feature transformations, and hyperparameter configurations, essential for constructing and selecting optimal pipelines. Designing effective search spaces and employing optimization strategies tailored for AutoML are central to achieving robust, scalable, and computationally efficient workflows.

The construction of an effective search space requires balancing flexibility and tractability. A well-defined search space must be expressive enough to encapsulate diverse solutions while compact enough to enable efficient exploration. Excessive flexibility, characterized by vast spaces, often leads to intractable optimization problems, whereas overly restrictive designs risk omitting high-performing configurations. Recent methods such as "tree-based pipeline optimization" demonstrate how hierarchical structures can organize pipeline components and foster efficient space traversal [16]. This hierarchical organization introduces modularity, enabling the optimization process to focus simultaneously on model selection, feature engineering, and hyperparameter tuning. Similar paradigms are also explored through evolutionary composition frameworks which achieve efficient search across complex spaces by leveraging genetic algorithms for compositional design [46].

Once a search space is defined, optimization methods guide the exploration to identify the most promising configurations. Bayesian Optimization (BO) has emerged as a popular method, especially in hyperparameter tuning, due to its ability to model the objective function with probabilistic surrogate models, such as Gaussian processes, enabling sample-efficient exploration [1]. BO excels in single-objective optimization but struggles with scalability in high-dimensional or combinatorial search spaces common in AutoML. Extensions like Tree-structured Parzen Estimators (TPE), designed for conditional spaces, provide computational advantages by handling discrete and continuous variables simultaneously [41].

Reinforcement learning (RL) offers an alternative approach that treats the optimization process as a sequential decision-making problem, with the reward signal derived from model performance. RL-based Neural Architecture Search (NAS) has demonstrated the efficacy of policy-gradient methods for dynamically navigating search spaces [11]. However, RL is computationally intensive, often requiring extensive trials and large-scale resources, making it unsuitable for real-time applications or resource-constrained environments.

Evolutionary algorithms present another robust category of optimization techniques, especially suited for complex, multimodal search spaces. Genetic algorithms, for instance, simulate evolutionary mechanisms such as selection, crossover, and mutation to iteratively refine solutions. Frameworks like LEAF exemplify the efficiency of leveraging evolutionary strategies for optimizing both neural network architectures and their hyperparameters simultaneously, enabling trade-offs between performance and complexity [5]. Compared to Bayesian methods, evolutionary algorithms excel in diverse search landscapes but may sacrifice sample efficiency due to their stochastic nature.

Combining knowledge transfer with search space reduction is an emerging trend aimed at addressing scalability challenges. Meta-learning techniques utilize performance data from previously solved tasks to initialize search processes in novel tasks, significantly reducing the exploration overhead. For example, meta-learning has been employed to refine search spaces dynamically, facilitating efficient optimization in tasks sharing structural similarities [47]. Similarly, leveraging pre-trained architectures in NAS pipelines capitalizes on foundational structures to accelerate the search process [4].

Despite these advances, significant challenges remain. Multi-objective optimization is becoming increasingly relevant, as users demand models that balance predictive accuracy with secondary objectives such as interpretability, fairness, and resource efficiency. Additionally, real-time and adaptive optimization for dynamic data environments, where data distributions evolve over time, is an area of active investigation [7]. Novel strategies that integrate low-cost surrogate objectives, such as partial model evaluation or synthetic data, show promise in mitigating computational burdens [48].

In conclusion, search space design and optimization techniques have matured significantly, offering both general-purpose strategies and tailored approaches for specific AutoML tasks. The future of this domain lies in synthesizing advancements across meta-learning, multi-fidelity optimization, and resource-efficient methods to create unified frameworks capable of handling diverse, large-scale, and dynamic environments. Continued refinement of search techniques and synergistic integration of domain knowledge will propel AutoML systems toward achieving unprecedented levels of autonomy and efficiency.

3.2 Neural Architecture Search (NAS) Advances

Neural Architecture Search (NAS) is a cornerstone of Automated Machine Learning (AutoML), aiming to automate the design of neural network architectures tailored to specific tasks or datasets. As a fundamental area within AutoML, NAS addresses the challenge of navigating vast and complex search spaces to identify optimal or near-optimal architectures efficiently. This subsection delves into recent advancements in NAS methodologies, comparing traditional techniques with emerging innovations while outlining trade-offs, challenges, and future directions.

At the core of NAS is a threefold process: first, defining a search space that encapsulates possible neural network configurations; second, employing a search strategy to traverse this space effectively; and third, utilizing performance evaluation methods to assess candidate architectures. Differentiable Neural Architecture Search (DARTS) has emerged as a transformative approach in this domain, leveraging continuous relaxation techniques to replace the inherently discrete search space with a differentiable one. This reformulation enables efficient optimization through gradient descent, significantly accelerating convergence compared to traditional reinforcement learning-based NAS methods, which often relied on policy-gradient techniques and involved computationally prohibitive trial-and-error exploration [49]. Despite its efficiency, DARTS has notable limitations, including susceptibility to instability due to overfitting on validation sets and a tendency to converge to suboptimal architectures if the search space is not adequately constrained.

Evolutionary algorithms (EAs) continue to hold a prominent role in NAS, particularly in population-based search strategies. These algorithms iteratively evolve a population of candidate architectures via genetic operations like mutation, crossover, and selection, striking a balance between exploration and exploitation. EAs have proven particularly effective in multi-objective scenarios, such as optimizing not only accuracy but also secondary objectives like resource efficiency, latency, and energy consumption [16]. While EAs are generally less computationally efficient than differentiable NAS methods, they excel in discovering diverse architectures and exhibit superior robustness when tackling heterogeneous or complex datasets [50].

The development of innovative and efficient search spaces has further propelled advancements in NAS. Cell-based and hierarchical search spaces have emerged as key strategies, wherein neural network "cells" or macro-modules serve as reusable building blocks, simplifying the search complexity and reducing dimensionality. These structured search spaces improve architectural generalization across tasks while enhancing computational efficiency [50]. Such innovations have also been incorporated into hardware-aware frameworks, addressing resource constraints by designing architectures optimized for specific requirements like low latency or energy consumption, which are critical in applications like mobile and edge computing [51].

Another significant trend is the integration of neural architecture search with transfer learning. By reusing pre-trained models or transferring learned architecture designs, this hybrid approach reduces the computational costs associated with NAS for new datasets or tasks. Leveraging meta-learning principles further constrains the search space, creating a balanced interplay between efficiency and adaptability to task-specific domains [16].

However, despite these advancements, challenges persist in scaling NAS to high-dimensional search spaces without compromising accuracy. The computational expense of NAS, particularly for datasets requiring large-scale or highly complex architectures, remains a barrier to wider adoption in real-world contexts. To mitigate these limitations, researchers have introduced multi-fidelity optimization techniques, which accelerate convergence by performing intermediate evaluations on reduced dataset resolutions or subsets, achieving significant efficiency improvements while maintaining competitive accuracy [16].

Looking ahead, the future of NAS lies in its integration with transformative areas like multimodal learning and foundation models. By leveraging large pre-trained models as baselines in NAS, the computational overhead of architecture searches can be further reduced, enhancing cross-domain adaptability. Ethical imperatives, such as fairness-aware and energy-efficient architecture design, will also play an increasingly critical role in guiding NAS methodologies toward more responsible and inclusive frameworks [52].

In conclusion, Neural Architecture Search has evolved into a sophisticated field, moving beyond brute-force approaches to embrace resource-aware and task-adaptive methods. Crucial future directions include addressing the trade-offs between computational cost, architectural diversity, and real-world applicability. Advancing these dimensions will contribute to the development of sustainable and pragmatic NAS frameworks, further cementing its role as an indispensable tool in Automated Machine Learning.

3.3 Meta-learning and Transfer Learning Approaches

Meta-learning and transfer learning are pivotal techniques in enhancing the efficiency, adaptability, and generalization capabilities of Automated Machine Learning (AutoML) systems. Both paradigms leverage knowledge from prior tasks or datasets, enabling the scaling of computational efficiency and performance for new problems. This subsection explores the theoretical and practical underpinnings of these approaches, comparing methodologies, and identifying their advantages, limitations, and emerging trends.

Meta-learning, often referred to as "learning to learn," focuses on building models that can adapt quickly to new tasks by learning patterns across a variety of prior tasks. Central to meta-learning is the meta-knowledge framework, wherein model parameters or task-level properties are optimized to generalize across datasets. Recent advancements in this area include model-agnostic meta-learning

(MAML), which optimizes parameters to quickly adapt to unseen tasks through gradient updates on few-shot learning datasets. Transfer learning leverages pretrained models or representations, often derived from large datasets (e.g., ImageNet for vision or GPT for text), making these approaches particularly suitable for data-scarce contexts. Combining meta-learning with transfer learning has been shown to further reduce the computational overhead of AutoML pipelines by narrowing the search space to configurations known to be effective across related tasks [5], [28].

The application of meta-learning to AutoML systems has fostered innovations in dynamic model selection, hyperparameter optimization, and search space reduction. For example, meta-knowledge encoded as meta-features can guide model selection by mapping dataset characteristics (e.g., number of features, sparsity) to prior performance metrics [3], [23]. An exemplar of this approach is Auto-sklearn, which initializes optimization by leveraging prior tasks through meta-learning, effectively expediting the search process. Moreover, portfolio construction for warm-starting AutoML processes often employs meta-learning to predict promising configurations, as demonstrated by frameworks like Auto-PyTorch [36].

Transfer learning, on the other hand, focuses on reusing pretrained model weights or architectures to address similar downstream tasks. A notable strategy involves initializing neural search spaces with architectures pretrained from similar domains to dramatically reduce convergence times in Neural Architecture Search (NAS) [5]. Another extension of transfer learning integrates task similarity measures to improve cross-task parameter initialization. Examples include systems such as FLAML, which deploy low-cost configurations learned from prior tasks to streamline both model selection and hyperparameter tuning [32]. Meta-learning frameworks often exploit those pretrained representations to identify latent dataset similarities and task transferability, as shown in key studies that quantify task affinities to guide transfer strategies [23], [27].

Despite their complementary strengths, meta-learning and transfer learning face overlapping challenges. Challenges in meta-learning often include the difficulty of selecting meta-features that generalize across diverse domains, coupled with substantial end-to-end training costs on high-dimensional archives of historical tasks. Conversely, transfer learning's efficacy largely relies on the relevance of pretrained models to target datasets, with domain gaps causing degraded performance. For instance, retraining large-scale foundation models such as BERT or ResNet for niche applications often results in substantial computational overhead and mismatched representations, a problem partially addressed by domain-specific fine-tuning [29].

Emerging trends aim to address these challenges by combining meta-learning and transfer learning in more flexible, resource-efficient ways. One trajectory focuses on automated relational meta-learning, which leverages graph-based or relational representations to characterize task similarities and inform transfer decisions [2]. Additionally, lightweight versions of AutoML frameworks are being developed to utilize low-resource environments efficiently. Techniques such as multi-fidelity optimization have proven effective in identifying early-stopping criteria for tasks unlikely to benefit from further computational effort [36].

Future directions lie in developing unified frameworks for continual learning, where meta-learning mechanisms dynamically evolve based on newly observed tasks. Such systems could identify task shifts or concept drift, enabling adaptable pipelines for real-world data streams [39]. Furthermore, the integration of meta-learning and transfer learning with graph-based task representations offers promising prospects for modeling complex inter-task relationships, providing deeper insights into generalization boundaries.

In conclusion, meta-learning and transfer learning have firmly established themselves as foundational pillars of AutoML, simultaneously expanding its reach and reducing its computational demands. However, realizing their full potential necessitates advancements in task similarity quantification, resource-efficient optimization, and integration within heterogeneous domains. Addressing these challenges will propel their efficacy, ensuring that AutoML continues to democratize access to powerful machine learning techniques.

3.4 Resource-aware Optimization Methods

Resource-aware optimization methods focus on designing machine learning workflows and models that achieve high predictive performance while adhering to constraints such as computational cost, memory usage, and energy efficiency. These constraints are increasingly critical in real-world use cases, particularly in environments like mobile devices, edge computing platforms, and large-scale distributed systems. This subsection explores foundational strategies, trade-offs, and recent advancements in resource-aware optimization methods for Automated Machine Learning (AutoML), while linking their relevance to broader goals such as multi-objective optimization and sustainable machine learning practices.

A crucial trade-off addressed by resource-aware optimization is balancing computational cost with model performance. Traditional AutoML approaches have often prioritized accuracy maximization without accounting for resource consumption, rendering high-performing models impractical for constrained environments. To counter this limitation, techniques such as cost-aware hyperparameter optimization explicitly include resource metrics, like training time or memory consumption, as part of the optimization objective. Tools like AutoHAS [35] integrate these resource constraints, ensuring that the resulting architectures and hyperparameters achieve an optimal balance between accuracy and efficiency. Similarly, resource-constrained Neural Architecture Search (NAS) methods [37] emphasize latency, memory, or energy efficiency, enabling deployment-ready solutions for resource-constrained devices.

Multi-objective frameworks have emerged as a powerful mechanism for tackling the inherently conflicting trade-offs between resource usage and performance, creating synergies with the principles outlined in multi-objective optimization (discussed in the subsequent subsection). Pareto-based optimization approaches systematically identify configurations that yield optimal trade-offs between accuracy and resource efficiency [27]. Constrained Bayesian optimization further extends this paradigm, employing penalty functions or acquisition functions to enforce resource limitations while optimizing hyperparameters or model architectures [33]. These methods facilitate informed exploration within feasible solution spaces, striking a balance between practical deployability and strong predictive performance.

Beyond hyperparameter optimization and architectural search, lightweight model architectures and efficient training pipelines have become central to resource-aware AutoML. Techniques such as pruning, quantization, and knowledge distillation offer practical solutions for reducing model size and computational overhead without significantly compromising accuracy. For instance, frameworks like FLAML [32] incorporate these techniques into AutoML workflows, enabling deployment on memory-constrained or energy-sensitive devices. Pruning removes redundant parameters, quantization leverages low-precision numeric formats to compress models, and distillation trains smaller, efficient "student" models using the knowledge from larger "teacher" models—collectively ensuring resource-efficient yet performant solutions.

In addition to model-level optimizations, distributed and asynchronous training strategies are vital for achieving scalability in resource-aware systems. Solutions like data-parallel neural architecture optimization [34] enable efficient processing of large datasets across distributed hardware. These strategies dynamically tune configuration parameters—such as batch sizes, learning rates, and the number of parallel processes—to maximize throughput while minimizing latency. Frameworks like LEAF [5] leverage asynchronous training combined with reinforcement learning to enhance efficiency under resource constraints, aligning with broader goals of scalability and adaptability in AutoML.

Energy-efficient solutions are becoming increasingly significant in light of global sustainability priorities. Tools like Carbontracker [2] provide mechanisms to monitor and minimize the energy usage of AutoML workflows, supporting environmentally conscious machine learning practices. Techniques such as multi-fidelity optimization and early-stopping criteria have proven especially useful for minimizing energy demand during intermediate search phases [53]. Emerging "Green AutoML" practices prioritize efficient resource utilization across tasks, reflecting a growing emphasis on sustainability in AutoML development.

Future directions in resource-aware AutoML are increasingly focused on adaptability and real-time dynamic adjustment. For example, continual learning frameworks that adjust model capacity based on real-time resource availability hold promise for enhancing efficiency in non-stationary data environments. Similarly, resource-efficient meta-learning methods [28], which transfer knowledge across tasks to reduce redundant computations, offer significant potential for improving scalability in both single-task and multi-task AutoML workflows.

However, notable challenges remain. Many resource-aware optimization frameworks still rely on overly complex or computationally expensive mechanisms, undermining the promise of full automation [54]. The design of adaptive systems that can seamlessly balance resources and performance across diverse contexts is an ambitious but essential research challenge. Furthermore, explainability remains a critical aspect of resource-aware optimization, particularly in domains like healthcare or autonomous systems where transparency and trust are non-negotiable.

In summary, resource-aware optimization methods are indispensable for bridging the gap between advanced machine learning capabilities and the practical constraints of limited-resource environments. By integrating multi-objective optimization principles with sustainable practices and adaptive strategies, resource-aware methodologies are laying the groundwork for democratizing AutoML—making it feasible and accessible across an ever-expanding range of devices, infrastructures, and applications. This reflects a broader vision for AutoML systems that balance performance with societal and environmental responsibility, seamlessly transitioning into the advanced multi-objective pipelines discussed in the following subsection.

3.5 Advances in Multi-objective Optimization

Multi-objective optimization (MOO) plays a critical role in automated machine learning (AutoML) by enabling the simultaneous optimization of multiple, often competing objectives, such as predictive accuracy, computational efficiency, interpretability, and fairness. This subsection provides a comprehensive exploration of the advances in MOO within AutoML, focusing on recent approaches, their strengths, limitations, and emerging trends. MOO is essential for balancing trade-offs in real-world applications where single-objective optimization is insufficient to capture the multifaceted demands of modern machine learning systems.

Traditional MOO techniques in AutoML often leverage Pareto-based methods to find solutions that lie on the Pareto front, which represents configurations where no improvement in one objective is possible without deteriorating another. Pareto-based approaches, such as NSGA-II and related evolutionary algorithms, have demonstrated success in producing diverse sets of trade-off solutions. For example, evolutionary AutoML frameworks such as LEAF [5] leverage genetic algorithms to optimize hyperparameters and model architectures across objectives such as model accuracy and size, enabling the deployment of lightweight models with minimal performance degradation. While evolutionary methods are effective for exploring large search spaces, they often incur substantial computational costs due to the iterative nature of evolutionary processes.

Gradient-based MOO has garnered significant attention recently, enabling more scalable optimization in high-dimensional spaces by modifying gradients to align with multiple objectives. Multi-gradient descent algorithms target the optimization of all objectives simultaneously by solving vector projection problems at each iteration, yielding solutions that converge faster than traditional evolutionary algorithms. This approach is especially useful for differentiable objectives such as accuracy and fairness metrics. However, gradient-based methods face challenges when objectives conflict sharply or are non-differentiable, such as interpretability constraints, which often necessitate alternative approaches.

A particularly innovative area within MOO for AutoML is the adoption of hybrid multi-fidelity methods, which aim to reduce computational overhead by employing surrogate models, low-fidelity evaluations, and early-stopping mechanisms. Techniques such as Freeze-Thaw Bayesian Optimization [55] and multi-fidelity Bayesian optimization frameworks [42], [56] dynamically adjust resource allocation by leveraging partial evaluations of configuration solutions. These approaches enable efficient exploration of the search space, trading off high-quality evaluations with computational efficiency, an increasingly critical requirement for large-scale AutoML systems.

Fairness-aware optimization represents an emerging trend in MOO, driven by the necessity of addressing bias in machine learning systems. Methods such as fairness constraints embedded into multi-objective optimization pipelines [41] ensure that AutoML frameworks can balance performance objectives with equitable treatment across demographic groups, mitigating disparities induced by biased data or algorithms. Although effective, such approaches often involve trade-offs between objectives like fairness and accuracy. Empirical results illustrate that achieving fairness can lead to marginally reduced performance, but the societal benefits significantly outweigh these drawbacks.

Despite these advances, challenges remain in MOO. Scalability to high-dimensional spaces and the curse of dimensionality continue to hinder the efficiency of existing methods, especially in dynamic and constrained AutoML systems. Resource-aware optimization techniques continue to be refined to mitigate this bottleneck [45]. Additionally, the lack of standardized evaluation metrics for trade-off solutions limits fair comparison across MOO approaches, underscoring the need for universally accepted benchmarks.

Future research is expected to focus on designing integrated frameworks that seamlessly blend MOO with meta-learning, reinforcement learning, and continual learning paradigms. Advances in explainable and interpretable AutoML pipelines are also critical for improving stakeholder trust, particularly in high-stakes domains such as healthcare and finance. Moreover, efforts to integrate ethical principles into MOO processes are poised to shape future developments in responsible AutoML, paving the way for sustainable and socially aware machine learning optimization. Advances in "Green AutoML" practices [32], [45] will also play a pivotal role as sustainability concerns become increasingly critical.

In conclusion, advances in MOO have profoundly enhanced AutoML systems by introducing sophisticated methods for balancing competing objectives. These approaches unlock new possibilities for deploying smarter, fairer, and more resource-efficient machine learning solutions, yet substantial opportunities remain to push the boundaries of MOO in AutoML further.

3.6 Advances in Optimization Frameworks for Automated Pipelines

Unified and efficient optimization of end-to-end machine learning pipelines is a cornerstone objective of modern AutoML systems, enabling seamless integration and coordination across pipeline stages such as data preprocessing, feature engineering, model selection, hyperparameter tuning, and evaluation. Given the inherent complexity and interdependencies of these stages, recent advancements have focused on developing novel optimization frameworks that bring together algorithmic innovation, system-level efficiency, and adaptability to diverse use cases.

One of the most significant contributions in this area is the Combined Algorithm Selection and Hyperparameter Tuning (CASH) paradigm, which has become foundational to many state-of-the-art AutoML systems. CASH methodologies simultaneously optimize algorithm selection and hyperparameter configurations, reducing the overall search space dimensionality while boosting system efficiency. Frameworks like Auto-sklearn [23] and FLAML [32] highlight these advancements. Auto-sklearn integrates meta-learning to facilitate a warm-start optimization process that leverages prior knowledge from similar datasets, enabling faster convergence on ideal solutions. In contrast, FLAML prioritizes lightweight resource-efficient optimization, employing adaptive search mechanisms to minimize computational overhead. These frameworks illustrate the trade-offs between computational cost and optimization efficacy—while meta-learning-based approaches excel in domains with ample historical data, they may falter on outlier tasks where prior knowledge is limited [23]. Conversely, lightweight frameworks favor universality and runtime performance, often excelling under constrained computational budgets [32].

Building upon CASH principles, a complementary direction has emerged in the form of end-to-end dynamic pipeline optimization frameworks. These frameworks aim to co-optimize all pipeline stages adaptively, addressing the holistic nature of machine learning workflows. Solutions like VolcanoML [57] advocate for modular decomposition of the search space into smaller, interdependent subspaces. This approach enables systematic and hierarchical search strategies, reminiscent of relational database optimization, to improve scalability and execution efficiency. Similarly, Auto-PyTorch [36] incorporates multi-fidelity optimization frameworks and portfolio-based initialization, iteratively refining pipeline quality across fidelity levels. By incrementally optimizing individual pipeline components within budgetary constraints, Auto-PyTorch outperforms traditional one-shot approaches, particularly in resource-limited scenarios [36]. However, these methods can face challenges in non-stationary contexts where dynamic adaptation to shifting data distributions is essential.

Incremental and real-time optimization frameworks are addressing such demands by introducing solutions that adapt pipelines in response to evolving workloads or data characteristics. Frameworks like Online AutoML [7] extend traditional AutoML pipelines into real-time settings, leveraging techniques such as asynchronous successive halving to accelerate optimization in streaming environments. These approaches excel in scenarios involving concept drift or continuously arriving data, although their reliance on initial pipeline configurations can be a bottleneck under prolonged, high-magnitude data shifts [58].

At a higher level, system-wide optimization frameworks further enhance pipeline efficiency by modeling workflows as directed acyclic graphs (DAGs). Techniques like StreamLINE [59] capitalize on interdependencies between pipeline components, dynamically pruning suboptimal branches to improve scalability and interpretability. Additionally, emerging energy-aware AutoML frameworks [60] prioritize sustainability by incorporating carbon footprint metrics into optimization objectives. These frameworks promote trade-off analysis between accuracy, runtime efficiency, and resource consumption, addressing the growing need for environmentally conscious machine learning.

Despite this progress, challenges remain in scaling optimization frameworks to accommodate heterogeneous data modalities, such as blending textual, visual, and tabular data, which exacerbates the curse of dimensionality. Adaptive frameworks that promise resilience under changing distributions still encounter obstacles in intermittently labeled or weakly supervised environments. Recent progress in multimodal learning, such as AutoMMLab [61], offers a promising avenue for extending pipeline generalization, but robust solutions for dynamic contexts remain a research priority.

Future innovations in this domain will likely center on hybrid methodologies that integrate evolutionary algorithms, meta-learning, and reinforcement learning, addressing the dual demands of scalability and interpretability. Real-time optimization frameworks could benefit from embedding transfer learning mechanisms for initialization, further refining their adaptability to unseen data [28]. Meanwhile, establishing standardized benchmarks that evaluate real-world deployment scenarios will be critical to advancing the field [62].

By combining system-level innovations with algorithmic progress, these frameworks signify a pivotal step toward democratized AI. With applications spanning critical domains such as healthcare, finance, and automated industrial workflows, the evolution of AutoML optimization promises increasingly adaptive, sustainable, and accessible systems, bridging the automation gap in data-driven decision-making.

4 Applications of Automated Machine Learning in Diverse Domains

4.1 Transforming Healthcare through Automated Machine Learning

Automated Machine Learning (AutoML) has emerged as a transformative force in healthcare, reshaping how diagnostic processes, treatment protocols, prognostics, and operational workflows are optimized. By mitigating the reliance on domain-specific manual interventions, AutoML accelerates machine learning deployment in diverse healthcare applications, offering greater accessibility to high-precision predictive models and data-driven decision-making. The marriage of AutoML and healthcare, however, is not without its complexities, encompassing challenges such as ensuring clinical interpretability, mitigating bias, and managing sensitive data.

In medical diagnostics, AutoML has proven indispensable for the analysis of complex medical imaging data. AutoML systems refine workflows by automating image processing, classification, and segmentation tasks, achieving state-of-the-art results in applications like cancer detection, pneumonia diagnosis, and diabetic retinopathy screening. The use of tree-based optimization tools such as TPOT [16] enables efficient automated pipeline design tailored to specific radiographic or histopathological datasets. Neural Architecture Search (NAS) techniques, particularly those integrating differentiable search mechanisms [4], have further enhanced diagnostics by discovering optimal deep learning architectures specific to medical imaging modalities like CT scans and MRIs. Nevertheless, these systems face nuanced trade-offs: while NAS-enabled systems provide precision, they often demand significant computational resources, potentially limiting their immediate adoption in resource-constrained clinical setups.

Beyond diagnostics, AutoML significantly advances personalized medicine, where individualized treatment plans are crafted based on genomics, proteomics, and patient health records. By automating feature extraction and selection, frameworks such as autofeat [12] facilitate the identification of clinically relevant biomarkers. In oncology, for instance, AutoML-driven predictive models combine genomic mutation data with phenotypic information to recommend targeted therapies, transforming treatment paradigms. However, the interpretability barrier remains a critical limitation here. Decision-makers often demand algorithmic transparency, especially when predictions directly influence patient outcomes, as highlighted in discussions on transparency in AutoML workflows [63]. Therefore, interpretability-centered advancements in explainable AI (XAI) must be integrated more seamlessly into AutoML to ensure trust across stakeholders.

Another domain where AutoML demonstrates considerable promise is drug discovery and development. Traditional drug discovery pipelines, often spanning years, can be accelerated with AutoML's ability to predict compound bioactivity, optimize chemical synthesis pathways, and simulate drug-target interactions. Evolutionary frameworks like LEAF [5] facilitate not only hyperparameter tuning but also structural optimization of predictive deep learning models, reducing computational expenses while maintaining high efficacy. These methods, though innovative, must overcome two pressing challenges: the interpretive complexity of biochemical interactions within machine-learned models and the need to scale processes for large chemical libraries without incurring prohibitive resource costs.

In operational settings, AutoML bolsters healthcare systems by streamlining resource allocation and optimizing workflows. Predictive models crafted via adaptive AutoML systems, such as OAML [7], dynamically adjust to fluctuating patient influx during epidemics or seasonal surges, providing real-time insights into triaging and hospital bed management. AutoML's ability to iteratively reconfigure pipelines in response to evolving objectives, such as concept drift in outpatient care, ensures resilience and adaptability. However, deploying such systems reliably in real-time under performance constraints demands innovative solutions for latency and computational tractability.

Despite its transformative capabilities, the adoption of AutoML in healthcare warrants careful attention to emerging challenges. For instance, biases embedded in training datasets can inadvertently propagate through AutoML pipelines, as observed in studies of fairness constraints [2]. Similarly, the ethical utilization of sensitive patient data remains an ongoing concern. Future research must integrate privacy-preserving AutoML mechanisms, such as federated learning paradigms, to ensure data security without compromising performance. Additionally, "Green AutoML" principles [2] hold the potential to address the environmental costs of resource-intensive model generation algorithms.

Looking ahead, the convergence of AutoML with domain-specific advances such as multimodal data integration and hybrid decision-making systems is likely to drive further innovation. For example, coupling AutoML with large pre-trained foundation models in healthcare [64] could enable more nuanced insights by unifying imaging, textual, and genomic data. In conclusion, while strides in accelerating healthcare workflows through AutoML are evident, achieving a balance between precision, scalability, interpretability, and ethics is quintessential to its seamless integration into healthcare ecosystems.

4.2 Financial Applications of Automated Machine Learning

The financial industry, characterized by its dynamic markets, high-stakes decision-making, and expansive datasets, has emerged as a prominent beneficiary of Automated Machine Learning (AutoML). In this data-centric sector, AutoML facilitates a broad spectrum of applications, including fraud detection, credit scoring, market forecasting, and customer segmentation. This subsection examines these use cases while analyzing the methodologies driving automation and addressing the trade-offs, challenges, and opportunities that define AutoML's role in financial services.

Fraud detection is a critical priority for financial institutions, as it addresses both operational vulnerabilities and reputational risks. AutoML systems excel in automating the generation of optimized anomaly detection models, effectively handling the imbalanced datasets commonly found in transactional data. Many approaches leverage ensemble learning strategies to identify fraudulent behavior within vast data streams. For example, AutoML frameworks empowered by imbalance-aware techniques, such as those described in [52], dynamically adjust sampling rates or integrate cost-sensitive learning to detect rare instances of fraud. While traditional methods—like isolation forests and supervised tree-based models—continue to serve as cornerstones, AutoML provides the added advantage of streamlining algorithm configuration and hyperparameter tuning with minimal manual input. However, the interpretability of these models, critical for compliance in regulated environments, remains a key concern. Recent developments, including rule-based models like those proposed in [65], suggest promising avenues to marry performance with transparency. Yet, scalability issues persist, particularly when these interpretable approaches are deployed in large-scale operational systems.

In credit scoring and risk assessment, AutoML plays an increasingly critical role in predicting default probabilities and evaluating creditworthiness. Frameworks based on gradient boosting, as outlined in [66], stand out for their ability to model intricate feature interactions while maintaining robustness across diverse datasets. With heightened regulatory attention on fairness, recent innovations in AutoML integrate fairness-aware mechanisms into pipelines to ensure equitable treatment across demographics, aligning with compliance requirements. These techniques achieve an important balance between predictive performance and fairness by embedding interpretable scoring systems into workflows—a necessity in the face of stringent regulatory scrutiny. However, a clear trade-off emerges: while complex black-box models, such as neural networks, often outperform traditional systems in specific scenarios, they face resistance due to their opacity, underlining the critical need for solutions that do not sacrifice interpretability for performance.

Market trend analysis and financial forecasting pose unique challenges for AutoML due to the highly volatile, non-stationary, and temporally dependent nature of financial time-series data. AutoML frameworks equipped with forecasting-specific algorithms, such as those discussed in [67], optimize the full modeling process, including preprocessing steps like seasonal decomposition and feature engineering, alongside model selection. Hybrid approaches, such as the combination of tree-based models with recurrent neural networks, are being explored to enhance predictive accuracy. Despite these advances, real-world deployment of such systems must contend with the key challenges of managing concept drift and ensuring temporal consistency in dynamic financial environments.

Customer segmentation and personalization applications further demonstrate the potential of AutoML in the financial sector. Modern frameworks use AutoML to identify strategic patterns that inform targeted marketing and personalized product offerings, from customized credit card programs to tailored investment strategies. Advances in feature engineering frameworks, like those outlined in [12], facilitate the construction of latent, high-impact features, enabling precise segmentation while maintaining interpretability. Furthermore, cutting-edge AutoML solutions integrate multimodal datasets, connecting transactional, demographic, and behavioral data for deeper insights into customer preferences. However, the computational demands of preprocessing and feature construction for multimodal data can present challenges, as evidenced in studies like [22], requiring innovative approaches to efficiency and scalability.

Despite considerable strides in enabling automation, several challenges remain in applying AutoML to financial services. Regulatory compliance, particularly regarding transparency and fairness, frequently conflicts with the inherent opacity of AutoML-generated model ensembles. Likewise, balancing the need for high performance with operational resource constraints requires a nuanced optimization strategy. Addressing these challenges will demand research into joint optimization methodologies that prioritize interpretability, efficiency, and fairness, as suggested in [39]. Simultaneously, incorporating domain expertise into AutoML processes, as proposed in frameworks like [18], promises to bolster model customization and trust, enhancing relevance and user confidence in the financial domain.

In summary, AutoML has emerged as a transformative force in financial services, democratizing access to sophisticated machine learning tools while enhancing operational efficiency and scalability. By addressing remaining barriers—such as regulatory transparency, computational scalability, and resource efficiency—the financial sector is well-positioned to fully leverage the power of AutoML. Innovations integrating fairness optimization, interpretable modeling, and human-centered design will play a pivotal role in shaping the evolution of AutoML applications, setting new benchmarks for data-driven decision-making in this critical industry.

4.3 Scientific Discovery and Research Enabled by AutoML

Automated Machine Learning (AutoML) has been transformative in accelerating scientific discovery and research by enabling the automation of critical processes across fields like materials science, chemistry, and physics. Through its ability to optimize workflows, AutoML facilitates not only more precise hypotheses but also significantly reduces the iteration time between experimentation and validation. By merging data-driven insights with domain knowledge, AutoML is expanding the horizons of scientific exploration.

In materials science, AutoML is reshaping how scientists approach the discovery of new materials with desired properties. Traditional materials design often involves trial-and-error experimentation, guided by expert intuition. AutoML systems streamline this process by automating property predictions and synthesis planning with advanced machine learning models. Frameworks such as LEAF [5], which incorporate evolutionary algorithms to design and optimize neural network architectures, have demonstrated significant success in predicting material properties while meeting real-world constraints like minimizing material costs. Additionally, the use of unsupervised tensor mining tools like AutoTen [68] enables researchers to extract informative patterns from vast, multi-aspect data, optimizing component identification and clustering. By automating such processes, AutoML accelerates material virtualization—facilitating the search for innovative materials in applications ranging from semiconductors to energy storage systems.

In chemistry, AutoML is revolutionizing workflows for reaction characterization and optimization. For example, in computational chemistry, surrogate modeling using AutoML techniques significantly enhances the efficiency and precision of simulations. The AutoML-GA framework [43] demonstrates how integrating AutoML with genetic algorithms can optimize reaction conditions and improve simulation accuracy while reducing the number of computational iterations. Furthermore, AutoML techniques extend beyond simple automation; they integrate interpretability into workflows, helping chemists understand the logical dependencies between reaction parameters. Symbolic regression-based AutoML, for instance, provides explicit, interpretable mathematical formulations for chemical reactions, underscoring the growing potential of explainable AI in advancing chemical research [50].

In physics, AutoML has facilitated advancements in high-energy physics and quantum computing by supporting the modeling and validation of complex theoretical frameworks. AutoML systems are increasingly utilized for hypothesis testing in areas characterized by high-dimensional and noisy datasets. For example, advanced AutoML frameworks employing neural architecture search (NAS) [27] have enabled the design of algorithms to process experimental data from particle colliders. These methods achieve state-of-the-art performance by dynamically adapting search spaces based on task-specific requirements. Furthermore, symbolic regression techniques enriched by AutoML have been instrumental in deriving governing equations from physical data, enabling researchers to uncover simplified representations of intricate phenomena [69].

Despite these advancements, challenges remain in terms of scalability and generalization. For instance, while evolutionary approaches like LEAF can tailor solutions to specific tasks, they often require substantial computational resources. Similarly, while frameworks like AutoTen excel in large-scale pattern discovery, the dependency on domain-specific heuristics limits their applicability in less-explored domains. To address these constraints, future research must prioritize resource-efficient AutoML paradigms, particularly those that leverage multi-fidelity optimization [36]. Another emerging trend is the integration of foundation models with AutoML to enable multimodal data processing, which shows promise for uncovering cross-disciplinary insights [29].

In summary, AutoML is driving a paradigm shift in scientific research, transforming data-driven methodologies into adaptable, automated systems. By reducing human intervention in labor-intensive computational steps, streamlining experimental workflows, and enabling accessible interpretability, AutoML is setting the stage for more efficient and democratized scientific inquiry. However, achieving true scalability and universal applicability will require the continued evolution of AutoML frameworks, focused on computational efficiency, cross-domain generalization, and ethical integration into scientific discovery pipelines. Combined with advances in algorithmic accountability and interpretability, AutoML has the potential to redefine the process of generating scientific knowledge.

4.4 Revolutionizing Industrial Automation with AutoML

Industrial automation, a cornerstone of manufacturing and process-driven industries, has experienced a transformative overhaul with the advent of technologies like IoT (Internet of Things), AI, and notably, Automated Machine Learning (AutoML). Positioned at the nexus of automation and data-driven innovation, AutoML has emerged as a critical enabler of advanced workflows including predictive maintenance, quality assurance, and process optimization. By reducing the complexity of model deployment and fostering scalability, AutoML is democratizing the application of machine learning across the ever-evolving landscape of industrial environments.

Predictive maintenance stands out as one of the most impactful areas where AutoML drives innovation in industrial settings. Modern manufacturing facilities leverage a vast array of sensors and IoT devices to continuously monitor machinery health and operational conditions. AutoML enables the automated creation of predictive models that can identify anomalies and foresee equipment failures in real time. Unlike traditional methods relying on static thresholds or predefined rules, AutoML pipelines dynamically adapt to changing machinery and process environments. This adaptability is bolstered by techniques like automated feature selection, as seen in tools such as OpenFE [19], which extract the most relevant features from heterogeneous industrial datasets. Simultaneously, multi-fidelity optimization methods [32] ensure robust predictive performance while minimizing computational requirements. Additionally, evolutionary strategies for hyperparameter tuning and neural architecture optimization allow AutoML solutions to cater to the unique operational constraints of industrial systems [5].

Quality assurance, another vital domain in industrial automation, benefits immensely from the streamlined workflows enabled by AutoML. High-precision manufacturing often hinges on accurate defect detection, a challenge compounded by the variability of defect patterns and the need for frequent model updates. Traditional machine learning methods require substantial manual effort for feature engineering and data curation, but AutoML automates these steps to accelerate the development of reliable models [15]. For example, frameworks employing neural architecture search (NAS), such as Auto-Keras [37], autonomously generate domain-specific architectures optimized for defect detection tasks, surpassing heuristic-based approaches in performance and efficiency. Furthermore, AutoML-driven transfer learning [28] facilitates the adaptation of pre-trained models to quality control processes at new production sites or for novel product lines, alleviating the dependency on extensive labeled datasets and minimizing retraining efforts.

Process optimization has also advanced significantly through AutoML, as it empowers industries to autonomously identify optimal configurations within complex production systems. Leveraging real-time IoT data streams, AutoML frameworks like Tune [31] automate the fine-tuning of production parameters, thereby improving throughput, reducing waste, and enhancing energy efficiency. These systems employ sophisticated optimization techniques, including Bayesian methods [33] and hybrid evolutionary algorithms, to explore expansive search spaces while respecting real-world constraints such as energy consumption and material availability [50]. Additionally, AutoML enables the creation of robust, simulation-based "what-if" analyses, equipping industries with insights into trade-offs between competing objectives such as cost, quality, and sustainability [50].

While the contributions of AutoML to industrial automation are groundbreaking, key challenges persist. The effectiveness of AutoML depends heavily on access to clean, high-quality data; however, industrial IoT sensors frequently generate noisy, incomplete, or inconsistent data. This underscores the importance of robust preprocessing pipelines, as exemplified by automated anomaly detection techniques like AutoOD [70]. Furthermore, the computational intensity of many AutoML processes raises concerns about resource efficiency and sustainability, driving interest in energy-conscious "green AutoML" approaches [2], [32].

The future of AutoML in industrial automation is poised to be shaped by innovations in cross-domain generalization and real-time adaptability. Developments in continual learning and adaptive frameworks will enable AutoML systems to tackle dynamic challenges such as sensor data drift [2], extending their utility to non-static environments. Additionally, the integration of multimodal data streams—combining audio, visual, and tactile information—promises to significantly enhance manufacturing intelligence and system-wide optimization [71]. Ethical considerations and fairness in AutoML workflows will also grow in importance, particularly in safety-critical applications involving collaborative human-machine environments.

By simplifying machine learning deployment, enhancing scalability, and delivering actionable insights, AutoML is spearheading a new wave of efficiency and resilience in industrial automation. Its integration with emerging industrial technologies signals a paradigm shift towards fully autonomous manufacturing ecosystems, driving performance gains while addressing the demands of an increasingly data-centric world.

4.5 Advancing Natural Language Processing and Computer Vision

The integration of Automated Machine Learning (AutoML) into Natural Language Processing (NLP) and Computer Vision (CV) has substantially advanced the field, automating core processes that were traditionally human-intensive while driving innovation in text analytics, image recognition, and multimodal learning. This subsection examines the pivotal contributions of AutoML to these domains, analyzing existing techniques, identifying trade-offs, and articulating emerging trends.

In NLP, AutoML frameworks have shown immense potential by optimizing tasks such as text classification, sentiment analysis, named entity recognition, and machine translation. Traditional human-guided approaches to these tasks involve substantial trial and error for selecting models, architectures, and hyperparameters. With tools like Neural Architecture Search (NAS) and meta-learning strategies, AutoML automates these processes, achieving results comparable to—or exceeding—state-of-the-art manually optimized pipelines [35], [72]. For example, NAS techniques have identified novel architectures that significantly enhance transformer-based models in machine translation tasks while maintaining computational efficiency. AutoML systems like FLAML [32] focus on lightweight alternatives, making NLP automation feasible even in resource-constrained environments, such as mobile device inference of language models.

For image recognition, AutoML has automated the design of deep convolutional neural networks (CNNs) and their hyperparameters, yielding state-of-the-art improvements in image classification, object detection, and semantic segmentation tasks. Evolutionary and reinforcement learning-based NAS methods [5], [26] have demonstrated the ability to discover high-performing architectures for vision problems. These methods outperform manually designed architectures by balancing accuracy and resource efficiency, as shown in applications such as medical imaging and autonomous driving. Further, AutoML frameworks now incorporate hardware-aware strategies, optimizing models not only for predictive performance but also with constraints like inference latency and memory usage [35]. Such adjustments are crucial in deploying vision solutions on devices with limited computational resources.

One of the most transformative capacities of AutoML in modern AI lies in multimodal learning, where synergistic processing of heterogeneous data, such as text, images, and audio, is required. Techniques like automated feature fusion and representation alignment have enabled systematic learning across modalities [34]. For example, AutoML-driven multimodal models excel in tasks such as video captioning and speech-to-image retrieval, which demand seamless integration of disparate data sources. At the intersection of NLP and CV, significant progress has been made in generating multimodal embeddings that leverage foundation models like transformers, adapted for specific applications using AutoML's dynamic search mechanisms. These approaches, exemplified by frameworks such as AutoHAS Efficient Hyperparameter and Architecture Search [35], facilitate end-to-end optimization of multimodal learning pipelines, ensuring state-of-the-art performance without the need for extensive manual intervention.

Despite these advances, challenges remain in scaling AutoML to large-scale NLP and CV datasets. While methods like progressive Bayesian optimization and freeze-thaw approaches [55], [56] attempt to mitigate the high computational demands of NAS and hyperparameter optimization, trade-offs between optimization time and final model performance persist. Moreover, the interpretability of AutoML-optimized models—particularly in high-stakes applications such as medical diagnosis and autonomous systems—continues to be a pressing concern [73]. Researchers have begun incorporating explainable AI tools, ensuring that AutoML-derived decisions are transparent and actionable.

Looking forward, emerging trends such as integrating AutoML with foundation models and advancing energy-efficient optimization techniques [35], [45] are expected to dictate the future trajectory of automation in NLP and CV. By adopting multimodal optimization under resource constraints and integrating ethical considerations, AutoML can further democratize AI, broadening its accessibility while addressing societal challenges. The harmonization of data, algorithmic creativity, and computational efficiency positions AutoML not as a mere automation paradigm, but as a transformative force in the evolution of machine learning systems at scale.

4.6 Applications in Climate, Environment, and Agriculture

Automated Machine Learning (AutoML) is driving transformative advancements in addressing critical challenges across climate science, environmental conservation, and agriculture. By automating end-to-end model creation processes—including data preprocessing, algorithm selection, and hyperparameter optimization—AutoML is unlocking unprecedented capabilities for tackling data-intensive and complex problems in these domains. This subsection explores its applications, analyzing contributions to climate modeling, ecological monitoring, and agriculture, while identifying limitations and imagining future opportunities.

In climate science, AutoML has proven to be an invaluable tool for optimizing the highly intricate models used to simulate atmospheric systems and forecast long-term climate trends. Traditional climate modeling workflows typically demand extensive expert intervention, including parameter selection, ensemble configuration, and sensitivity analyses for variables such as greenhouse gas concentrations. AutoML frameworks streamline these efforts. For example, neural architecture search (NAS) has been employed in climate simulation tasks, enabling the adaptive and automated modeling of complex atmospheric interactions [74]. Moreover, AutoML's ability to integrate real-time data from satellites, weather sensors, and historical databases has enhanced the accuracy of short-term weather forecasts by dynamically adapting predictive models. Despite these benefits, AutoML's application in climate science is constrained by the computational demands of large-scale simulations. Energy-efficient approaches, like FLAML [32], demonstrate the necessity of prioritizing resource-conscious tools to make AutoML a viable solution for climate applications on a global scale.

In ecological monitoring, AutoML has reshaped the analysis of large-scale geospatial and remote sensing data from satellites, drones, and other sensors. These advancements are empowering researchers to track biodiversity, detect patterns of deforestation, and measure habitat degradation more efficiently. Tools such as AutoGluon have shown scalability in processing multimodal datasets by combining imagery, spatial metadata, and temporal data, thereby improving classification accuracy for ecological tasks like detecting forest cover loss and mapping wildlife corridors [75]. A prominent challenge in ecological data is class imbalance, as events such as deforestation or species extinction are inherently rare. AutoML frameworks incorporating methods for addressing class imbalance, as discussed in [52], have proven essential. Techniques like synthetic oversampling and feature transformation enhance model reliability without sacrificing fine-grained predictive accuracy, making AutoML particularly well-suited for tackling ecological issues.

In agriculture, AutoML is revolutionizing precision farming by optimizing models for tasks such as crop yield forecasting, disease detection, pest management, and resource allocation. By automating traditional feature engineering and leveraging transfer learning techniques, AutoML simplifies the processing of diverse agricultural datasets—from ground sensors to satellite imagery—to predict critical outcomes such as soil health or irrigation requirements [28]. Additionally, AutoML has facilitated the development of edge-ready models designed for resource-constrained environments, enabling real-time monitoring and decision-making. For instance, AutoML-generated neural networks, optimized for efficiency, now power IoT-enabled systems that monitor field conditions and deploy targeted interventions, such as variable-rate seeding or precise pesticide application. These breakthroughs are particularly impactful in data-scarce agricultural settings, where frameworks like AutoPrognosis excel by balancing prediction accuracy and computational efficiency [76].

Despite these advancements, significant challenges remain. The computational intensity of AutoML, particularly in neural architecture search, continues to be a barrier to adoption, especially in resource-constrained environments such as low-income regions disproportionately affected by climate change. Efforts to enhance the interpretability of AutoML systems, such as those demonstrated by tools like STREAMLINE [59], hold promise for increasing trust among key stakeholders, including policymakers, conservationists, and farmers. However, more work is needed to make AutoML processes transparent and accessible. Additionally, the integration of environmentally conscious principles, such as Green AutoML [60], is essential to ensure that AutoML aligns with sustainability goals.

Looking ahead, AutoML has the potential to integrate multimodal data sources across climate, ecological, and agricultural domains, creating deeply adaptive systems for evolving use cases. By addressing the trade-offs among computational cost, model interpretability, and scalability, AutoML could become a foundational technology for global efforts in sustainability and resilience. Such advancements promise to unlock innovative solutions to the pressing challenges of climate change, environmental conservation, and agricultural productivity.

5 Challenges and Ethical Considerations in Automated Machine Learning

5.1 Scalability and Resource Constraints

Scalability and resource constraints present prominent challenges in the development and deployment of Automated Machine Learning (AutoML) systems. As the scale, complexity, and accessibility of AutoML grow, addressing computational limitations, memory use, and resource optimization has become paramount. These challenges are particularly exacerbated in environments with restricted resources, such as edge devices or distributed systems, where operational latency, hardware limitations, and expense all converge.

The high computational cost of AutoML processes stems primarily from the expansive search spaces involved in tasks such as hyperparameter optimization (HPO) and neural architecture search (NAS). For example, NAS often requires evaluating numerous candidate architectures across large datasets, resulting in significant energy consumption and runtime overhead. Methods like Bayesian optimization have been employed to mitigate such costs by guiding the search space exploration intelligently, but even these approaches encounter difficulties as the dimensionality of the tasks increases [4], [41]. Other techniques, including bandit-based algorithms like Hyperband, aim to reduce unnecessary resource consumption by allocating computational effort adaptively based on early performance indicators. While effective in many cases, these methods still demand considerable computational power, limiting their viability in low-resource settings [41].

Latency and runtime bottlenecks pose further scalability challenges, especially in real-world industrial applications requiring near-real-time decisions. For instance, training a large-scale model involving complex preprocessing and feature engineering pipelines can delay the deployment cycle when operating over high-dimensional datasets [3]. One solution is the use of parallel processing frameworks and distributed computing, such as MapReduce or Spark, to enhance scalability. However, distributed execution introduces new trade-offs, including increased communication costs and potential system failures during parallel execution [6]. The potential of asynchronous genetic programming and successive halving for optimizing pipelines in dynamic environments has also shown promise, particularly for addressing data drift in online scenarios [7].

Moreover, deploying AutoML in resource-constrained environments, such as mobile devices or IoT systems, necessitates architectural compactness and energy efficiency. Lightweight NAS techniques, such as those employing differentiable search mechanisms or pruning, aim to reduce the computational footprint while maintaining sufficient model accuracy [5]. Model quantization and knowledge distillation are additional techniques aimed at ensuring that AutoML frameworks can function effectively within the strict memory and energy limitations of edge devices. LEAF, for instance, optimizes both architecture size and performance through evolutionary strategies, thereby facilitating scalable yet efficient solutions [5].

Data storage and management remain another critical bottleneck for AutoML systems, particularly as intermediate results and search records occupy substantial memory. Incremental storage strategies, such as sparse data management and on-the-fly computation of features, reduce memory overhead while maintaining computational accuracy [12]. Additionally, sparse representations and memory-optimized algorithms can curtail resource-intensive storage requirements, particularly in iterative tasks like HPO and model ensembling [33].

Emerging trends in resource-aware AutoML solutions integrate green computing practices to ensure sustainability while expanding scalability. For instance, multi-fidelity optimization techniques enable efficient exploration of high-dimensional spaces by leveraging partial evaluations at reduced resource costs, thereby minimizing overall energy consumption [41]. In addition, modular, reusable models—often trained in the context of transfer learning—hold the potential to bypass resource-heavy model development cycles, particularly for edge and small-sample scenarios [47].

Future research underscores the need to mitigate the growing environmental impact of large-scale AutoML systems while maintaining accessibility and usability. Incorporating context-aware optimization and further integrating hardware-aware NAS are promising directions for developing cost-efficient, scalable pipelines. Augmenting AutoML systems with adaptive, hierarchical resource allocation strategies provides another avenue to ensure their viability across variable deployment contexts. Ultimately, balancing performance with resource efficiency will be critical to bringing AutoML to scale.

5.2 Interpretability and Trust in AutoML

The interpretability and trust of Automated Machine Learning (AutoML) systems are paramount, especially in high-stakes applications where decisions could significantly influence society or finances. Stakeholders such as clinicians, regulators, and policymakers demand transparent explanations of model predictions to ensure accountability, fairness, and informed decision-making. This subsection explores the challenges of achieving interpretability in AutoML outputs, examines methodologies within the paradigm of explainable artificial intelligence (XAI), and discusses trade-offs, emerging trends, and future directions.

AutoML systems often function as black-box frameworks, leveraging advanced techniques like ensemble learning, neural architecture search (NAS), and meta-learning to optimize pipelines. While these approaches yield high accuracy, they often obfuscate the rationale behind important decisions—such as feature selection, hyperparameter tuning, and final predictions—thereby limiting stakeholder trust and hindering adoption, particularly in compliance-sensitive sectors such as healthcare and financial services. Frameworks like TPOT [16] and Auto-sklearn [17] exemplify the black-box nature of current AutoML tools, delivering remarkable performance but often lacking transparency regarding internal processes. These transparency gaps present a formidable barrier, especially in domains where explicability is a legal or ethical mandate under regulations such as GDPR [65].

To address these challenges, interpretability methods in AutoML have bifurcated into two major strategies: promoting inherently interpretable model designs and applying post-hoc explanation techniques. Inherently interpretable approaches restrict AutoML pipelines to simpler, more transparent model forms, such as rule-based systems or linear regression. For example, optimizing sparse rule lists has been shown to improve both interpretability and predictive power for certain categorical datasets [65]. AutoML frameworks incorporating expert-driven feature engineering, such as OpenFE [19], further enhance interpretability by generating semantically meaningful and comprehensible features for downstream models.

In contrast, post-hoc explanation methods seek to elucidate decisions made by otherwise opaque models. Techniques like Local Interpretable Model-agnostic Explanations (LIME) and Shapley values have been integrated into AutoML workflows to generate feature-level or instance-level interpretability. Shapley values, for instance, help quantify the importance of features in AutoML predictions, offering insights into black-box behaviors. However, these methods face limitations in high-dimensional data contexts due to their computational complexity [77]. Emerging approaches like surrogate modeling, which simplifies complex models into interpretable symbolic regression forms, provide an alternative by approximating decision boundaries while maintaining some level of understandability [78]. However, such simplifications often involve trade-offs that may compromise predictive accuracy.

The issue of trust in AutoML frameworks is deeply intertwined with interpretable tools' efficacy and user experience. Even when AutoML delivers superior performance, research indicates that users may hesitate to trust these systems if the decision-making process remains opaque [18]. This necessitates the development of explanation tools tailored to domain-specific needs, enabling stakeholders to derive actionable insights. Context-aware methods like Context-Aware Automated Feature Engineering (CAAFE) embed semantic knowledge into feature engineering processes, producing interpretable outputs that align with application-specific requirements [13].

Despite these advancements, challenges persist. The diversity of applications and data modalities complicates the standardization of interpretability techniques. For example, methods optimized for tabular data in finance often fail to generalize to unstructured data common in computer vision [59]. Furthermore, the lack of universally accepted metrics to evaluate interpretability adds complexity, particularly in compliance-driven industries. While metrics such as interpretability-accuracy trade-offs and user satisfaction scores offer potential frameworks, they are rarely adopted across the board [79].

Looking forward, embedding interpretability as a central objective in AutoML pipelines appears vital for fostering trust and usability. Tools like PyGlove, which decouple search algorithms from models through symbolic programming, may offer modular solutions for complexity and transparency issues [49]. Multi-objective optimization frameworks that simultaneously consider fairness, interpretability, and performance could further support trust in automated systems by aligning their outputs with ethical and operational requirements [17].

To ensure widespread adoption and social responsibility, the AutoML community must prioritize transparent benchmarks that integrate interpretability metrics and strengthen the scalability of XAI tools to handle diverse and demanding datasets. User-centric designs and iterative refinement processes that facilitate stakeholder input will be critical in enhancing trust. By addressing these priorities, AutoML systems can drive responsible, impactful deployment and significantly contribute to the evolving landscape of ethical and interpretable machine learning technologies tailored to a broad array of societal needs.

5.3 Bias and Fairness in Automated Machine Learning

Bias and fairness present critical challenges within Automated Machine Learning (AutoML) pipelines, fueling growing academic and ethical concerns. AutoML's promise lies in its ability to democratize access to machine learning by minimizing human intervention. However, the implicit risks associated with introducing or exacerbating biases demand careful examination. Such biases can originate from various components of an AutoML system, including the training data, feature selection processes, algorithm choice, and objective optimization. Given that algorithmic decisions made by AutoML frameworks can directly influence social and policy outcomes in high-stakes domains like healthcare, finance, and criminal justice, ensuring fairness is a responsibility that cannot be overstated.

One fundamental source of bias stems from the training data. Datasets used in AutoML frequently encode historical and societal inequities, which are subsequently propagated or amplified during automated preprocessing and learning tasks. For instance, imbalanced demographic representation in datasets skews predictive outcomes, favoring majority groups. Standard AutoML data preprocessing techniques, such as automated missing data imputation and categorical encoding, can exacerbate these biases when fairness constraints are not explicitly integrated. While existing AutoML tools often incorporate multi-objective optimization capabilities, fairness metrics like demographic parity, equalized odds, or disparate impact ratio are rarely prioritized alongside traditional performance metrics such as accuracy or F1 scores. Recent advancements such as fairness-aware data preprocessing methods have shown promise. Tools like Auto-WEKA [1] demonstrate efficiency in selecting optimal algorithm-performance configurations but lack inherent fairness optimization. Extending such frameworks to include bias mitigation mechanisms during preprocessing remains a critical research direction.

Feature selection and engineering further amplify the fairness challenge. Automated approaches for feature selection, while facilitating efficiency, often seize on discriminative features that inadvertently encode protected attributes. For example, domains such as loan applications may involve ZIP codes as features, which serve as proxies for race or socioeconomic status, leading to discriminatory outcomes. Techniques like reinforcement learning-based feature selection [69] have been proposed to automate and improve feature set quality; however, they seldom evaluate the selected features' fairness implications. Knowledge-driven systems like OpenFE [19] have shown that leveraging expert heuristics could isolate unfair representations, but embedding ethical principles systematically into such AutoML frameworks remains an open challenge.

Algorithm recommendation mechanisms and optimization strategies also share responsibility for fairness-related pitfalls. During the model generation and evaluation phase, AutoML frameworks often focus on metrics that prioritize performance or computational efficiency, disregarding nuanced fairness considerations. For instance, state-of-the-art AutoML frameworks such as AutoGluon [75], while robust in structured data applications, do not inherently incorporate mechanisms tuned to evaluate bias-sensitive objectives. However, emerging fairness-aware optimization approaches provide promising avenues. Multi-objective optimization frameworks that integrate fairness metrics during the search process have begun to show that Pareto optimal solutions balancing accuracy and fairness are achievable. Although these efforts are still nascent, they underscore the potential for algorithm-level bias mitigation.

Despite these advances, significant trade-offs persist between optimizing fairness and traditional metrics like accuracy. While fairness constraints reduce disparities across demographic groups, they may constrain model performance relative to unconstrained optimization objectives. Moreover, incorporating fairness into an AutoML framework often increases system complexity and computational overhead, posing challenges for scalability. Balancing these competing objectives dynamically, particularly in computation-constrained environments like edge systems, introduces new research demands [32].

In understanding these challenges, future work should prioritize ethical frameworks and explicit fairness-aware design. Fairness-enhanced AutoML could leverage adaptive algorithms that incorporate fairness scores into evaluation and candidate elimination processes. Furthermore, the integration of explainable AI (XAI) can help uncover hidden biases during feature engineering and model selection, enabling actionable insights for human oversight [80]. Transparent decision-making pipelines will not only bolster user trust but also align AutoML outputs with regulatory requirements, such as the EU's General Data Protection Regulation (GDPR), which mandates explicable and non-discriminatory algorithmic decisions.

Advancing fairness in AutoML is not merely a technical challenge but also necessitates interdisciplinary cooperation. Inputs from sociologists, ethicists, and domain experts can help AutoML developers contextualize fairness metrics for diverse applications. By embedding fairness-aware principles across all stages of an AutoML pipeline—from data preprocessing to multi-objective optimization—the field can move closer to producing equitable, unbiased, and trustworthy machine learning systems that adhere to societal expectations and ethical imperatives.

5.4 Environmental and Energy Implications

The rapid growth of Automated Machine Learning (AutoML) has brought significant computational demands, prompting urgent concerns about its environmental implications. AutoML systems often rely on resource-intensive optimization processes like Neural Architecture Search (NAS), hyperparameter tuning, and full pipeline automation, which demand substantial computational resources. This results in high energy consumption and a considerable carbon footprint, underscoring the need for greater attention to sustainability. This subsection delves into the environmental impact of AutoML, explores strategies for sustainable design, and highlights trade-offs while proposing pathways toward eco-friendly solutions.

AutoML's most computationally expensive components, such as NAS and hyperparameter optimization, significantly contribute to its environmental footprint. Techniques such as evolutionary algorithms and reinforcement learning applied to NAS are especially energy-intensive due to their iterative exploration of vast model architecture search spaces [11], [26]. For instance, a typical NAS process for vision tasks has been estimated to emit as much CO_2 as multiple cross-continental flights due to prolonged GPU usage [11]. Similarly, exhaustive methods in hyperparameter optimization, such as grid search or metaheuristic strategies like Bayesian optimization, further exacerbate energy consumption, particularly as computation scales quadratically with the number of parameters explored [32], [35]. While these methods improve accuracy, their environmental sustainability is becoming increasingly untenable as AutoML expands.

Addressing these challenges requires prioritizing computational efficiency without significantly compromising performance. One promising approach involves low-fidelity and multi-fidelity optimization strategies employed in frameworks like FLAML and Auto-PyTorch Tabular. These techniques dynamically allocate resources, focusing on the most promising configurations early in the search process, thereby conserving energy by avoiding full-budget trials wherever possible [32], [36]. Similarly, transfer learning has offered effective solutions by reusing pre-trained models to warm start optimization tasks, reducing the need to explore redundant configurations [28], [81]. These innovations emphasize the value of leveraging existing computational investments to streamline AutoML processes.

Another promising area involves transparency in quantifying the environmental impact of AutoML workflows.

Tools like Carbontracker and emission profiling frameworks enable energy audits, allowing practitioners to measure the carbon footprint of their processes and identify opportunities for optimization. For example, hardware-aware NAS prioritizes architectures that are optimized for energy-efficient execution, particularly on resource-constrained devices like edge systems [35]. Moreover, incorporating energy consumption as an explicit objective in optimization processes, alongside traditional performance metrics, represents a viable pathway toward multi-objective, environmentally conscious AutoML [82].

Algorithmic innovations also play a crucial role in mitigating energy inefficiency. Lightweight AutoML frameworks, such as FLAML, exemplify how adaptive search space prioritization can align automation objectives with energy constraints while maintaining competitive performance outcomes [32]. Similarly, neural surrogate models, which approximate model performance without requiring exhaustive full evaluations, can further reduce the energy required for optimization [38]. These solutions highlight the potential for intelligently designed algorithms to integrate energy efficiency and performance considerations, driving the development of "green" AutoML systems.

Nonetheless, trade-offs between computational ambitions and ecological impact remain a significant challenge. For certain high-stakes applications, such as scientific research or autonomous systems, maximizing accuracy and performance may justify elevated energy costs. However, in many cases, fractional improvements in performance can be reasonably traded for energy-efficient configurations, promoting sustainable practices without significantly impacting overall utility [39]. To foster broader adoption of environmentally responsible AutoML, it is essential to embed ecological accountability directly into automation processes, aligning incentives for researchers, developers, and other stakeholders.

Looking ahead, scaling sustainable AutoML will require both pragmatic solutions and long-term innovation. Emerging hardware paradigms—such as quantum computing and neuromorphic architectures—may reduce computational overhead, though their integration into general-purpose AutoML remains speculative. Incorporating ethical frameworks that evaluate the social and environmental costs of model selection represents another critical frontier, ensuring that decisions about computational trade-offs are equitable and responsible. Ultimately, achieving the vision of "green AutoML" will require collective efforts to advance sustainability without limiting the accessibility, scalability, or efficacy of machine learning automation.

5.5 Algorithmic Accountability and Societal Impact

As Automated Machine Learning (AutoML) systems are increasingly embedded into diverse sectors such as healthcare, finance, and governance, concerns about algorithmic accountability and societal impact have become central to their development and deployment. This subsection examines the ethical and practical implications of delegating decision-making to these automated systems, proposes frameworks for accountability, and highlights challenges that arise at the human-machine interface. It confronts the trade-offs between automation and oversight, emphasizing the imperative of balancing innovation with responsibility.

One of the most pressing issues in AutoML is the question of accountability in model-generated decisions. When AutoML systems are deployed in high-stakes domains, such as medical diagnosis or criminal justice, pinpointing responsibility for errors or biased decisions becomes critical. Unlike traditional machine learning workflows, where domain experts control key choices such as feature selection or model parameters, AutoML abstracts many of these decisions for efficiency. This abstraction, while advantageous for scalability and accessibility, often leads to a lack of transparency in the decision-making process [83]. The challenge is amplified as AutoML workflows involve dynamic optimization, such as Combined Algorithm Selection and Hyperparameter Optimization (CASH), which adaptively searches vast parameter spaces [84]. These processes, while efficient, produce models that are effectively "black boxes," leaving stakeholders unsure of how or why decisions were made. Accountability frameworks must, therefore, prioritize interpretability and ensure that users are able to understand and interrogate the decisions produced by AutoML pipelines.

Societal-level risks of AutoML include its interactions with structural inequities and systemic biases. For instance, biases embedded in training data can be propagated and amplified through automated pipelines, even when fairness-aware techniques are applied [73]. These risks are particularly pronounced in applications such as credit scoring or hiring systems, where inequalities in historical data can carry forward discriminatory outcomes [85]. Current fairness-focused methods, such as demographic parity and disparate impact mitigation techniques, must be recalibrated to cope with the multi-objective optimization challenges of AutoML, which simultaneously handle multiple conflicting goals (e.g., accuracy, efficiency, and fairness) [41]. However, applying fairness constraints often introduces trade-offs that impact overall system performance, raising questions about how these trade-offs should be mediated and by whom.

An important dimension of algorithmic accountability is the inclusion of human oversight in AutoML architectures. While the automation of search algorithms and optimization reduces human intervention, it simultaneously introduces the risk of detachment between data practitioners and end-users. Frameworks such as Explainable AI (XAI) tools, which employ methods like Shapley values and feature attribution, are being integrated into AutoML pipelines to bridge this gap and facilitate decision transparency [86], [87]. Nevertheless, the effectiveness of XAI tools remains inconsistent in practice, requiring further development to handle the complexity of AutoML workflows.

Another emerging area of concern pertains to the societal consequences of AutoML-driven decisions. The widespread adoption of AutoML risks exacerbating digital divides between communities with differing levels of technological access and literacy. For example, communities with limited access to high-quality data or computational resources may struggle to benefit equitably from AutoML's potential [32]. To address such disparities, AutoML research must prioritize democratized access and participatory design practices to ensure that systems are inclusive and account for diverse needs and contexts. Moreover, accountability frameworks could be strengthened through regulatory mechanisms requiring audit trails and adherence to ethical guidelines, as demonstrated in compliance-driven domains like GDPR [83].

Emerging regulatory landscapes further complicate efforts to align AutoML outputs with societal values. Policies governing artificial intelligence (AI) and machine learning increasingly emphasize algorithmic fairness, transparency, and accountability. For instance, in dynamic reinforcement learning scenarios, hyperparameter configurations must evolve based on real-time performances, adding complexity to ensuring compliance with regulatory standards [88]. These challenges necessitate not only better technical solutions but also interdisciplinary collaborations between technologists, ethicists, policy-makers, and end-users.

To advance the responsible deployment of AutoML systems, future research must converge on designing accountability frameworks that embed ethical considerations into automated processes seamlessly. Techniques such as symbolic regression and meta-learning can aid in this effort by producing interpretable models and encoding human-readable abstractions of learned behaviors [72]. Moreover, the development of adaptive oversight protocols and modular AutoML pipelines could provide context-aware decision checks that blend automated efficiency with human judgment.

In conclusion, while AutoML offers vast potential for automating complex workflows, ensuring its societal alignment and accountability necessitates concerted effort. Transparent algorithm design, regulatory compliance, and participatory development processes must collectively shape the trajectory of AutoML. Only through integrating accountability into the design of AutoML pipelines can we ensure that the technologies align with societal values and create equitable outcomes for all.

5.6 Ethical Frameworks and Operationalization

Operationalizing ethical frameworks in Automated Machine Learning (AutoML) is crucial not only for safeguarding societal, environmental, and individual well-being but also for ensuring its alignment with the broader accountability discourse discussed in the preceding subsection. This subsection examines the practical embedding of ethical principles—such as fairness, transparency, sustainability, and accountability—into AutoML workflows, evaluating existing methodologies, their operational trade-offs, and the future directions necessary to foster ethical integrity in automated systems.

The formalization of ethical principles within AutoML begins with ethics-driven design, emphasizing core moral imperatives like beneficence, autonomy, and justice throughout the machine learning pipeline. Tools for fairness-aware optimization, bias detection, and advanced interpretability mechanisms are central to integrating these principles meaningfully. For instance, fairness-aware optimization systems are designed to balance predictive accuracy with equitable outcomes, using multi-objective trade-offs across fairness measures like demographic parity or equalized odds [52], [89]. However, fairness optimization can introduce significant computational complexity and conflicts with accuracy maximization, necessitating adaptive prioritization depending on the domain-specific context. As underscored in efforts such as [90], a notable challenge lies in automating the detection and correction of biases without unintentionally reinforcing existing inequities in datasets or algorithms.

A key technical advancement in operationalizing fairness within AutoML involves multi-objective Pareto optimization, which identifies optimal trade-offs between competing ethical objectives such as accuracy and fairness [91]. Yet, scalability remains a pressing challenge for such implementations, particularly in domains requiring high-dimensional data analysis, such as neural architecture search (NAS) and feature engineering. Addressing these concerns requires resource-efficient optimization strategies, including adaptive subspace exploration or energy-efficient pruning methods [32], to accommodate the computational demands of fairness evaluations.

Transparency, as emphasized in the prior subsection, remains pivotal for ethical AutoML, ensuring stakeholder trust in model decisions, especially in critical sectors like healthcare and finance. Recent explainability advancements have leveraged tools such as SHAP values, LIME, and post hoc surrogate models integrated into AutoML systems [75], [92]. These tools enable stakeholders to trace decision-making processes and verify ethical compliance. Nevertheless, balancing transparency with predictive performance presents ongoing challenges, as highly interpretable models often sacrifice generalization ability. Platforms such as [36] demonstrate the potential of domain-specific interpretability metrics for achieving better trade-offs in tabular data modeling.

Sustainability is another vital dimension of ethical AutoML, particularly given the environmental costs of large-scale computational processes. Frameworks such as Carbontracker and modular design principles proposed by [60] offer methodologies for monitoring and minimizing the environmental footprint of AutoML workflows. Practical techniques, such as reusing pre-trained models and implementing multi-fidelity optimization, as demonstrated in [57], highlight innovative ways to reduce resource-intensive search processes across pipelines. These strategies highlight the necessity of integrating sustainability considerations alongside fairness and transparency in both research and industry-standard AutoML workflows.

Finally, the dynamic nature of societal norms, shifting data distributions, and unforeseen post-deployment outcomes necessitates continuous ethical adaptability in AutoML systems. Real-time ethical evaluations can be enabled through adaptive learning techniques, leveraging meta-learning or continual-learning frameworks, such as those explored in [58]. Additionally, pre-deployment "bias stress-testing sandboxes," as advocated by [4], may serve as simulation environments for evaluating bias under various conditions, offering a proactive means of ensuring equitable outcomes.

Despite these advancements, operational barriers to ethical AutoML persist. These include the underrepresentation of marginalized groups in datasets, the lack of standardized ethical evaluation metrics, and the limited scalability of fairness-aware tools. Emerging trends suggest a move toward holistic, multi-dimensional ethical assessments—bridging considerations of fairness, sustainability, and transparency—and algorithm-centric accountability frameworks tailored to AutoML scenarios involving low-resource environments or imbalanced datasets [52]. Close collaboration between machine learning practitioners, policymakers, and domain experts remains critical to addressing these challenges.

Looking forward, AutoML systems must prioritize end-to-end ethical assurance, driven by evolving regulatory mandates and global digital responsibility standards. Future research should focus on creating lightweight ethics-integration methods that minimize computational demands while maximizing societal benefits. By fostering interdisciplinary advancements, AutoML can transcend its utility as a convenience tool and become a pillar of responsible innovation, seamlessly intersecting with the broader accountability frameworks outlined earlier.

6 Emerging Trends and Future Research Directions in Automated Machine Learning

6.1 Integration of Foundation Models and Automated Machine Learning

Foundation models, such as large-scale language models and multimodal architectures, represent significant advancements in machine learning by leveraging pre-trained representations learned from extensive datasets across diverse domains. These models offer a unique opportunity to enhance Automated Machine Learning (AutoML) workflows by addressing challenges such as search space complexity, multi-modality integration, and limited-label scenarios. This subsection examines the integration of foundation models within AutoML systems, shedding light on how this synergy reshapes automation, efficiency, and adaptability in machine learning workflows, with a particular emphasis on multimodal and task-specific adaptations.

The most significant advantage of foundation models lies in their ability to encapsulate domain-agnostic knowledge, thereby significantly reducing the workload required for model training and hyperparameter optimization across downstream tasks. Their inclusion in AutoML pipelines has enabled several transformative advancements. For instance, foundation models can serve as pre-trained backbones to expedite model selection and fine-tuning at the application level. By incorporating foundation models into pipeline search, AutoML frameworks reduce search space dimensionality and can accelerate optimization processes compared to starting from scratch. Methods that extend AutoML systems with foundation models, such as AgentHPO [64], have demonstrated superior trial efficiency by fine-tuning high-performing base models using historical data and intelligently reducing redundant tuning cycles.

One promising approach is the use of foundation models as controllers in AutoML workflows. These models act as policy agents that dynamically select preprocessing steps, feature engineering methods, or candidate models for optimization. This resembles existing work in reinforcement learning-based AutoML [11], but the incorporation of foundational knowledge significantly enhances contextual awareness and adaptability. Moreover, continuous dialogue between AutoML frameworks and pre-trained models can address concept drift, making such systems viable for dynamic and non-static learning scenarios, as shown in the online optimization paradigms explored in [7].

Another immediate application is the use of foundation models for automated feature generation and multi-modal processing. Context-aware methods such as CAAFE (Context-Aware Automated Feature Engineering) [13] exemplify the potential of language models in enriching tabular datasets by synthesizing semantically meaningful features derived from descriptive metadata. By embedding foundation models into AutoML workflows, these systems expand feature utility in a way that bridges structural gaps in datasets, which previously depended on domain-specific manual engineering. Furthermore, the ability of large multimodal foundation models, such as hybrid visual-language systems, to align heterogeneous data types has expedited the integration of multimodal AutoML systems. Existing frameworks designed for feature fusion across text, image, and time-series data stand to benefit substantially from foundation models' pretrained multimodal embeddings, providing significant gains in tasks requiring heterogeneous input alignment.

Despite substantial potential, limitations persist in the integration of foundation models with AutoML. One critical challenge is the computational cost. Large foundation models, such as GPT or multimodal transformers, are resource-intensive and introduce trade-offs in energy consumption during AutoML optimization phases [41]. Therefore, research into resource-efficient adaptation methods, such as parameter-efficient fine-tuning (e.g., LoRA or adapters), is essential for sustainable integration. Furthermore, the "black-box" nature of foundation models raises concerns regarding interpretability, complicating AutoML workflows where transparency is vital for high-stakes domains like healthcare or finance [10].

Emerging trends suggest richer hybridization of foundation models and AutoML. For instance, agent-driven AutoML paradigms, in which foundation models autonomously propose experiments and refine pipelines (e.g., MLR-Copilot) [93], provide a glimpse into the future of automated AI research. Additionally, innovations in combining foundation models with meta-learning techniques could enable hyper-personalized AutoML workflows, where task embeddings guide pipeline generation with unparalleled precision and efficiency [47].

The integration of foundation models within AutoML not only amplifies the automation of machine learning but also redefines its adaptability and application scope. As computational methods improve and foundation models become more accessible, AutoML frameworks enriched by these models can achieve transformative outcomes, particularly in complex, multimodal, or data-scarce applications. Future work lies in addressing resource constraints, enhancing transparency, and refining hybrid optimization strategies, ensuring that this synergy reaches its full potential.

6.2 Advances in Multimodal Automated Machine Learning

The proliferation of multimodal data—spanning formats such as text, images, audio, video, and sensor streams—has introduced considerable challenges to conventional Automated Machine Learning (AutoML) frameworks. These challenges arise from the inherent heterogeneity of multimodal data, necessitating effective representation learning, robust fusion strategies, and optimized processing pipelines. Addressing these complexities, emerging innovations in multimodal AutoML are paving the way for frameworks capable of dynamically integrating and optimizing heterogeneous data sources throughout the machine learning lifecycle. This subsection examines state-of-the-art methodologies, highlights exemplary frameworks, and explores emerging trends shaping multimodal AutoML.

A fundamental challenge in multimodal AutoML is aligning representation spaces across diverse modalities. Data types such as images or video typically rely on high-dimensional, dense feature embeddings, while textual or categorical data often use sparse or sequence-based representations. Frameworks like OpenFE [19] and AutoGluon-Tabular [75] emphasize modality-specific feature engineering to address this misalignment. These systems incorporate modular pipelines that optimize preprocessing and representation for each modality while facilitating their effective integration. For example, OpenFE uses feature boosting and pruned candidate selection tailored for tabular data, while recent benchmarks demonstrate the efficacy of transformers in capturing cross-modal interactions, as evidenced in stack ensembling methods that combine tree models with multimodal transformers for datasets containing both tabular and text fields [22].

Fusion and interaction modeling play pivotal roles in unlocking the full potential of multimodal data, particularly when predictive signals are distributed unevenly among modalities. Early fusion approaches, which concatenate embeddings from different modalities, often suffer from overfitting and noise sensitivity. Late fusion methods, which train separate models for each modality before combining their predictions, address these limitations but may fail to capture intricate cross-modal dependencies. Cutting-edge AutoML frameworks now incorporate hybrid fusion strategies to overcome these barriers. For instance, self-attention mechanisms and gating paradigms, as implemented in systems like CAAFE [13], enable context-aware feature transformations, preserving semantic alignment between textual and numeric attributes.

Evaluating and optimizing multimodal pipelines is another critical area for multimodal AutoML. Traditional metrics designed for single-modality data, such as independent accuracy scores, fall short in assessing performance on multimodal datasets. To address this, emerging strategies incorporate multi-objective evaluation criteria that simultaneously measure individual modalities and their interactions. Innovative methods ensure cross-modal equity by preventing dominant modalities, such as images, from overshadowing subtler signals in text or audio [52], [67]. Frameworks like TPOT [16] and SapientML [94] leverage Pareto-efficient and meta-learning principles, dynamically guiding pipeline optimization to navigate the complexities of multimodal datasets effectively.

The field is also witnessing remarkable advancements in multimodal neural architecture search (NAS), which uses AutoML techniques to design architectures optimized for heterogeneous data. Systems such as AutoOD [70] integrate curiosity-guided NAS, employing reinforcement learning to refine search trajectories for modality-specific sensitivities dynamically. Additionally, differentiable NAS, coupled with resource-aware strategies, addresses the high dimensionality and computational demands inherent in multimodal data processing, optimizing architectures for memory and energy efficiency [49].

Looking forward, several critical challenges and opportunities are shaping the future of multimodal AutoML. The development of generalized frameworks capable of seamlessly integrating emerging modalities, such as IoT sensor data or real-time streaming inputs, represents a significant research horizon [95]. Additionally, ensuring adaptability in the face of catastrophic forgetting and concept drift within evolving multimodal data streams will be essential for dynamic and lifelong learning contexts [67]. Ethical considerations, such as maintaining fairness among modalities and minimizing environmental costs during optimization, are imperative to fostering equitable and sustainable advancements in multimodal AutoML.

In conclusion, the rapid evolution of multimodal AutoML frameworks signals a transformative shift in automation, moving beyond single-modality optimization toward holistic integration of diverse data formats. These advancements establish the groundwork for tackling increasingly complex real-world problems, setting the stage for future research that integrates adaptive, sustainable, and interpretable methodologies. By refining its multimodal learning principles, AutoML has the potential to advance the limits of automation, enabling robust, flexible, and accountable solutions for diverse applications.

6.3 Adaptation of AutoML to Dynamic and Lifelong Learning Scenarios

The adaptation of Automated Machine Learning (AutoML) to dynamic and lifelong learning scenarios represents a pivotal research frontier aimed at addressing the dynamic, evolving nature of real-world environments. Traditional AutoML frameworks typically operate under the assumption of static datasets with defined tasks, focusing on once-off optimization processes for model pipeline generation. However, these static assumptions limit the deployment of AutoML in settings characterized by evolving data distributions (concept drift), new task emergence, and incremental learning requirements. Recent advancements are therefore pivoting toward the integration of continual learning paradigms, adaptive optimization strategies, and resource-efficient mechanisms to extend AutoML's applicability in dynamic and lifelong contexts.

A core avenue of research involves aligning AutoML with continual learning frameworks to enable systems capable of adapting to non-stationary data streams while minimizing catastrophic forgetting. Catastrophic forgetting, wherein previously learned knowledge deteriorates when models encounter new tasks, remains a significant challenge in lifelong learning. Innovative approaches such as parameter isolation, model distillation, and regularization-based techniques are being integrated into AutoML pipelines. For instance, neural architecture search (NAS) methods have been augmented with continual learning mechanisms to dynamically allocate resources for new tasks while preserving prior knowledge [5]. Furthermore, task-incremental adaptive pipelines are employed to reconfigure search spaces based on historical task relationships, reducing the demand for retraining from scratch [28]. Such advancements optimize model reuse and enhance computational efficiency in dynamic settings.

AutoML systems designed to address concept drift—a common challenge in real-world applications where data distributions change over time—leverage adaptive optimization strategies. Techniques such as weighted ensemble learning and adaptive hyperparameter tuning have been proposed to ensure stability under shifting data distributions [24]. Additionally, emerging AutoML pipelines integrate drift detection mechanisms at the feature and label levels, allowing for early identification of distributional changes and dynamic retraining or adaptation of pipelines [39]. Recent studies on multi-fidelity optimization [36] further demonstrate the potential to enhance efficiency by prioritizing important fidelity levels in response to detected evolution patterns, reducing computational overhead during adaptation cycles.

Resource-efficient lifelong learning frameworks are critical, especially in edge or distributed environments with constrained hardware. AutoML solutions tailored for such contexts increasingly rely on lightweight models and task compression strategies. AutoML systems, such as FLAML [32], provide a strong example of low-cost computation for continuous optimization tasks. The integration of energy-efficient NAS approaches has also shown effective performance for edge scenarios, as demonstrated by techniques that optimize operations based on hardware capacity [39].

Beyond technical advancements, the interplay between lifelong learning and meta-learning holds significant promise for enhancing AutoML's adaptability in dynamic scenarios. Meta-learning enables AutoML frameworks to abstract transferable knowledge across tasks, improving initialization strategies for dynamic optimization. Studies such as [28] underline the efficacy of meta-learning in harnessing prior task-specific experiences to accelerate convergence on novel tasks, reducing the time and computational resources required for pipeline optimization.

While the progress in adaptive AutoML is promising, challenges remain. Evolving data often introduces nontrivial constraints on scalability, necessitating the development of smarter decomposition and parallelization techniques for search spaces [96]. Additionally, achieving robust model performance under extreme task variability and rapid distributional shifts calls for more sophisticated mechanisms for uncertainty quantification and adaptive error correction.

In the future, research on embedding ethical considerations, such as fairness and interpretability, into lifelong AutoML frameworks will be essential to ensure trust in dynamic deployments. Moreover, a unified theoretical framework bridging continual learning principles with AutoML optimization objectives could facilitate broader applicability in lifelong settings. Overall, the integration of AutoML into dynamic and lifelong learning contexts stands to revolutionize automated decision-making across a plethora of real-world domains by making systems more resilient, adaptable, and autonomous.

6.4 Low-Resource, Small-Sample, and Data-Scarce Applications in AutoML

The challenges posed by low-resource, small-sample, and data-scarce scenarios represent a critical frontier in the development of Automated Machine Learning (AutoML) systems. As AutoML continues to expand its reach into diverse applications, enabling effective model generalizability and robustness despite limited data availability has become paramount. This subsection explores how advancements such as few-shot learning, transfer learning, synthetic data generation, and lightweight AutoML frameworks address these challenges, while also presenting new opportunities and research directions for improving AutoML's efficacy in data-constrained environments.

Few-shot learning has emerged as a pivotal approach for tackling data scarcity by leveraging prior knowledge to generalize from a minimal number of labeled examples. Through meta-learning techniques, AutoML systems can extract task-agnostic representations that enable rapid adaptation to novel tasks with limited labeled data. For example, strategies like Transfer Neural AutoML have demonstrated significant efficiency by reusing pre-trained models and optimizing search strategies for new tasks, thereby reducing computational and labor overheads [28]. However, the effectiveness of few-shot learning depends heavily on the quality and diversity of pre-existing tasks used in meta-training. This reliance may limit its applicability in highly domain-specific or underrepresented areas. To address these challenges, future investigations could focus on expanding task diversity or propose hybrid approaches that combine few-shot learning with techniques like data augmentation or domain-specific enhancements.

Transfer learning is another indispensable strategy for data-scarce applications, as it allows AutoML systems to repurpose knowledge from large pre-trained models for new, smaller datasets. Frameworks such as AutoFCL, which fine-tune fully connected layers for small datasets, and AutoTune, which employs Bayesian optimization to adjust convolutional neural networks during transfer learning, exemplify the effectiveness of this paradigm in enhancing task-specific performance [54], [97]. Nevertheless, transfer learning's success often hinges on the similarity between source and target domains, posing limitations when pre-trained models are unavailable or poorly aligned with the intended application. Bridging this gap necessitates the development of mechanisms capable of quantifying and mitigating domain discrepancies during the adaptation process, ensuring effective knowledge transfer even across disparate domains.

Another promising avenue for addressing data scarcity lies in automated synthetic data generation and augmentation. Techniques integrated into frameworks like AutoDS amplify sparse datasets using generative models and transformation-based augmentation pipelines, significantly enhancing downstream model performance [18]. Complementary tools like OpenFE extend this approach by automating the creation of new informative features under resource constraints [19]. Nonetheless, synthetic data techniques face the challenge of avoiding bias introduction or the generation of artifacts that may undermine real-world robustness. Addressing these concerns will require designing data augmentation methods that are more domain-aware and capable of enforcing bias control to maintain the reliability of results.

Resource-efficient AutoML systems provide another critical pathway for applications constrained by hardware, time, or financial resources. Solutions like FLAML emphasize computational efficiency and cost-effectiveness while maintaining high model performance [32]. These frameworks often exploit multi-fidelity optimization to allocate limited resources judiciously across pipeline searches and evaluation cycles. Similarly, approaches such as AutoHAS integrate architecture search with constrained hyperparameter tuning using reinforcement learning, showcasing advances in efficiency for small-sample scenarios [35]. However, balancing computational complexity with model fidelity under resource constraints remains a pressing issue. Further empirical benchmarking and refining the interplay between efficiency and performance will be vital to advancing lightweight AutoML systems for low-resource contexts.

A notable trend is the synergistic integration of multiple techniques to holistically address the challenges of data scarcity. For example, coupling fine-tuned transfer learning with domain-aware feature engineering and resource-efficient optimization strategies provides a means to mitigate individual limitations. Additionally, emerging ideas, such as leveraging large language models for AutoML task configuration, are opening new opportunities to amplify automation in constrained applications [98]. These high-level innovations may unlock novel pathways to optimize AutoML systems for minimal resource contexts.

In summary, addressing the unique challenges of low-resource, small-sample, and data-scarce scenarios in AutoML requires integrated strategies blending few-shot learning, transfer learning, synthetic augmentation, and lightweight computational frameworks. Future research must prioritize domain-specific customizations, rigorously evaluate the trade-offs between resource constraints and performance, and develop methods to counter biases introduced by augmentation techniques. By tackling these issues, AutoML can broaden its applicability, especially in underrepresented domains and edge computing scenarios, ensuring equitable access to intelligent automation regardless of data limitations.

6.5 Ethical Considerations and Responsible AutoML Development

The rise of Automated Machine Learning (AutoML) has introduced significant opportunities for democratizing access to machine learning (ML) tools. However, it also amplifies critical ethical concerns like fairness, transparency, bias mitigation, sustainability, and accountability. Addressing these challenges effectively is imperative for creating responsible AutoML systems that align with societal values and avoid reinforcing systemic issues. The following provides an analytical review of approaches to incorporating ethical principles into AutoML workflows, along with recommendations for ensuring equitable and sustainable decision-making.

Bias and fairness remain at the forefront of ethical concerns in AutoML workflows. The automated nature of these systems often propagates and amplifies biases present in training data, particularly in areas such as feature selection and objective optimization [73]. Methods for bias mitigation in AutoML include bias-aware feature engineering, which involves embedding domain-specific fairness constraints directly into algorithms. Recent advancements in fairness-aware optimization, such as multi-objective approaches balancing accuracy and equity, have shown promise in achieving consistent improvements in fairness metrics like demographic parity and equalized odds [88]. However, trade-offs between fairness and traditional optimization objectives, such as predictive performance, remain a persistent challenge. These trade-offs highlight the difficulty in designing systems capable of dynamically prioritizing fairness without adverse effects on model accuracy [41].

Transparency is another critical dimension of ethical AutoML development. The highly automated and, often, black-box nature of current systems creates interpretability gaps that erode trust among users and stakeholders. Techniques such as the use of inherently interpretable models and post-hoc explanation tools like SHAP and LIME have been incorporated into AutoML workflows to improve transparency [44]. Additionally, symbolic regression methods have been proposed to synthesize interpretable and scalable models for highly complex datasets, balancing transparency and performance objectives [78]. Despite these advancements, black-box components like neural architecture search (NAS) remain difficult to interpret, necessitating urgent innovation in explainable artificial intelligence (XAI) tailored for AutoML.

Sustainability in AutoML presents a significant ethical frontier, given the computational intensity of operations like neural architecture optimization and large-scale hyperparameter tuning. Novel methods, such as constrained Bayesian optimization (CBO), have been explored to simultaneously optimize energy efficiency and model performance thresholds, thereby reducing environmental costs [45]. Additionally, frameworks like FLAML [32] and techniques such as progressive sampling [56] prioritize resource efficiency by limiting redundant computations. However, developing consistent benchmarks and protocols for measuring environmental footprints within AutoML workflows remains an open challenge critical to fostering accountability and improvement.

The operationalization of ethical frameworks in AutoML extends beyond technical optimizations and requires embedding ethical considerations at every stage of the pipeline. This necessitates automated methods for embedding model accountability, such as bias-detection sandboxes [99], and the development of validation mechanisms for ethical consistency under dynamic conditions. Furthermore, meta-learning approaches have demonstrated potential in calibrating AutoML pipelines to meet user-defined fairness, interpretability, or energy constraints, evolving ethical objectives based on use-case specificity [100].

Despite progress, vital gaps persist, including the absence of global standards for ethical benchmarks and the limited interoperability of ethical AutoML tools across application domains. Future advancements must prioritize adaptive mechanisms that integrate fairness, transparency, and sustainability objectives directly into AutoML workflows. A promising direction lies in leveraging real-time learning for ethical decision-making, where models can adapt dynamically to shifts in ethical constraints or unforeseen societal impacts [101]. Moreover, frameworks for balancing multi-objective ethical trade-offs, underscored by empirical evidence, should be a cornerstone of future research efforts.

In conclusion, aligning AutoML systems with ethical principles requires a multifaceted approach balancing technical innovation, transparency, and societal impact. As AutoML platforms increasingly influence high-stakes applications, the need for robust, operationalized ethical frameworks becomes paramount. By ensuring open, inclusive collaboration between ML researchers, ethicists, and domain experts, AutoML can evolve into a tool not only of automation but of equity and sustainability for the broader society.

6.6 Green AutoML and Sustainability in Machine Learning Automation

Environmental sustainability in machine learning automation has become an increasingly critical concern, particularly given the rising computational demands and associated carbon emissions of large-scale workflows. Automated Machine Learning (AutoML), characterized by iterative processes such as hyperparameter optimization, neural architecture search (NAS), and ensemble construction, is particularly resource-intensive. In response, the emerging domain of "Green AutoML" emphasizes the development of energy-efficient methodologies and standardized benchmarking protocols to reduce the environmental impact while preserving competitive performance.

Central to Green AutoML is the adoption of resource-aware optimization techniques that strike a balance between computational efficiency and model accuracy. Cost-oriented hyperparameter optimization frameworks, such as FLAML, incorporate energy consumption as a factor within the optimization objective, steering algorithms toward configurations that conserve resources while maintaining predictive performance [32], [60]. For instance, FLAML achieves substantial reductions in computational requirements through lightweight heuristics for model and hyperparameter selection, cutting down resource consumption by orders of magnitude without sacrificing accuracy [32].

Meanwhile, the traditionally resource-intensive domain of Neural Architecture Search (NAS) is undergoing significant transformations to align with sustainability goals. Resource-constrained NAS strategies, including hardware-aware NAS, optimize architectures under constraints such as latency, memory footprint, and energy consumption [60]. Techniques such as evolutionary algorithms and reinforcement learning play a pivotal role by efficiently pruning resource-demanding candidates early in the search process. For example, the LEAF framework demonstrates that network minimization during architecture optimization not only reduces energy consumption but also improves deployment feasibility in constrained environments [5].

In parallel, benchmarking is emerging as a critical practice to promote transparency around energy costs and environmental impact in AutoML workflows. Tools like Carbontracker and sustainability checklists are enabling the community to standardize evaluations of carbon footprints across experiments [60]. Integration of such protocols into development cycles helps identify inefficiencies, enabling iterative refinements that embed sustainability considerations throughout the pipeline.

Another promising approach is the reuse of intermediate computations and models to minimize redundant evaluations, a critical factor in reducing energy overhead. AutoML frameworks increasingly leverage pre-trained models and shared search histories to accelerate the optimization process [28]. Transfer learning-based AutoML techniques are particularly effective, reusing architectures and representations across tasks to drastically cut both energy and time consumption during training [28].

However, achieving sustainability in AutoML often involves navigating trade-offs. Multi-objective optimization techniques, especially those based on Pareto frontiers, address the tension between computational efficiency and prediction quality. Yet, achieving an optimal balance for high-dimensional problems remains challenging, as it often requires careful calibration of trade-off weights between competing objectives.

The integration of foundation models into AutoML workflows also presents a potential pathway toward sustainability. Large-scale pre-trained models enable AutoML systems to initialize tasks with generalized representations, reducing the need for resource-intensive training from scratch [102]. Despite this promise, the high computational demands of foundation models themselves add complexity to their role in Green AutoML, especially in resource-constrained scenarios.

Looking ahead, the success of Green AutoML hinges on continued advancements in hardware-aware algorithm design, efficient search heuristics, and robust resource benchmarking protocols. Institutionalizing "green metrics" within AutoML performance evaluations provides a pathway to embed sustainability as a fundamental design principle, expanding the reach of AutoML to low-resource environments and underrepresented domains. Furthermore, developing adaptive AutoML systems capable of dynamically scaling computational requirements based on task complexity or resource availability offers an opportunity to enhance sustainability while ensuring performance and generalizability. Future research should focus on scalable solutions to these complex challenges, solidifying AutoML's role as both an accessible and environmentally responsible technology.

7 Conclusion

This subsection synthesizes the insights from the preceding discussions, presenting an overarching evaluation of the advancements, limitations, and emerging possibilities in Automated Machine Learning (AutoML). AutoML has fundamentally redefined how machine learning (ML) workflows are approached, lowering the barriers to entry for non-expert users, expediting model development, and enabling scalable deployment across diverse domains. This transformative shift is rooted in its ability to automate core ML processes—such as algorithm and hyperparameter selection [1], pipeline optimization [16], and feature engineering [12]—ushering in an era where expertise-intensive tasks are streamlined for broad accessibility.

The foundational principles of AutoML, as evidenced by the surveyed methodologies, have demonstrated substantial progress toward full pipeline automation. For example, the innovative use of combinatorial and search techniques like Bayesian optimization and genetic algorithms has been widely embraced to balance model performance and computational efficiency [5], [41]. The dynamic adaptation of techniques to context-dependent needs, such as constrained environments or edge deployments, also highlights AutoML's robustness in addressing real-world challenges [6]. Simultaneously, precision automation in niche domains like neural architecture search further exemplifies its potential in resource-constrained conditions, where hyperparameter tuning and model generation can now coexist seamlessly [4].

Despite these advancements, challenges remain pervasive. Scalability issues, particularly in large-scale or multimodal datasets, continue to pose bottlenecks for optimization frameworks, as the computational costs of exhaustive searches limit their applicability in resource-constrained scenarios [8]. Similarly, the interpretability and transparency of AutoML systems have emerged as critical concerns, especially in domains like healthcare or finance, where high-stakes decisions demand explainability and trust [10], [63]. Furthermore, bias and fairness issues within automated pipelines underscore the need for systematic inclusion of ethical safeguards to mitigate unintended consequences [2].

Recent trends suggest that AutoML is evolving toward a broader integration of interdisciplinarity and ethical frameworks. Techniques like meta-learning are enabling faster adaptation across related tasks by leveraging prior knowledge [47], while advancements in continuous learning address dynamic and lifelong adaptation requirements essential for volatile data environments [7]. Additionally, developments in Green AutoML aim to reduce the carbon footprint associated with large-scale optimization processes, signaling a shift toward sustainability in automation [2]. Moreover, the integration of large language models (LLMs) for context-aware decision-making, as demonstrated in recent frameworks like AgentHPO and CAAFE, represents a promising frontier that blends automation with semantic insights [13], [64].

Looking ahead, key priorities for advancing AutoML include refining scalability through distributed optimization strategies, bolstering interpretability mechanisms to enhance user trust, and formalizing ethical governance structures to ensure responsible automation. Rigorous benchmarking protocols, such as those proposed by Benchopt [103], will be instrumental in fostering reproducibility and standardizing performance evaluation. Furthermore, the inclusion of dynamic, multimodal AutoML systems that seamlessly integrate heterogeneous data sources represents an exciting avenue of exploration [2]. Ultimately, AutoML must transcend mere automation, serving as a collaborative system that empowers rather than replaces human expertise, ensuring that its societal impact remains equitable, scalable, and ethically sound [9].

While considerable progress has been made, the journey toward realizing the full potential of AutoML remains ongoing. Bridging the gap between theoretical advances and practical applications will require concerted interdisciplinary efforts. AutoML stands poised not only as a catalyst for technological innovation but also as a framework for rethinking how machine intelligence can interact with human ingenuity across diverse landscapes.

REFERENCES

[1] C. Thornton, F. Hutter, H. Hoos, and K. Leyton-Brown, "Auto-weka: combined selection and hyperparameter optimization of classification algorithms," *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012. 1, 4, 6, 18, 26

[2] R. E. Shawi, M. Maher, and S. Sakr, "Automated machine learning: State-of-the-art and open challenges," *ArXiv*, vol. abs/1906.02287, 2019. 1, 4, 8, 9, 11, 12, 14, 26

[3] M. Zöller and M. F. Huber, "Benchmark and survey of automated machine learning frameworks," *J. Artif. Intell. Res.*, vol. 70, pp. 409–472, 2019. 1, 4, 8, 16

[4] X. He, K. Zhao, and X. Chu, "Automl: A survey of the state-of-the-art," *ArXiv*, vol. abs/1908.00709, 2019. 1, 6, 11, 16, 20, 26

[5] J. Liang, E. Meyerson, B. Hodjat, D. Fink, K. Mutch, and R. Miikkulainen, "Evolutionary neural automl for deep learning," *Proceedings of the Genetic and Evolutionary Computation Conference*, 2019. 1, 4, 5, 6, 8, 9, 11, 13, 14, 16, 23, 25, 26

[6] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *ArXiv*, vol. abs/1611.01236, 2016. 1, 16, 26

[7] B. Celik, P. Singh, and J. Vanschoren, "Online automl: an adaptive automl framework for online learning," *Machine Learning*, vol. 112, pp. 1897–1921, 2022. 1, 2, 6, 10, 11, 16, 21, 26

[8] M. Wang, W. Fu, X. He, S. Hao, and X. Wu, "A survey on large-scale machine learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, pp. 2574–2594, 2020. 1, 26

[9] D. Xin, E. Y. Wu, D. Lee, N. Salehi, and A. G. Parameswaran, "Whither automl? understanding the role of automation in machine learning workflows," *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021. 1, 26

[10] J. Drozdal, J. D. Weisz, D. Wang, G. Dass, B. Yao, C. Zhao, M. J. Muller, L. Ju, and H. Su, "Trust in automl: exploring information needs for establishing trust in automated machine learning systems," *Proceedings of the 25th International Conference on Intelligent User Interfaces*, 2020. 1, 21, 26

[11] E. Real, C. Liang, D. R. So, and Q. V. Le, "Automl-zero: Evolving machine learning algorithms from scratch," in *International Conference on Machine Learning*, 2020, pp. 8007–8019. 2, 6, 18, 21

[12] F. Horn, R. Pack, and M. Rieger, "The autofeat python library for automated feature engineering and selection," in *PKDD/ECML Workshops*, 2019, pp. 111–120. 2, 3, 11, 12, 16, 26

[13] N. Hollmann, S. G. Müller, and F. Hutter, "Large language models for automated data science: Introducing caafe for context-aware automated feature engineering," in *Neural Information Processing Systems*, 2023. 2, 3, 17, 21, 22, 26

[14] Z. Zhang, X. Wang, and W. Zhu, "Automated machine learning on graphs: A survey," in *International Joint Conference on Artificial Intelligence*, 2021, pp. 4704–4712. 2

[15] A. Mumuni and F. Mumuni, "Automated data processing and feature engineering for deep learning and big data applications: a survey," *ArXiv*, vol. abs/2403.11395, 2024. 2, 14

[16] R. S. Olson, N. Bartley, R. Urbanowicz, and J. Moore, "Evaluation of a tree-based pipeline optimization tool for automating data science," *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, 2016. 2, 3, 6, 7, 11, 17, 22, 26

[17] H. Hassani and E. Silva, "The role of chatgpt in data science: How ai-assisted conversational interfaces are revolutionizing the field," *Big Data Cogn. Comput.*, vol. 7, p. 62, 2023. 2, 17

[18] D. Wang, J. Andres, J. D. Weisz, E. Oduor, and C. Dugan, "Autods: Towards human-centered automation of data science," *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021. 2, 3, 12, 17, 24

[19] T. Zhang, Z. Zhang, Z. Fan, H. Luo, F. Liu, L.-Y. D. Liu, Q. Liu, W. Cao, and J. Li, "Openfe: Automated feature generation with expert-level performance," in *International Conference on Machine Learning*, 2022, pp. 41 880–41 901. 3, 14, 17, 18, 22, 24

[20] R. S. Olson, R. Urbanowicz, P. C. Andrews, N. A. Lavender, L. C. Kidd, and J. Moore, "Automating biomedical data science through tree-based pipeline optimization," in *EvoApplications*, 2016, pp. 123–137. 3

[21] A. Quemy, "Two-stage optimization for machine learning workflow," *Inf. Syst.*, vol. 92, p. 101483, 2019. 3

[22] X. Shi, J. W. Mueller, N. Erickson, M. Li, and A. J. Smola, "Benchmarking multimodal automl for tabular data with text fields," *ArXiv*, vol. abs/2111.02705, 2021. 3, 12, 22

[23] M. Feurer, K. Eggensperger, S. Falkner, M. Lindauer, and F. Hutter, "Auto-sklearn 2.0: Hands-free automl via meta-learning," *J. Mach. Learn. Res.*, vol. 23, pp. 261:1–261:61, 2020. 4, 8, 10

[24] J. Parker-Holder, R. Rajan, X. Song, A. Biedenkapp, Y. Miao, T. Eimer, B. Zhang, V. Nguyen, R. Calandra, A. Faust, F. Hutter, and M. Lindauer, "Automated reinforcement learning (autorl): A survey and open problems," *J. Artif. Intell. Res.*, vol. 74, pp. 517–568, 2022. 4, 23

[25] W. Fan, K. Liu, H. Liu, P. Wang, Y. Ge, and Y. Fu, "Autofs: Automated feature selection via diversity-aware interactive reinforcement learning," *2020 IEEE International Conference on Data Mining (ICDM)*, pp. 1008–1013, 2020. 4

[26] E. Real, S. Moore, A. Selle, S. Saxena, Y. Suematsu, J. Tan, Q. V. Le, and A. Kurakin, "Large-scale evolution of image classifiers," in *International Conference on Machine Learning*, 2017, pp. 2902–2911. 4, 5, 14, 18

[27] M. Wistuba, A. Rawat, and T. Pedapati, "A survey on neural architecture search," *ArXiv*, vol. abs/1905.01392, 2019. 4, 8, 13

[28] C. Wong, N. Houlsby, Y. Lu, and A. Gesmundo, "Transfer learning with neural automl," in *Neural Information Processing Systems*, 2018, pp. 8366–8375. 4, 5, 8, 9, 11, 14, 15, 18, 23, 25

[29] Z. Tang, H. Fang, S. Zhou, T. Yang, Z. Zhong, T. Hu, K. Kirchhoff, and G. Karypis, "Autogluon-multimodal (automm): Supercharging multimodal automl with foundation models," *ArXiv*, vol. abs/2404.16233, 2024. 4, 8, 13

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. 4

[31] R. Liaw, E. Liang, R. Nishihara, P. Moritz, J. E. Gonzalez, and I. Stoica, "Tune: A research platform for distributed model selection and training," *ArXiv*, vol. abs/1807.05118, 2018. 4, 14

[32] C. Wang, Q. Wu, M. Weimer, and E. Zhu, "Flaml: A fast and lightweight automl library," in *Conference on Machine Learning and Systems*, 2019. 4, 5, 8, 9, 10, 14, 15, 18, 19, 20, 23, 24, 25

[33] P. Koch, O. Golovidov, S. Gardner, B. Wujek, J. Griffin, and Y. Xu, "Autotune: A derivative-free optimization framework for hyperparameter tuning," *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018. 4, 9, 14, 16

[34] R. Egele, P. Balaprakash, V. Vishwanath, I. M. Guyon, and Z. Liu, "Agebo-tabular: Joint neural architecture and hyperparameter search with autotuned data-parallel training for tabular data," *SC21: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–16, 2020. 5, 9, 15

[35] X. Dong, M. Tan, A. W. Yu, D. Peng, B. Gabrys, and Q. V. Le, "Autohas: Efficient hyperparameter and architecture search." *arXiv: Computer Vision and Pattern Recognition*, 2020. 5, 8, 14, 15, 18, 19, 24

[36] L. Zimmer, M. Lindauer, and F. Hutter, "Auto-pytorch tabular: Multi-fidelity metalearning for efficient and robust autodl," *ArXiv*, vol. abs/2006.13799, 2020. 5, 8, 10, 13, 18, 20, 23

[37] H. Jin, Q. Song, and X. Hu, "Auto-keras: An efficient neural architecture search system," *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018. 5, 8, 14

[38] C. Yang, Y. Akimoto, D. W. Kim, and M. Udell, "Oboe: Collaborative filtering for automl model selection," *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018. 5, 19

[39] S. S. Karmaker, M. M. Hassan, M. J. Smith, L. Xu, C. Zhai, and K. Veeramachaneni, "Automl to date and beyond: Challenges and opportunities," *ACM Computing Surveys (CSUR)*, vol. 54, pp. 1 – 36, 2020. 5, 8, 12, 19, 23

[40] J. Wainer and G. Cawley, "Nested cross-validation when selecting classifiers is overzealous for most practical applications," *ArXiv*, vol. abs/1809.09446, 2018. 5

[41] B. Bischl, M. Binder, M. Lang, T. Pielok, J. Richter, S. Coors, J. Thomas, T. Ullmann, M. Becker, A. Boulesteix, D. Deng, and M. Lindauer, "Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 13, 2021. 5, 6, 10, 16, 19, 21, 24, 26

[42] A. Klein, S. Falkner, S. Bartels, P. Hennig, and F. Hutter, "Fast bayesian optimization of machine learning hyperparameters on large datasets," *ArXiv*, vol. abs/1605.07079, 2016. 5, 10

[43] O. Owoyele, P. Pal, A. V. Torreira, D. Probst, M. Shaxted, M. Wilde, and P. Senecal, "Application of an automated machine learning-genetic algorithm (automl-ga) coupled with computational fluid dynamics simulations for rapid engine design optimization," *International Journal of Engine Research*, vol. 23, pp. 1586 – 1601, 2021. 5, 13

[44] W. Zheng, T. Chen, T.-K. Hu, and Z. Wang, "Symbolic learning to optimize: Towards interpretability and scalability," *ArXiv*, vol. abs/2203.06578, 2022. 5, 24

[45] P. Mitra and F. Biessmann, "Automated computational energy minimization of ml algorithms using constrained bayesian optimization," *ArXiv*, vol. abs/2407.05788, 2024. 6, 10, 15, 24

[46] B. Chen, H. Wu, W. Mo, I. Chattopadhyay, and H. Lipson, "Autostacker: a compositional evolutionary learning system," *Proceedings of the Genetic and Evolutionary Computation Conference*, 2018. 6

[47] J. Vanschoren, "Meta-learning: A survey," *ArXiv*, vol. abs/1810.03548, 2018. 6, 16, 21, 26

[48] P. Gijsbers, E. LeDell, J. Thomas, S. Poirier, B. Bischl, and J. Vanschoren, "An open source automl benchmark," *ArXiv*, vol. abs/1907.00909, 2019. 6

[49] D. Peng, X. Dong, E. Real, M. Tan, Y. Lu, G. Bender, H. Liu, A. Kraft, C. Liang, and Q. V. Le, "Pyglove: Symbolic programming for automated machine learning," *ArXiv*, vol. abs/2101.08809, 2021. 7, 17, 22

[50] N. O. Nikitin, P. Vychuzhanin, M. Sarafanov, I. S. Polonskaia, I. Revin, I. V. Barabanova, G. Maximov, A. V. Kaluzhnaya, and A. Boukhanovsky, "Automated evolutionary approach for the design of composite machine learning pipelines," *ArXiv*, vol. abs/2106.15397, 2021. 7, 13, 14

[51] T. Lee, J. E. Gottschlich, N. Tatbul, E. Metcalf, and S. Zdonik, "Greenhouse: A zero-positive machine learning system for time-series anomaly detection," *ArXiv*, vol. abs/1801.03168, 2018. 7

[52] D. Zha, K.-H. Lai, Q. Tan, S. Ding, N. Zou, and X. Hu, "Towards automated imbalanced learning with deep hierarchical reinforcement learning," *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022. 7, 12, 15, 20, 21, 22

[53] D. Deng, F. Karl, F. Hutter, B. Bischl, and M. Lindauer, "Efficient automated deep learning for time series forecasting," in *ECML/PKDD*, 2022, pp. 664–680. 9

[54] S. H. S. Basha, S. K. Vinakota, V. Pulabaigari, S. Mukherjee, and S. Dubey, "Autotune: Automatically tuning convolutional neural networks for improved transfer learning," *Neural networks : the official journal of the International Neural Network Society*, vol. 133, pp. 112–122, 2020. 9, 23

[55] K. Swersky, J. Snoek, and R. P. Adams, "Freeze-thaw bayesian optimization," *ArXiv*, vol. abs/1406.3896, 2014. 10, 15

[56] X. Zeng and G. Luo, "Progressive sampling-based bayesian optimization for efficient and automatic machine learning model selection," *Health Information Science and Systems*, vol. 5, 2017. 10, 15, 24

[57] Y. Li, Y. Shen, W. Zhang, J. Jiang, B. Ding, Y. Li, J. Zhou, Z. Yang, W. Wu, C. Zhang, and B. Cui, "Volcanoml: speeding up end-to-end automl via scalable search space decomposition," *The VLDB Journal*, vol. 32, pp. 389–413, 2021. 10, 20

[58] B. Celik and J. Vanschoren, "Adaptation strategies for automated machine learning on evolving data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, pp. 3067–3078, 2020. 10, 20

[59] R. Urbanowicz, R. F. Zhang, Y. Cui, and P. Suri, "Streamline: A simple, transparent, end-to-end automated machine learning pipeline facilitating data analysis and algorithm comparison," *ArXiv*, vol. abs/2206.12002, 2022. 10, 16, 17

[60] T. Tornede, A. Tornede, J. Hanselle, M. Wever, F. Mohr, and E. Hullermeier, "Towards green automated machine learning: Status quo and future directions," *ArXiv*, vol. abs/2111.05850, 2021. 11, 16, 20, 25

[61] Z. Yang, W. Zeng, S. Jin, C. Qian, P. Luo, and W. Liu, "Autommlab: Automatically generating deployable models from language instructions for computer vision tasks," *ArXiv*, vol. abs/2402.15351, 2024. 11

[62] A. Balaji and A. Allen, "Benchmarking automatic machine learning frameworks," *ArXiv*, vol. abs/1808.06492, 2018. 11

[63] Q. Wang, Y. Ming, Z. Jin, Q. Shen, D. Liu, M. J. Smith, K. Veeramachaneni, and H. Qu, "Atmseer: Increasing transparency and controllability in automated machine learning," *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019. 11, 26

[64] S. Liu, C. Gao, and Y. Li, "Large language model agent for hyperparameter optimization," *ArXiv*, vol. abs/2402.01881, 2024. 12, 21, 26

[65] E. Angelino, N. Larus-Stone, D. Alabi, M. Seltzer, and C. Rudin, "Learning certifiably optimal rule lists," *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017. 12, 17

[66] J. Thomas, S. Coors, and B. Bischl, "Automatic gradient boosting," *ArXiv*, vol. abs/1807.03873, 2018. 12

[67] S. Meisenbacher, M. Turowski, K. Phipps, M. Ratz, D. Muller, V. Hagenmeyer, and R. Mikut, "Review of automated time series forecasting pipelines," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 12, 2022. 12, 22

[68] E. Papalexakis, "Automatic unsupervised tensor mining with quality assessment," *ArXiv*, vol. abs/1503.03355, 2015. 13

[69] U. Khurana, H. Samulowitz, and D. Turaga, "Feature engineering for predictive modeling using reinforcement learning," *ArXiv*, vol. abs/1709.07150, 2017. 13, 18

[70] Y. Li, Z. Chen, D. Zha, K. Zhou, H. Jin, H. Chen, and X. Hu, "Autood: Automated outlier detection via curiosity-guided search and self-imitation learning," *ArXiv*, vol. abs/2006.11321, 2020. 14, 22

[71] D. Luo, C. Feng, Y. Nong, and Y. Shen, "Autom3l: An automated multimodal machine learning framework with large language models," *ArXiv*, vol. abs/2408.00665, 2024. 14

[72] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. Sidiropoulos, "Learning to optimize: Training deep neural networks for interference management," *IEEE Transactions on Signal Processing*, vol. 66, pp. 5438–5453, 2017. 14, 20

[73] J. N. Rijn and F. Hutter, "Hyperparameter importance across datasets," *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2017. 15, 19, 24

[74] X. Dong, D. Kedziora, K. Musial, and B. Gabrys, "Automated deep learning: Neural architecture search is not the end," *ArXiv*, vol. abs/2112.09245, 2021. 15

[75] N. Erickson, J. W. Mueller, A. Shirkov, H. Zhang, P. Larroy, M. Li, and A. Smola, "Autogluon-tabular: Robust and accurate automl for structured data," *ArXiv*, vol. abs/2003.06505, 2020. 15, 18, 20, 22

[76] R. Knauer and E. Rodner, "Squeezing lemons with hammers: An evaluation of automl and tabular deep learning for data-scarce classification applications," *ArXiv*, vol. abs/2405.07662, 2024. 15

[77] A. Santos, S. Castelo, C. Felix, J. P. Ono, B. Yu, S. Hong, C. T. Silva, E. Bertini, and J. Freire, "Visus: An interactive system for automatic machine learning model building and curation," *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, 2019. 17

[78] M. Sipper, "Binary and multinomial classification through evolutionary symbolic regression," *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2022. 17, 24

[79] A. Bauerle, H. Neumann, and T. Ropinski, "Classifier-guided visual correction of noisy labels for image classification tasks," *Computer Graphics Forum*, vol. 39, 2018. 17

[80] R. Schönhof, A. Werner, J. Elstner, B. Zopcsak, R. Awad, and M. F. Huber, "Feature visualization within an automated design assessment leveraging explainable artificial intelligence methods," *ArXiv*, vol. abs/2201.12107, 2022. 18

[81] E. Ozturk, F. Ferreira, H. Jomaa, L. Schmidt-Thieme, J. Grabocka, and F. Hutter, "Zero-shot automl with pretrained models," in *International Conference on Machine Learning*, 2022, pp. 17 138–17 155. 18

[82] S. Adriaensen, A. Biedenkapp, G. Shala, N. H. Awad, T. Eimer, M. Lindauer, and F. Hutter, "Automated dynamic algorithm configuration," *J. Artif. Intell. Res.*, vol. 75, pp. 1633–1699, 2022. 19

[83] Y.-W. Chen, Q. Song, and X. Hu, "Techniques for automated machine learning," *ACM SIGKDD Explorations Newsletter*, vol. 22, pp. 35 – 50, 2019. 19, 20

[84] Y. Li, J. Jiang, J. Gao, Y. Shao, C. Zhang, and B. Cui, "Efficient automatic cash via rising bandits," in *AAAI Conference on Artificial Intelligence*, 2020, pp. 4763–4771. 19

[85] R. G. Mantovani, T. Horváth, R. Cerri, S. B. Junior, J. Vanschoren, and A. Carvalho, "Better trees: an empirical study on hyperparameter tuning of classification decision tree induction algorithms," *Data Min. Knowl. Discov.*, vol. 38, pp. 1364–1416, 2018. 19

[86] M. Claesen, J. Simm, D. Popovic, Y. Moreau, and B. Moor, "Easy hyperparameter search using optunity," *ArXiv*, vol. abs/1412.1114, 2014. 19

[87] M. Lindauer, K. Eggensperger, M. Feurer, A. Biedenkapp, D. Deng, C. Benjamins, R. Sass, and F. Hutter, "Smac3: A versatile bayesian optimization package for hyperparameter optimization," *ArXiv*, vol. abs/2109.09831, 2021. 19

[88] J. C. Barsce, J. Palombarini, and E. Martínez, "Towards autonomous reinforcement learning: Automatic setting of hyperparameters using bayesian optimization," *2017 XLIII Latin American Computer Conference (CLEI)*, pp. 1–9, 2017. 20, 24

[89] H. Escalante, "Automated machine learning - a brief review at the end of the early years," in *Automated Design of Machine Learning and Search Algorithms*, 2020, pp. 11–28. 20

[90] S. Krishnan, M. Franklin, K. Goldberg, and E. Wu, "Boostclean: Automated error detection and repair for machine learning," *ArXiv*, vol. abs/1711.01299, 2017. 20

[91] Y. Sun, Q. Song, X. Gui, F. Ma, and T. Wang, "Automl in the wild: Obstacles, workarounds, and expectations," *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023. 20

[92] M. Zöller, W. Titov, T. Schlegel, and M. F. Huber, "Xautoml: A visual analytics tool for understanding and validating automated machine learning," *ACM Transactions on Interactive Intelligent Systems*, vol. 13, pp. 1 – 39, 2022. 20

[93] R. Li, T. Patel, Q. Wang, Q. Wang, and X. Du, "Mlr-copilot: Autonomous machine learning research based on large language models agents," *ArXiv*, vol. abs/2408.14033, 2024. 21

[94] R. K. Saha, A. Ura, S. Mahajan, C. Zhu, L. Li, Y. Hu, H. Yoshida, S. Khurshid, and M. Prasad, "Sapientml: Synthesizing machine learning pipelines by learning from human-written solutions," *2022 IEEE/ACM 44th International Conference on Software Engineering (ICSE)*, pp. 1932–1944, 2022. 22

[95] L. Yang and A. Shami, "Iot data analytics in dynamic environments: From an automated machine learning perspective," *ArXiv*, vol. abs/2209.08018, 2022. 22

[96] Y. Li, Y. Shen, W. Zhang, J. Jiang, B. Ding, Y. Li, J. Zhou, Z. Yang, W. Wu, C. Zhang, and B. Cui, "Volcanoml: speeding up end-to-end automl via scalable search space decomposition," *The VLDB Journal*, vol. 32, pp. 389–413, 2021. 23

[97] S. H. S. Basha, S. K. Vinakota, S. Dubey, V. Pulabaigari, and S. Mukherjee, "Autofcl: automatically tuning fully connected layers for handling small dataset," *Neural Computing and Applications*, vol. 33, pp. 8055 – 8065, 2020. 23

[98] S. Zhang, C. Gong, L. Wu, X. Liu, and M. Zhou, "Automl-gpt: Automatic machine learning with gpt," *ArXiv*, vol. abs/2305.02499, 2023. 24

[99] F. Neutatz, M. Lindauer, and Z. Abedjan, "Automl in heavily constrained applications," *VLDB J.*, vol. 33, pp. 957–979, 2023. 24

[100] M. Zecchin and O. Simeone, "Adaptive learn-then-test: Statistically valid and efficient hyperparameter selection," *ArXiv*, vol. abs/2409.15844, 2024. 25

[101] A. Mohan, C. Benjamins, K. Wienecke, A. Dockhorn, and M. Lindauer, "Autorl hyperparameter landscapes," in *AutoML*, 2023, pp. 13/1–27. 25

[102] A. Tornede, D. Deng, T. Eimer, J. Giovanelli, A. Mohan, T. Ruhkopf, S. Segel, D. Theodorakopoulos, T. Tornede, H. Wachsmuth, and M. Lindauer, "Automl in the age of large language models: Current challenges, future opportunities and risks," *ArXiv*, vol. abs/2306.08107, 2023. 25

[103] T. Moreau, M. Massias, A. Gramfort, P. Ablin, P.-A. B. B. Charlier, M. Dagr'eou, T. D. la Tour, G. Durif, C. Dantas, Q. Klopfenstein,

J. Larsson, E.-H. Lai, T. Lefort, B. Mal'ezieux, B. Moufad, B. D. Nguyen, A. Rakotomamonjy, Z. Ramzi, J. Salmon, and S. Vaiter, "Benchopt: Reproducible, efficient and collaborative optimization benchmarks," *ArXiv*, vol. abs/2206.13424, 2022. 26