Published: 09 December 2025
Online AM: 25 June 2025
Accepted: 21 May 2025
Revised: 09 May 2025
Received: 23 September 2024

RESEARCH-ARTICLE

Practitioner Motives to Use Different Hyperparameter Optimization Methods

NICLAS KANNENGIEßER, Karlsruhe Institute of Technology, Karlsruhe, Baden-Wurttemberg, Germany

NIKLAS HASEBROOK

FELIX MORSBACH, Karlsruhe Institute of Technology, Karlsruhe, Baden-Wurttemberg, Germany

MARC ANDRÉ ZÖLLER, University of Stuttgart, Stuttgart, Baden-Wurttemberg, Germany

JÖRG K H FRANKE, University of Freiburg, Freiburg im Breisgau, Baden-Wurttemberg, Germany

MARIUS LINDAUER, Gottfried Wilhelm Leibniz University Hannover, Hannover, Niedersachsen, Germany

View all

Open Access Support provided by:

Karlsruhe Institute of Technology

Gottfried Wilhelm Leibniz University Hannover

Technical University of Munich

University of Stuttgart

University of Freiburg

ACM Transactions on Computer-Human Interaction, Volume 32, Issue 6 (December 2025)
https://doi.org/10.1145/3745771
EISSN: 1557-7325

Practitioner Motives to Use Different Hyperparameter Optimization Methods

NICLAS KANNENGIEẞER, Institute of Applied Informatics and Formal Description Methods, Karlsruhe Institute of Technology, Karlsruhe, Germany
NIKLAS HASEBROOK, Zeb Consulting, Berlin, Germany
FELIX MORSBACH, Chair of Privacy and Security, Karlsruhe Institute of Technology, Karlsruhe, Germany
MARC-ANDRÉ ZÖLLER, Institute of Industrial Manufacturing and Management, University of Stuttgart, Stuttgart, Germany
JÖRG K. H. FRANKE, Machine Learning Lab, University of Freiburg, Freiburg im Breisgau, Germany
MARIUS LINDAUER, Institute of Artificial Intelligence, Leibniz University Hannover, Hannover, Germany
FRANK HUTTER, Machine Learning Lab, University of Freiburg, Freiburg im Breisgau, Germany
ALI SUNYAEV, TUM School of Computation, Information and Technology, Technical University of Munich, Heilbronn, Germany

Programmatic hyperparameter optimization (HPO) methods, such as Bayesian optimization and evolutionary algorithms, are known for their sample efficiency in identifying optimal configurations for machine learning (ML) models. However, practitioners often use less efficient methods, such as grid search, potentially resulting in under-optimized models. This discrepancy suggests that HPO method selection may be influenced by practitioner-specific motives, which remain insufficiently understood hindering user-centered advancement of HPO tools. To uncover these motives, we conducted 20 semi-structured interviews and an online survey with 49 ML practitioners. We revealed six primary goals (e.g., increasing ML model understanding) and 14 contextual factors (e.g., available computational resources) that influence practitioners' choices of HPO

Funded by the European Union

This work was supported by KASTEL Security Research Labs. F. Hutter and M. Lindauer acknowledge funding by the European Union (via ERC Consolidator Grant DeepLearning 2.0, grant no. 101045765, and ERC Starting Grant "ixAutoML," grant no. 101041029, respectively). The views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.
Authors' Contact Information: Niclas Kannengießer (corresponding author), Institute of Applied Informatics and Formal Description Methods, Karlsruhe Institute of Technology, Karlsruhe, Germany; e-mail: niclas.kannengiesser@kit.edu; Niklas Hasebrook, Zeb Consulting, Berlin, Germany; e-mail: niklas.hasebrook@gmail.com; Felix Morsbach, Chair of Privacy and Security, Karlsruhe Institute of Technology, Karlsruhe, Germany; e-mail: felix.morsbach@kit.edu; Marc-André Zöller, Institute of Industrial Manufacturing and Management, University of Stuttgart, Stuttgart, Germany; e-mail: mazoeller@gmail.com; Jörg K. H. Franke, Machine Learning Lab, University of Freiburg, Freiburg im Breisgau, Germany; e-mail: frankej@cs.uni-freiburg.de; Marius Lindauer, Institute of Artificial Intelligence, Leibniz University Hannover, Hannover, Germany; e-mail: m.lindauer@ai.uni-hannover.de; Frank Hutter, Machine Learning Lab, University of Freiburg, Freiburg im Breisgau, Germany; e-mail: fh@cs.uni-freiburg.de; Ali Sunyaev, TUM School of Computation, Information and Technology, Technical University of Munich, Heilbronn, Germany; e-mail: sunyaev@tum.de.

ACM 1557-7325/2025/12-ART59
https://doi.org/10.1145/3745771

methods. This study provides a conceptual foundation for understanding real-world HPO practices and informs the development of more user-centered and context-adaptive HPO tools in automated ML (AutoML).

CCS Concepts: • **Human-centered computing → Empirical studies in HCI**; HCI theory, concepts and models; *User models*;

Additional Key Words and Phrases: Artificial Intelligence (AI), Automated Machine Learning (AutoML), Human-AI Collaboration, Hyperparameter Optimization (HPO), User-centered HPO

ACM Reference format:
Niclas Kannengießer, Niklas Hasebrook, Felix Morsbach, Marc-André Zöller, Jörg K. H. Franke, Marius Lindauer, Frank Hutter, and Ali Sunyaev. 2025. Practitioner Motives to Use Different Hyperparameter Optimization Methods. *ACM Trans. Comput.-Hum. Interact.* 32, 6, Article 59 (December 2025), 33 pages.
https://doi.org/10.1145/3745771

1 Introduction

The performance of **machine learning (ML)** models is highly sensitive to their hyperparameter configurations [3, 13, 38, 47, 60, 89]. Identifying optimal hyperparameter configurations for training ML models is, however, a complex and often daunting task—even for seasoned ML experts—because of large search spaces of hyperparameter values and commonly unknown relationships between ML model performance, hyperparameter configuration, and dataset. Consequently, practitioners—individuals regularly involved in the development of viable ML models that are meant for productive use in research or industry—experiment with various hyperparameter configurations to identify optimal ones, often relying on trial and error. This iterative process of exploring, testing, and adjusting hyperparameter configurations related to ML models is known as **hyperparameter optimization (HPO)**.

HPO by manual tuning is often cumbersome, tedious, and error prone. To help practitioners with HPO, research with a technical focus (e.g., [8, 22]) developed several programmatic HPO methods, including grid search, random search, Bayesian optimization, and evolutionary algorithms, and implemented such methods as software tools (e.g., Hyperopt and Hyperband [4, 54]). Existing HPO methods, programmatic and non-programmatic ones, differ considerably in the way they optimize hyperparameter configurations. Such differences lead some HPO methods to superiority over others. That superiority is usually demonstrated using conventional performance metrics from computer science, including minimization of generalization errors and increase of sample efficiency [30, 55, 90]. However, practitioners often use HPO methods that are inferior according to conventional performance metrics [9]. For example, practitioners often prefer to perform grid search over the more sample-efficient Bayesian optimization [74].

The dominant use of seemingly inferior HPO methods suggests that practitioners have motives beyond the fulfillment of conventional goals pursued in HPO, such as increasing ML model performance. However, the motives behind practitioners' choices of HPO methods—shaped by contextual factors—remain unclear. This unclarity about why practitioners use which HPO methods inhibits more user-centered development of HPO methods and tools that support practitioners in attaining their goals, especially those beyond conventional performance metrics. To support development of more user-centered HPO methods and tools for **automated ML (AutoML)** [81], practitioner motives for HPO need to be better understood. We approach the following research question: *Why do practitioners choose different HPO methods?*

We applied a two-step research approach consisting of an interview study and a survey study based on an online questionnaire. First, we conducted semi-structured interviews with 20 ML experts to unveil HPO methods that practitioners commonly use, the goals pursued by practitioners when applying HPO methods, and the contextual factors that influence practitioners' choices of HPO methods to attain goals. Second, we performed an online survey with 49 participants to collect evidence for the external validity of the relevance of the HPO methods, goals, and contextual factors identified in the interviews.

Our main ambition is to support more user-centric development of AutoML methods and tools for HPO by bridging the gap between technical advancements and human factors. By integrating practitioner motives into HPO research, we aim to ensure that future tools are not only technically sound but also aligned with real-world needs and decision-making processes. In particular, this work has three main contributions. First, this work presents a conceptual foundation constituted of principal goals (e.g., improving ML model performance and target audience compliance) and contextual factors (e.g., compute resources and method traceability) influencing selections of HPO methods. This is useful to enhance human-in-the-loop ML by clarifying information needs and improving practitioner engagement. Moreover, the conceptual foundation supports goal-driven development of HPO methods and tools and new benchmarks with a focus on human factors. Second, we present a mapping of HPO methods, goals, and contextual factors to explain practitioners' decision-making. This mapping informs user-centered HPO design and adaptive automation features. Moreover, it highlights key input parameters for developing context-sensitive HPO tools. Third, we analyzed practitioners' perceived success with different HPO methods, revealing strengths and areas for improvement. These findings contribute to the development of more effective decision-support systems for HPO method selection. Moreover, the reported success rates help identify limitations of HPO methods, guiding development of better-tailored and more effective HPO methods and tools.

The remainder of this work is structured into five sections. First, we describe the state of research on HPO in Section 2. Section 3 reports the approach we applied to answer the research question. Then, we present the results of this study, including four HPO methods, six goals, and 14 contextual factors in Section 4. In Section 5, we discuss the principal findings of this work, explain the contributions of this study, describe possible threats to the validity of the results, and outline future research directions. We conclude with our main takeaways in Section 6.

2 Background and Related Work

Driven largely by technical advancements, HPO is a key research area with significant potential to shape ML model development [43]. To support a better understanding of the results of this study, we briefly describe the technical foundations of principal HPO methods. Moreover, we elucidate related research on involving human beings in automated HPO, a prominent field within AutoML research.

2.1 HPO Methods

There are five principal, model-agnostic HPO methods: manual tuning, grid search, random search, evolutionary algorithms, and Bayesian optimization.[1] These HPO methods are briefly described below.

Manual tuning refers to a set of HPO methods where practitioners choose hyperparameter configurations based on explicit and implicit knowledge and influences of contextual factors. The dependency of manual tuning on practitioners' experiences and even unconscious heuristics in decisions [28] makes manual tuning very individual to practitioners, rendering the explication of manual tuning difficult. Thus, manual tuning is usually hardly replicable [63, 73]. Commonly,

[1] There are also gradient-based methods, but because these methods are often perceived as being brittle and are often model-specific, they are out of the scope of this study.

only intermediate information (e.g., tuned hyperparameters) is available, while reasons for selecting hyperparameter configurations often remain unclear. This complicates the formalization of specific HPO methods in manual tuning, making it unclear how practitioners actually proceed. Common strategies include starting optimization from well-performing hyperparameter values [87], using a sound experimental design for incremental improvement [32], and removing irrelevant hyperparameters from the search space [66]. In addition, difficulty in explication makes it hard to evaluate the sample efficiency of strategies in manual tuning. While evidence that manual tuning outperforms advanced HPO methods is lacking, prior publications offer initial evidence that advanced HPO methods can outperform manual tuning in several use cases [13, 23, 47, 60, 89, 91].

To tackle shortcomings of manual tuning, AutoML [22] is envisioned to automate all aspects related to the development of ML models in a problem-agnostic manner, for example, by programmatic HPO methods. Research on AutoML has primarily approached HPO from a technical perspective (e.g., [8, 22]). Typical works (e.g., [2, 21, 34, 37, 45, 54, 82]) focus on performance optimization of ML models in terms of smaller generalization errors, smaller ML model size, or lower latency. Often, such works investigate HPO from a mathematical perspective and treat HPO as a black-box optimization problem: given a problem instance in the form of a dataset and a loss function, a black-box optimizer (e.g., Bayesian optimization) searches for hyperparameter configurations in a predefined search space to enhance an ML model in terms of a given metric (e.g., accuracy on a validation set). Multi-objective optimization methods can be used to specify additional properties of the resulting ML model, such as algorithmic fairness, fast inference, and low model complexity [7, 16, 26, 49].

Grid search is one of the earliest HPO methods that can be executed programmatically to solve the black-box optimization problem. Grid search refers to the process of evaluating the Cartesian product of a finite set of hyperparameter configurations. Every possible combination of hyperparameter values included in the defined subset of the search space is evaluated [61]. Thus, grid search does not scale well with the number of hyperparameters. Grid search relies on a deterministic procedure to select hyperparameter configurations to be evaluated. The deterministic procedure allows reproducing experiments. For reproduction, the originally applied search space and discretization strategy must be known.

Although easy to implement, parallelize, and reproduce, grid search has become increasingly unsuited for modern HPO problems due to the curse of dimensionality [3]. In practice, not all hyperparameters have a similar impact on the final performance of ML models [3, 68]. Due to its rigid search strategy, grid search often allocates much of the optimization budget to less relevant regions of the search space [3]. Put differently, sample efficiency of grid search tends to be lower compared to later HPO methods (e.g., random search, Bayesian optimization, and evolutionary algorithms [19, 74, 80]), in particular, because grid search cannot make use of the low effective dimensionality of HPO problems [2].

Random search refers to the process of sampling random hyperparameter configurations from a defined search space until a budget is exhausted [3]. Random search handles hyperparameters of varying importance more effectively than grid search and achieves greater sample efficiency in high-dimensional search spaces, especially when hyperparameters have differing influence on ML model performance [3]. Random search can be reproduced if the used search space, the randomness generator, and the corresponding seed are known.

To address large search spaces, grid search and random search have been supplemented by methods that exploit knowledge of well-performing regions within the set of possible hyperparameter configurations [74]. A well-established approach for balancing exploration and exploitation is evolutionary optimization. Inspired by biological evolution, *evolutionary algorithms* iteratively mutate a population of candidate solutions to obtain solutions with better performance. Evolutionary algorithms often perform well in optimizing black-box functions [65] but are rather inefficient in terms of samples [8]. HPO based on evolutionary algorithms can be reproduced with fixed random seeds if the search space and randomness generator are known.

Bayesian optimization can be an alternative to evolutionary algorithms. Bayesian optimization refers to the process of using a sequential approach based on a surrogate model to find appropriate hyperparameter configurations for ML models in defined search spaces (e.g., [12, 22, 27, 72]). In Bayesian optimization, an optimizer constructs an internal probabilistic model, mapping hyperparameter configurations to expected ML model performance, to achieve an optimal balance between exploration and exploitation [24, 27, 72]. Bayesian optimization can be extended to deal with high-dimensional search spaces, for example, using additive surrogate models [48] or local trust regions [20]. HPO based on Bayesian optimization can be reproduced with fixed random seeds if the search space, the acquisition function, and the surrogate model, including its hyperparameters, are known.

2.2 Research on HPO

Extant research on HPO methods is mainly driven by technological advancements that aim at increasing sample efficiency of HPO methods [2, 42, 74], reducing time for evaluating objective functions [15, 54, 79], and transferring knowledge from prior optimization runs to similar problem instances [18, 81]. Such technological advancements are valuable but designed to reach better results in terms of conventional performance metrics. Practitioner motives to use HPO methods beyond conventional performance metrics are largely neglected.

Studies exploring practitioners' experiences with programmatic HPO methods provide valuable insights into the perceived advantages and disadvantages of these methods [31, 84, 85]. Practitioners acknowledge advantages of programmatic HPO methods, which are commonly related to faster turn-around time for building ML models and, thus, higher productivity in developing ML models [83, 85]. Automatically tuned ML models are often used by practitioners to create initial baseline ML models for subsequent manual tuning or to gain data insights [14, 83, 85]. However, AutoML practitioners often bemoan insufficient confidence in the results of programmatic HPO methods and, therefore, refuse to blindly use ML models optimized with programmatic HPO methods [17, 50, 51, 85]. Even though acknowledging that programmatic HPO is useful to develop well-performing ML models [83], practitioners often refuse to use those HPO methods to not be accountable for ML models they do not understand [17]. A lack of confidence is often linked to the perceived black-box nature of programmatic HPO methods, such as Bayesian optimization and evolutionary algorithms, limiting transparency of optimizer internals. Practitioners prefer support that augments their daily data science work (e.g., through guidance), rather than fully automating it [14].

Research on human-guided HPO focuses on involving humans in programmatic HPO methods to improve HPO with dormant domain expertise [40, 85, 86]. This requires identifying how and when to involve humans in HPO to achieve the best combination of automation and human knowledge [14]. Especially involvement of practitioners in ML model development, including HPO, seems promising for a higher level of automation. For other tasks, including data acquisition and requirement analysis, practitioners prefer strong human involvement with a low level of automation [84]. Interactions of practitioners with software tools for programmatic HPO were structured into different modes of cooperation between practitioners and software tools, ranging from manual tuning to full automation, in the literature [14, 51, 84].

Extant research describes valuable concepts of how practitioners could interact with software tools for programmatic HPO [78, 87] and how to design visual analytics tools for HPO to support

Table 1. Overview of the Demographic Data of the 20 Interviewees

Field	Highest Degree of Education	Years of Experience	Skill Level	ML Field
Academia (14)	Bachelor (2)	< 2 (4)	ML Innovator (10)	CV (8)
Industry (6)	Master (16)	2–4 (6)	ML Engineer (10)	NLP (6)
	PhD (2)	5–7 (7)		RL (5)
		> 7 (3)		TSF (3)

The numbers in parentheses show the number of interviewees with the respective characteristics. The interviewees could name multiple ML fields. CV, Computer vision; NLP, Natural language processing; RL, Reinforcement learning; TSF, Time series forecasting.

practitioners [40, 86, 92]. Yet, the different programmatic HPO methods are not further differentiated, and practitioner motives to select different HPO methods remain unclear. Supporting better understanding of practitioner motives for selecting HPO methods is the main goal of this study.

3 Methods

We applied a mixed-methods research approach consisting of two main steps. First, we conducted semi-structured interviews with ML experts to develop a set of commonly used HPO methods, goals pursued by using HPO methods, and contextual factors that influence the choice for HPO methods. Second, we conducted a survey using an online questionnaire to collect evidence of the external validity of the interviews. The following details the two steps.

3.1 Semi-Structured Interviews with ML Experts

To identify practitioners' goals pursued in HPO and understand decisions for specific HPO methods to achieve these goals, we chose an exploratory, qualitative research approach and conducted semi-structured expert interviews.

Data Collection. To find interviewees for the study, we reached out to personal contacts from ongoing research projects, authors of scientific studies, and companies that develop ML models. The contacted persons had heterogeneous experiences with HPO and ML, ranging from novices to experts and different ML fields, including **computer vision (CV)**, **natural language processing (NLP)**, and **reinforcement learning (RL)**. Among the contacted potential interviewees, 20 agreed to participate in the interview study.

The interviewed experts were all ML practitioners—individuals who regularly develop ML models for practical use in research or industry. All interviewees had actively contributed to at least one successfully deployed ML model in research or industry. The experts were associated with 13 different organizations and had an average work experience in ML of about 5 years (see Table 1). Among the interviewees, two held a PhD and 12 were PhD students from academia with Master's degrees—most nearing completion of their doctorates. All industry participants held at least a Master's degree and had, on average, more than 5 years of experience developing ML models for production. The 12 PhD students worked at universities in the field of applied ML. One PhD worked in the industry as an ML engineer, and another worked at a university in the field of applied ML. Two interviewees with Bachelor's degrees were pursuing Master's degrees while working as ML developers in the industry as student trainees.

Ten participants had a skill level of ML innovators—practitioners engaged in ML research, including studies on core algorithms and the application of ML in scientific discoveries [87]—while the other 10 were ML engineers—experienced ML practitioners with formal training in applied ML

[87]. The study participants used HPO in the context of CV, NLP, RL, and **time series forecasting (TSF)**. Interviewees from the industry also mentioned to have used these ML techniques in bioinformatics (2), robotics (2), e-commerce (1), and finance (1). Because AutoML, including HPO based on programmatic methods, aims to be domain agnostic [22], usage of AutoML tools should be independent of the different domains.

We developed an interview guide (see Appendix A) for the semi-structured interviews [35, 64]. The interview guide structured the interviews into four sections: *briefing*, *HPO in ML*, *participant background and personal experiences*, and *debriefing*. Each section outlined its purpose and the corresponding interview questions. We sent the interview guide to participants in advance to help them prepare for the interview. Since the interviews focused on a past ML project, we asked each participant to select one project to discuss during the interview. To mitigate social desirability bias—especially since some participants were recruited via personal contacts—we ensured that interviewers were not personally acquainted with the interviewees. Moreover, we phrased the questions as neutral and open-ended to encourage unbiased and detailed responses. For each interview, we created a non-judgmental atmosphere and clarified that there were only valid answers to our questions with insights valuable to our research. We also explained to the interviewees that their responses would be anonymized and no links to their identity would be possible.

We started each interview with a briefing of the participants by explaining the background and goals of the study. Second, in line with the section *HPO in ML* in the interview guide, we asked the ML experts to name HPO methods they used in ML projects. Moreover, we gathered insights into why they selected HPO methods for optimization and asked about contextual factors that influenced their selection of HPO methods. Third, we wanted to learn more about the interviewees' *participant background and personal experiences*, including years of experience in ML development, main areas of using ML, and highest degree of education. Fourth, in *debriefing*, we asked the interviewees for their final thoughts and remarks on the study and topic, and informed them about the further proceeding in the context of the study. We conducted interviews with ML experts following methodological guidelines, ensuring an open-minded approach without pressure on the interviewee and avoiding any influence on their responses through neutral questions [35, 57, 59]. The 20 interviews took between 18 and 61 minutes, with an average time of 31 minutes. We transcribed the interviews in preparation for the analysis.

Data Analysis. We analyzed each interview transcript using thematic analysis [10, 11] in groups of three authors. Thematic analysis comprises six steps: (1) familiarize yourself with the data, (2) generate initial codes, (3) search for themes, (4) review themes, (5) define and name themes, and (6) produce the report.

Three of the authors coded each interview together. After familiarizing themselves with the transcripts (Step 1), the authors independently coded them (Step 2) to identify HPO methods applied by practitioners, to extract practitioners' goals in HPO and to reveal contextual factors that influence the interviewees' decisions for HPO methods. We incorporated contextual factors to better understand influences on practitioners' decisions for using HPO methods. The three authors independently read the transcripts, identified quotes relevant to this study, and labeled the quote with a name (i.e., a code) that expresses a potentially relevant HPO method, goal, or contextual factor. Each of the three authors coded the transcripts independently without a predefined coding scheme. After coding each interview independently, the authors presented their coding results to each other and discussed and harmonized their results in multiple iterations per interview and across interviews. In each iteration, the analysts aimed for mutually exclusive codes while ensuring comprehensive coverage. Through this explorative approach involving three analysts per interview, we aimed to reduce biases arising from the opinions of single analysts. The first coding iteration revealed 241 preliminary codes.

The authors harmonized their codes into 21 mutually exclusive codes to ensure that different codes did not share the same semantics. For example, the coders merged the contextual factors *knowledge about Bayesian optimization* and *knowledge about grid search* into the contextual factor *HPO method comprehension*. During the harmonization process, the authors aimed for unambiguous agreements regarding the codes and their intended meaning. To achieve this goal, the authors resolved conflicts in their coding results through intensive discussion and refinement.

Three of the authors developed candidate themes (Step 3) to group the harmonized codes based on semantic relationships. If a code did not suit an existing theme, we created a new theme. For example, we assigned the contextual factor *available compute resources* to the theme *technical environment*, while we created a new theme *own knowledge* for the contextual factor *HPO method comprehension*. The set of candidate themes was comprised of four themes associated with HPO methods, six themes associated with goals, and three themes associated with contextual factors.

In Step 4, we reviewed and refined the candidate themes within the author team in multiple iterations. We again aimed to reach mutual exclusiveness of the themes and assigned codes and maintain exhaustiveness of the results. Subsequently, we developed an intuitive name for each theme and a definition (Step 5). Finally, we assigned the set of 13 themes to three categories: HPO methods, principal goals, and contextual factors and wrote up a summary of the results (Step 6).

After coding the transcripts of all 20 interviews, the analysis of the last seven transcripts did not reveal additional HPO methods, goals, and contextual factors. We assume to have reached theoretical saturation [25, 36] after the first 13 interviews with seven interviews confirming this assumption. These last seven interviews analyzed were from three practitioners from industry and four from academia who hold a master's degree and are pursuing a PhD in the field of applied ML. Given the diverse backgrounds of the interviewees and the substantial number of confirmatory interviews, we deemed the results to be sufficiently robust and moved on with an online survey to collect evidence for the external validity of the coding results.

3.2 Online Survey

We conducted a survey study using an online questionnaire to collect evidence for the external validity of the interview study results and to learn whether practitioners perceive that they succeeded in achieving their goals through their decisions to use specific HPO methods.

Questionnaire Structure. The online questionnaire was structured into four sections: *Introduction*, *HPO Methods and Goals*, *Contextual Factor Integration*, and *Demographics*. In the *Introduction* section, we described the motivation for and the structure of the questionnaire. In *Methods and Goals*, we showed participants a matrix that listed all goals and HPO methods identified in the interview study. Participants were asked to select all pairs of HPO methods and goals to achieve specific goals. Because the main purpose of the survey was to collect evidence for the validity of the results from the interview study, study participants were restricted to selecting only HPO methods (i.e., manual tuning, grid search, random search, Bayesian optimization, and evolutionary algorithms) and goals (e.g., increase ML model understanding, decrease necessary computations, and decrease practitioner effort) identified in the previous interview study. In a second question, we asked participants to indicate, for each selected pair of HPO method and goal, whether they feel to have successfully achieved each goal.

In the *Contextual Factor Integration* section, we wanted to better understand how practitioners perceived the influence of contextual factors identified in the interview study on their selections of HPO methods. For each pair of HPO methods and goals previously selected, we asked the

Table 2. Overview of the Demographic Data of the 49 Participants That Completed the Online Questionnaire

Field	Highest Degree of Education	Years of Experience	Skill Level	ML Field
Academia (27)	High school (1)	<2 (8)	ML Innovator (22)	CV (12)
Industry (22)	Bachelor (2)	2–4 (20)	ML Engineer (19)	TD (19)
	Master (27)	5–7 (14)	Novice (8)	NLP (15)
	Diploma (2)	8–10 (4)		TSF (8)
	PhD (17)	>10 (3)		RL (5)

TD, tabular data.
The numbers in parentheses show the number of interviewees with the respective characteristics. The participants could choose multiple ML fields.

participants to express their perceived influence of each contextual factor on the selection of an HPO method to achieve a particular goal on a five-point Likert scale—0 represents very low perceived influence, 2 corresponds to a neutral response (i.e., the contextual factor was not perceived as influential), and 4 represents very high perceived influence.

Finally, in the *Demographics* section, we collected information about the participants to provide context to the data collected.

Data Gathering. To solicit participants for the online questionnaire, we contacted practitioners via e-mail and promoted the study via social media platforms. We did not invite participants from the interview study.

A total of 166 participants responded to the initial question in the *HPO Methods and Goals* section. Among them, 85 completed the entire section, and 57 completed the *Contextual Factor Integration* section. Of these, 49 provided demographic data and completed the full questionnaire. Most of these 49 participants worked in large organizations with more than 500 employees, including automotive companies, companies specializing in IT support and services, and universities. Table 2 shows more demographic details about the participants who completed the questionnaire.

Data Analysis. By analyzing the responses to the online questionnaire, we sought to learn how frequently practitioners tend to choose which HPO methods to pursue specific goals, given which contextual factors. To prepare the analysis, we discarded all responses from participants who did not complete the questionnaire. We only analyzed completed survey responses. We extracted the number of identical responses and related them to each other.

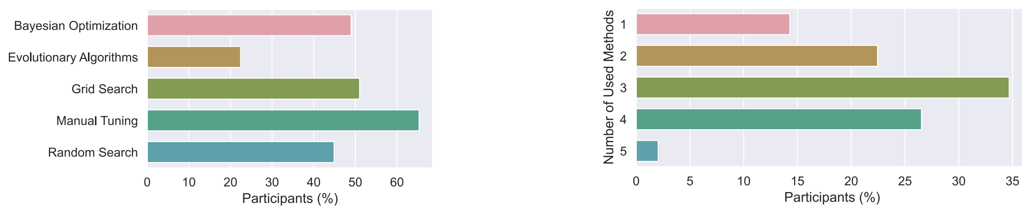
4 Results

The study participants (i.e., the interviewees and the survey participants) applied five principal HPO methods to pursue six goals, influenced by fourteen contextual factors (see Sections 4.1.1–4.1.3). Participants reported varying success rates in pursuing their goals (see Section 4.2).

4.1 HPO Practices

The following first briefly describes what HPO methods the study participants used. Second, we report the goals pursued by the study participants and which of the four HPO methods practitioners used to attain what goals. Third, we introduce fourteen contextual factors and how they can influence practitioners in their decisions for HPO methods to achieve specific goals.

4.1.1 HPO Methods Used by Practitioners. The study participants applied five HPO methods: manual tuning, grid search, random search, Bayesian optimization, and evolutionary algorithms.



(a) Percentage of survey participants that already used the individual HPO methods.

(b) Number of HPO methods used by participants.

Fig. 1. Overview of HPO methods used by the 49 survey participants.

Each HPO method was discussed by between two and eight interviewees. Most interviewees discussed one or two HPO methods in detail.[2] Most survey participants primarily used manual tuning, followed by grid search, Bayesian optimization, random search, and evolutionary algorithms (see Figure 1(a)). Academic participants tended to rely more on manual tuning and grid search compared to those from industry. Most survey participants stated to have used at least three HPO methods in their past ML projects; about 2% have even used at least five HPO methods. Approximately 15% of the survey participants have used only a single HPO method (see Figure 1(b)).

Even though literature indicates that Bayesian optimization yields better results in a shorter time than evolutionary algorithms, grid search, and random search (e.g., [74, 80]), practitioners tend to use seemingly inferior HPO methods. Practitioners seem to not only aim at finding hyperparameter configurations for optimal ML model performance but also pursue different goals.

4.1.2 Goals of Practitioners Pursued with Different HPO Methods. We identified six goals that the participants pursued in HPO (see Table 3). In the following, we first introduce each goal based on the interview results. Then, we describe the results of the survey study.

Comply with Target Audience. The goal *comply with target audience* refers to aligning practitioners' choices with the expectations of their target audience. Three interviewees from academia stated they had decided on HPO methods to comply with the expectations of their target audiences regarding applied HPO methods and the resulting ML model. For example, two of the three interviewees described Bayesian optimization as uncommon in their research communities and felt obliged to explain it in their scientific publications. They would have needed to explain Bayesian optimization in detail, even though they believed the method itself was not central to their research. Therefore, they decided to use grid search as they assumed this HPO method to be well-known in their research communities.

Decrease Necessary Computations. Extensive searches for optimal hyperparameter configurations in large search spaces typically require substantial computational resources. Necessary computations for HPO can be decreased by using an HPO method that requires fewer compute resources than other methods but is still sufficiently useful.

> "*This whole method was already super, super expensive [...] and if you would perform hyperparameter optimization again, then it becomes even more expensive.*"
>
> –Interviewee #8, ML Innovator in Academia

[2]We asked the interviewees to discuss a prominent recent example of how they used HPO methods, not to discuss all HPO methods they ever used.

Table 3. Principal Goals Practitioners Pursue in HPO

Goal	Description
Comply with Target Audience	The state where the applied HPO method and the resulting ML model fulfill the expectations of addressees
Decrease Necessary Computations	The state where an ML model is trained with an HPO method that requires less compute resources than other methods but still is sufficiently useful for a given purpose
Decrease Practitioner Effort	The state in which a practitioner applies an HPO method for training an ML model that requires less resources compared to other HPO methods (e.g., time for learning a new HPO method or implementing corresponding software tools)
Increase ML Model Performance	The state where a refined version of an ML model outperforms its original version in terms of a specified metric
Increase ML Model Understanding	The state where a practitioner is able to predict changes in an ML model's behavior caused by altering hyperparameter configurations based on an understanding of the ML model's inner workings
Satisfy Requirements	The state where the development and training of an ML model satisfies social and technical demands imposed by stakeholders

Unlike the previous goal (*comply with target audience*), decreasing necessary computations is not self-contained. Instead, it functions as a supplementary objective, generally addressed in conjunction with at least one goal and in response to contextual constraints. For example, practitioners may seek to decrease necessary computations to optimize the use of limited resources, enabling adequate improvements in ML model performance.

Decrease Practitioner Effort. Practitioners choose HPO methods to reduce overhead, for example, in terms of the additional time required to understand the HPO method or to integrate the HPO method into workflows. One industry interviewee reported that the sheer number of advanced HPO methods made it challenging to select the most suitable method for their use case. Experiencing the paradox of choice [70], practitioners felt uncomfortable committing to a particular HPO method and tool and therefore opted for manual tuning instead.

To decrease their efforts in HPO, the interviewees applied grid search and manual tuning. In particular, practitioners stated to have applied manual tuning to avoid efforts related to setting up HPO tools in cluster infrastructures.

> "*HPO is time-consuming sometimes because it requires some extra lines of code to wrap all your models with this HPO method and then set up the scripts to run them on a cluster.*"
>
> –Interviewee #5, ML Innovator in Industry

Like *decrease necessary computations*, the goal *decrease practitioner effort* is not self-contained and commonly used in combination with other goals. For example, study participants used HPO tools to enhance ML model understanding by iteratively refining manually defined sets of hyperparameter values. Those practitioners leveraged HPO tools to automate the reconfiguration of defined hyperparameter sets to accelerate manual tuning.

Increase ML Model Understanding. Increasing ML model understanding refers to reaching the state where a practitioner can predict changes in an ML model's behavior due to tuning hyperparameter values based on an understanding of the ML model's inner workings. To increase their understanding of ML models, the interviewees reported having applied manual tuning. The

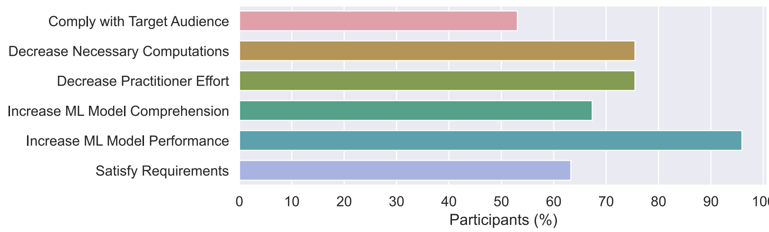


Fig. 2. Relative frequencies of goals pursued by the 49 survey participants.

interviewees claimed that manual tuning can improve their understanding of hyperparameter influences on ML models as they formulate hypotheses about hyperparameter influences on ML models and evaluate them immediately. The interviewees explained that they iteratively improved their ML model understanding by tuning hyperparameter values and testing their hypotheses. This dynamic between ML practitioners and ML models is researched as interactive ML. Interactive ML supports practitioners in actively exploring the model space and testing hypotheses about hyperparameter effects [46]. In addition, visual analytics tools help practitioners by providing graphical representations of model behavior and parameter interactions (e.g., [39, 77]).

Increase ML Model Performance. ML model performance is increased when a refined version of the ML model outperforms its original version in terms of a specified metric. The interviewees chose manual tuning, grid search, random search, and Bayesian optimization HPO methods to achieve this goal for example, to prototype novel ML models.

> "*If the only concern is to find the best model possible and no one asks how I got there, and I do not have a lot of time, I probably would use a random search.*"
>
> –Interviewee #6, ML Engineer in Academia

Satisfy Requirements. The goal to satisfy requirements refers to reaching the state in which the development and training of an ML model fulfill social and technical constraints imposed by stakeholders (e.g., business clients, ethics commissions) and the environment (e.g., available compute resources). Ten interviewees described that their decisions for HPO methods were influenced by the goal of fulfilling such requirements. For example, one interviewee reported preferring manual tuning to meet hard-to-formalize requirements, such as a smooth behavior of the model output.

The survey results indicate that all goals extracted from the interview study are also pursued by the survey participants (see Figure 2). More than 95% of the survey participants pursued the goal *increase ML model performance* and 75% aimed to achieve *decrease necessary computations*; 75% sought to *decrease practitioner effort*; 67% of the practitioners aimed to *increase ML model understanding*. The least pursued goals are *satisfy requirements* (63%) and *comply with target audience* (53%).

Figure 3 shows how often the survey participants used HPO methods to reach the goals identified in the interview study; 67% of the survey participants tried to decrease necessary computations by using Bayesian optimization and 59% using evolutionary algorithms. About 50% of the participants tried to decrease the necessary computations by applying manual tuning. Random search and grid search were least often used, with 43% and 29%, respectively. Decreasing practitioner effort was of interest for less than 50% of the participants, with very similar responses for all HPO methods (yet a notable exception of manual tuning with only 31%). The survey participants primarily used manual tuning to enhance their comprehension of ML models. Interestingly, participants also tried to use Bayesian optimization twice as often as grid search and random search to achieve this goal

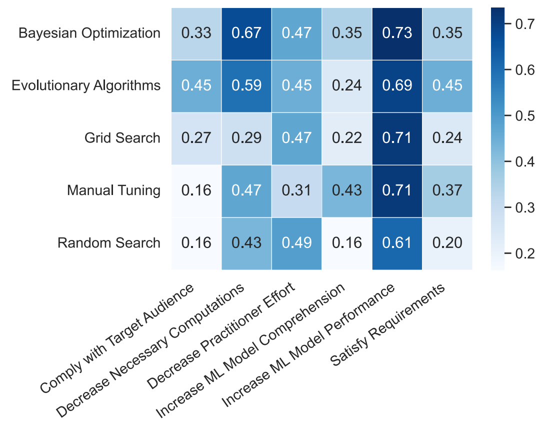


Fig. 3. Frequency of goal and HPO method combinations. Per cell, all presented values are normalized to the number of participants having applied the corresponding HPO method.

despite its black-box nature. Participants from academia have only tried to satisfy requirements or be compliant with the target audience half as often as participants from industry.

Increasing ML model performance is the most common goal in HPO that is pursued across all HPO methods taken into account in this study. Most often, practitioners used Bayesian optimization to achieve this goals.

The participants mostly aimed to *satisfy requirements* using evolutionary algorithms (45%), Bayesian optimization (35%), and manual tuning (37%). They less often used grid search (24%) and random search (20%) to achieve this goal.

Comply with target audience was the least pursued goal for the participants. About 45% of the participants used evolutionary algorithms for this goal. We could not uncover differences between manual tuning and Bayesian optimization, with 30% of the participants using the respective method. Only random search and manual tuning were very seldom used to achieve this goal (about 15%).

In contrast to their academic counterparts, industry participants rarely used Bayesian optimization or evolutionary algorithms to increase model understanding. Participants from academia preferred to use grid search to reduce their effort. Additionally, industry practitioners more frequently relied on manual tuning to meet specific requirements.

It is evident that participants employed different HPO methods to pursue similar goals, making it difficult to establish a clear mapping between individual methods and goals. For example, while one interviewee selected manual tuning to *increase ML model understanding*, another favored Bayesian optimization to pursue the same goal.

> "*Because especially when entering new areas, we would like to understand step by step what is working and what is not.*"
>
> –Interviewee #9, ML Innovator in Academia

> "*I almost always select Bayesian optimization to get an idea in which region I find the [hyper-] parameter [values].*"
>
> –Interviewee #15, ML Innovator in Industry

The ambiguous responses suggest that practitioners' choices of HPO methods cannot be explained solely by their goals. For example, practitioners used different HPO methods to reach identical goals. This underscores the complexity of understanding why practitioners choose specific

Table 4. Principal Contextual Factors Related to Own Knowledge That Can Influence Practitioner Decisions for HPO Methods

Theme	Contextual Factor	Description
Own Knowledge	HPO Method Comprehension	The self-perceived level of knowledge a practitioner has about the inner workings of an HPO method
	ML Model Comprehension	The self-perceived degree of understanding of the inner workings of an ML model with which a practitioner explains changes in the behavior of the ML model caused by altering hyperparameter values
	Personal Experience	The available internal knowledge of a practitioner that has been generated by past activities (e.g., personal best practices to solve a specific problem type)
Social Environment	Acceptance of Advanced HPO Methods	The extent to which advanced HPO methods (e.g., Bayesian optimization) are valued by a target group
	Literature	The knowledge acquired on the basis of published text documents (e.g., articles, blog entries, and papers)
	Shared Opinions	The knowledge acquired on the basis of advice from peers (e.g., colleagues)
	Tension for Resources	The degree to which constrained compute resources cause conflicts between practitioners regarding the allocation of those resources
Technical Environment	Available Compute Resources	The amount of compute resources available for HPO
	Cost of Objective Function	The amount of compute resources required to evaluate a single point within a hyperparameter value space
	HPO Method Traceability	The extent to which a sequence of sample points can be backtraced or predicted
	HPO Setup Readiness	The degree to which an HPO tool and associated test environments are ready to use (e.g., preinstalled HPO tools on a cluster)
	Parallelization Possibilities	The degree to which multiple independent ML models can be simultaneously evaluated
	Search Space Size	The number of possible hyperparameter configurations
	Usability of HPO Tools	The perceived ease with which practitioners achieve their goals by using an HPO method and corresponding tools

HPO methods. To better understand practitioners' reasons for selecting HPO methods, contextual factors that influence practitioners in their decisions for HPO methods to reach goals seem important. By also illuminating contextual factors in combination with the use of HPO methods and goals to be achieved, the motives of practitioners to use HPO methods should be better understood.

4.1.3 Contextual Factors That Influence HPO Method Selections. We identified 14 contextual factors that can influence practitioner decisions for using HPO methods to achieve specific goals. The contextual factors can be grouped into three themes (see Table 4): *own knowledge*, *social environment*, and *technical environment*.

Own Knowledge. Practitioner decisions for HPO methods are influenced by practitioners' internal knowledge of HPO and ML models. We identified three contextual factors related to own knowledge: *HPO method comprehension*, *ML model understanding*, and *personal experiences.*

HPO method comprehension refers to the degree to which practitioners understand how HPO methods work and how to apply them. Practitioners tend to neglect HPO methods they do not sufficiently understand. For example, two interviewees stated they had disregarded Bayesian optimization because they felt they did not sufficiently understand its inner workings. One interviewee from the industry perceived grid search to be faster to implement and easier to use compared to Bayesian optimization because using the latter would have required the interviewee to learn an HPO method they were not experienced in. Another interviewee perceived random search as uncontrolled, which caused them to decide against it. Two interviewees decided to use grid search because they perceived it as easy to understand and implement.

ML model comprehension refers to a practitioner's ability to explain changes in an ML model's behavior caused by altering hyperparameter values based on an understanding of the inner workings of the ML model. The perceived degree of ML model understanding plays an important role. Interviewees who perceived their ML model understanding as high stated to have chosen manual tuning. Due to their thorough ML model understanding, those interviewees claimed that they could find appropriate hyperparameter configurations without extensive HPO. The interviewees perceived programmatic HPO methods as not taking advantage of known effects of hyperparameters on ML model development and performance. Examples of such known effects include a high learning rate, which can accelerate learning but may lead to instability; a low learning rate, which slows convergence; strong regularization, which can cause underfitting; and weak regularization, which can lead to overfitting [88].

> "*Effects of hyperparameters are often deducible, but optimizers [here: HPO methods] usually do not support functionalities for this.*"
>
> –Interviewee #1, ML Innovator in Academia

Interviewees who deemed their ML model understanding as low tended to use random search or Bayesian optimization. Low ML model understanding made it difficult for interviewees to predict the challenges they would encounter in HPO. To better react to unforeseen challenges, interviewees stated to choose manual tuning instead. For example, manual tuning can facilitate spotting and correcting mistakes when errors occur during the development of novel ML models because feedback loops are faster compared to those of programmatic HPO methods:

> "*Because we altered the standard architecture as a whole, we were not really sure what problems we would face. So that was one of the reasons to stick with manual tuning.*"
>
> –Interviewee #3, ML Engineer in Academia

Personal experiences refers to the available internal knowledge that a practitioner generated through past activities (e.g., personal best practices for solving a specific type of problem). The interviewees stated that they tend to use HPO methods with which they had positive experiences:

> "*I have also had good experiences with it [here: Bayesian optimization] in a previous paper.*"
>
> –Interviewee #2, ML Innovator in Academia

Social Environment. Choices for HPO methods are influenced by the social environment of practitioners, especially by four contextual factors: *acceptance of advanced HPO methods*, *literature*, *shared opinions*, and *tension for resources.*

Acceptance of advanced HPO methods refers to the extent to which advanced HPO methods, such as Bayesian optimization, are valued by a target group. Low acceptance of advanced HPO methods in a community targeted by a practitioner can make them avoid extensive HPO entirely and choose manual tuning. For example, an academic stated that they perceived the use of advanced HPO methods and extensive HPO as not being valued by their community. According to the interviewee, their community encourages the use of pre-trained ML models in combination with manual fine-tuning to avoid extensive HPO. Although the interviewee perceived Bayesian optimization as more suitable for increasing ML model performance, they felt discouraged by the attitude of their community and applied manual tuning instead.

Shared opinions covers external knowledge acquired on the basis of advice from peers (e.g., colleagues). The interviewees explained to have chosen HPO methods that are considered as commonly used in their labs or by their peers. In various communities, different HPO methods are applied so frequently that their use becomes habitual. For example, manual tuning was commonly used in one research group, while Bayesian optimization was considered the primarily applied HPO method in another one. The interviewees associated with those communities applied the, respectively, manifested HPO methods. This indicates that immediate social environment has a noticeable influence on practitioners' HPO method choices.

Literature refers to external knowledge acquired on the basis of published text documents (e.g., articles, blog entries, papers). Practitioners, from academia and industry alike, are guided in their choices of HPO methods by recommendations from literature on ML models similar to their own. All practitioners, who primarily based their decisions on literature, chose Bayesian optimization that attests high sample efficiency (e.g., [80]).

Tension for shared resources refers to the degree to which constrained compute resources cause conflicts between practitioners. Availability of only shared resources can cause tensions among colleagues, for example, when practitioners must compete for compute resources to perform HPO. Such tensions led one academic scientist to choose manual tuning to avoid arguing with colleagues over compute resources.

Technical Environment. Contextual factors associated with the technical environment refer to technical constraints (e.g., caused by insufficient compute resources) that influence practitioners' selections of HPO methods. The interviewees stated seven contextual factors associated with the technical environment: *available compute resources*, *cost of objective function*, *HPO method traceability*, *HPO setup readiness*, *parallelization possibilities*, *search space size*, and *usability of HPO tools*.

Available compute resources refers to the amount of compute resources available for HPO. Practitioners choose manual tuning when faced with constrained available compute resources. They perceive that in combination with a high degree of ML model understanding, they can outperform programmatic HPO methods. If the available compute resources are too scarce, exploration of large search spaces is hard. Practitioners need to reduce search spaces, decrease the number of necessary function evaluations, or decrease computational cost per function evaluation (e.g., by low-fidelity approximations) to be able to perform HPO. To decrease the number of necessary function evaluations, three scientists stated to have used manual tuning because they were able to predict the influence of hyperparameter values on ML model performance. Given constrained available compute resources and a sufficient degree of ML model understanding, scientists in this study perceived manual tuning as superior to Bayesian optimization and random search.

Two interviewees chose HPO methods depending on the *cost of the objective function* they sought to optimize (i.e., training of an ML model). The cost of the objective function refers to the amount of compute resources required to evaluate a single point within the hyperparameter value space. Similar to constrained compute resources, the interviewees chose manual tuning when faced with too expensive objective functions. When the interviewees perceived their degree of ML model understanding as high, they deemed manual tuning more efficient.

HPO method traceability refers to the extent to which a sequence of sample points can be backtracked or predicted by the practitioner, which requires that the selection of samples by the HPO method is comprehensible for and reproducible by practitioners.

HPO setup readiness refers to the degree to which HPO tools and test environments are ready to use (e.g., preinstalled HPO tools on the cluster). Some study participants stated that they were unwilling to set up new HPO tools but preferred to use already set up tooling, regardless of the quality produced by the corresponding HPO method.

Parallelization possibilities of HPO methods refers to the degree to which independent ML models can be simultaneously evaluated. Parallelization possibilities can be constrained by, for example, too few software licenses. Two interviewees chose Bayesian optimization if parallelization of HPO was not possible due to a misconception of the sequential process in Bayesian optimization. Another interviewee stated that they chose Bayesian optimization if their objective function is expensive and HPO parallelization is not possible.

Search space size refers to the number of possible hyperparameter combinations. The interviewees stated that the number of hyperparameters included in the HPO impacted their decisions for HPO methods. For example, they stated to opt for random search over grid search and Bayesian optimization if the number of hyperparameters is large.

Usability of HPO tools refers to the perceived ease with which practitioners can achieve their goal by using an HPO method and corresponding implementations. Interviewees from industry remarked that many advanced HPO tools may be useful for academic purposes but lack the necessary level of maturity to be viable in practice. Within the scope of usability, practitioners demanded more automation of cumbersome tasks in HPO such as infrastructure orchestration:

> "*What beats everything for me is that I have a dashboard that's somewhere in the cloud that orchestrates my various agents, where I can sort of say online, 'Start another agent on this machine,' or that on the machine I just have to say, 'Start another agent on this sweep here,' and I don't have to worry about the agents talking to each other or having a shared database running on some cluster. This functionality, it overrides everything. If I had some mega highly optimized Bayesian optimization tool that didn't have that functionality, I wouldn't use it.*"
>
> –Interviewee #14, ML Innovator in Industry

The results of the survey study show that each contextual factor influenced at least 70% of the participants in selecting HPO methods (see Figure 4). More than 85% of the survey participants considered the decision factors *personal experience*, *search space size*, and *available compute resources* when selecting HPO methods. The contextual factors least considered are *acceptance of advanced methods*, *tension for resources*, and *parallelization possibilities*. They were only relevant for less than 75% of survey participants in their past ML projects. The remaining contextual factors from all three themes, including *shared opinions*, *model comprehension*, and *HPO method traceability*, have been considered by 75–85% of the survey participants.

The identified contextual factors are of different self-perceived relevance for the selection of HPO methods (see Figure 5). The self-perceived relevance of contextual factors is interdependent with the consideration of contextual factors. *Usability of HPO tools* and *search space size* are the most relevant contextual factors, closely followed by *available compute resources* and *HPO setup readiness*. Other relevant contextual factors are *personal experience*, *cost of the objective function*, and the *HPO method* and *ML model comprehension*. All contextual factors associated with the social environment are less relevant to survey participants, with *tension for resources* being considered the least.

Figure 6 illustrates the self-perceived relevance of contextual factors for each of the HPO methods in the scope of this study. The survey participants considered *available compute resources*,

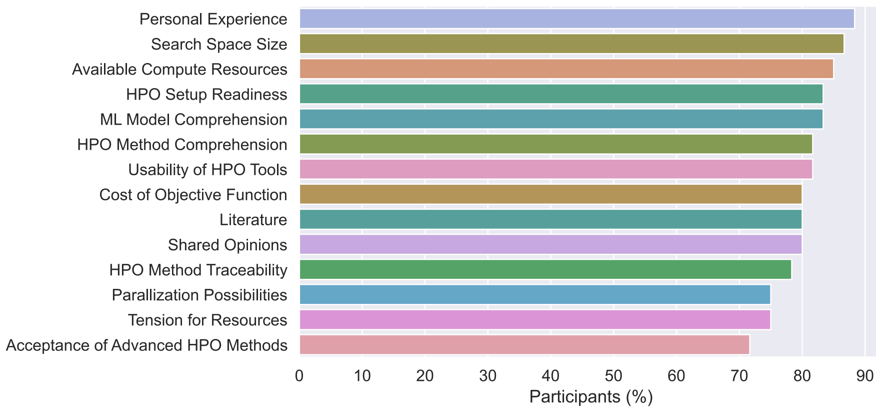


Fig. 4. Percentage of survey participants that incorporated the individual contextual factors.

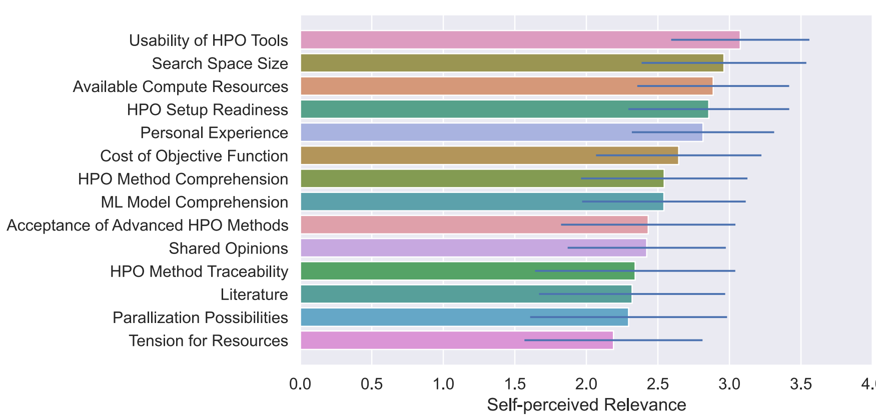
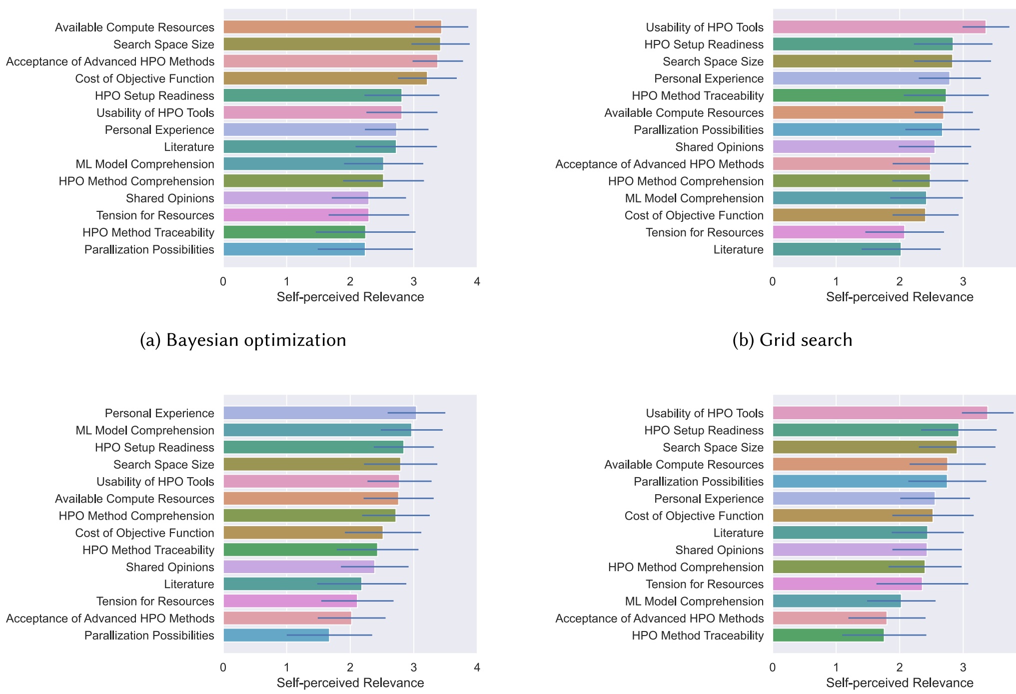


Fig. 5. Overview of the average self-perceived relevance of contextual factors. Results are reported on a scale from 0 (very low) to 5 (very high). Blue lines indicate error bars of one standard deviation (SD).

search space size, *acceptance of advanced methods*, and *cost of the objective function* mainly when selecting Bayesian optimization. This aligns with commonly cited reasons in the literature for using Bayesian optimization [27, 43, 76]. The selection of grid search was mostly influenced by *usability of HPO tools*, *HPO setup readiness*, and *search space size* with similar results for random search. A potential explanation could be the availability of these HPO methods in established and publicly available ML libraries like *scikit-learn*. Moreover, survey participants considered their *personal experience*, *ML model comprehension*, and *HPO setup readiness* most relevant when selecting manual tuning, making *own knowledge* more important than *technical environment*. This indicates that the relative importance of contextual factors in a specific instance leads to different selections of HPO methods.

4.2 Perceived Success of Using HPO Methods to Achieve Specific Goals

Practitioners seem to have diverse and individual motivations for using specific HPO methods. However, these choices do not always lead to the desired outcomes. To distinguish between



(a) Bayesian optimization (b) Grid search

(c) Manual tuning (d) Random search

Fig. 6. Overview of the average self-perceived relevance of contextual factors per HPO method. Results are reported on a scale from 0 (very low) to 5 (very high). Blue lines indicate error bars of one SD.

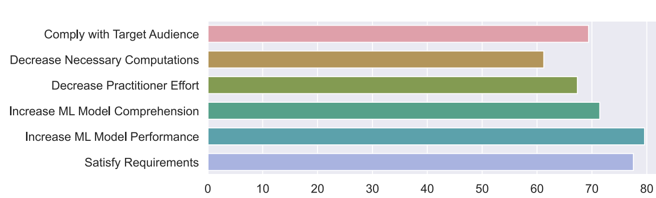


Fig. 7. Self-reported success rate per goal by the 49 survey participants.

successful and unsuccessful experiences of practitioners in using HPO to reach their goals, we asked the study participants to what extent they perceive they have reached which goals using what HPO methods.

Figure 7 shows the success rate per goal perceived by the survey participants. Roughly 75% of the participants responded to have successfully *increased ML model performance*, *complied with target audience*, *increased ML model understanding*, or *satisfied requirements*. Industry participants were successful in trying to *satisfy requirements* or *comply with target audience*; 67% of the participants stated that they were able to achieve the goal *decrease practitioners effort*. Only 62% of participants considered themselves successful at *reducing computational requirements*.

The self-perceived success rates strongly vary between combinations of goals and HPO methods (see Figure 8). Survey participants reported lower success rates in *reducing computational*

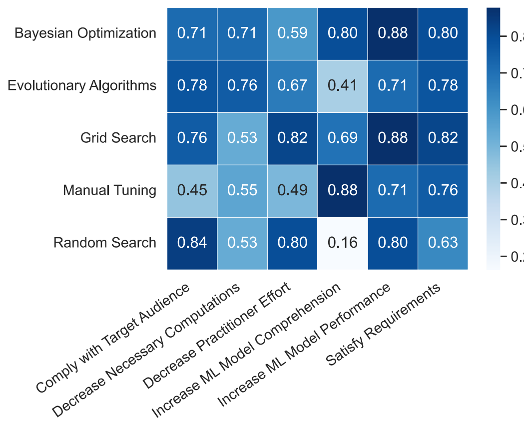


Fig. 8. Self-reported success rates per goal-method combination by the 49 survey participants.

requirements when using manual tuning, grid search, or random search. They perceived themselves as rather successful in reaching this goal when using Bayesian optimization or evolutionary algorithms. *Decreasing practitioner effort* was best achieved using grid search or random search according to the survey participants. Bayesian optimization and evolutionary algorithms were perceived as less effective in decreasing effort. A potential explanation could be that those HPO methods often require more effort to be set up compared to others [50]. Moreover, participants also perceived manual tuning as ineffective in decreasing their efforts. Participants perceived manual tuning as very helpful to *increase ML model understanding*. Even though Bayesian optimization is considered a black-box optimization technique [24], it was also perceived as suitable to increase ML model understanding. Random search, grid search, and evolutionary algorithms were perceived as unsuitable for increasing ML model understanding. Most participants did not perceive noteworthy differences between the effectiveness of HPO methods to successfully *increase ML model performance* and to *satisfy requirements*. Only evolutionary algorithms were perceived as significantly more successful in trying to satisfy requirements. Survey participants successfully used grid search, random search, evolutionary algorithms, and Bayesian optimization to *meet the expectations of their target audience*. Manual tuning was applied with lower success rates for this goal.

5 Discussion

The results presented in the previous section offer novel insights into how practitioners use HPO methods. In this section, we highlight and discuss the principal findings from those insights. We describe how the results of this study contribute to more user-centric development of HPO methods and tools, explicate the limitations of this study, and outline future research directions.

5.1 Principal Findings

Besides improving ML model performance, practitioners are most interested in reducing the number of necessary computations and their personal effort required for HPO, which reflects the basic motivation for development of HPO tools [84]. To decrease necessary computations, the study participants predominantly used Bayesian optimization. The frequent use of Bayesian optimization to decrease necessary computations suggests that practitioner perceptions of the benefits of Bayesian optimization are coherent with its benefits empirically shown in prior research [80].

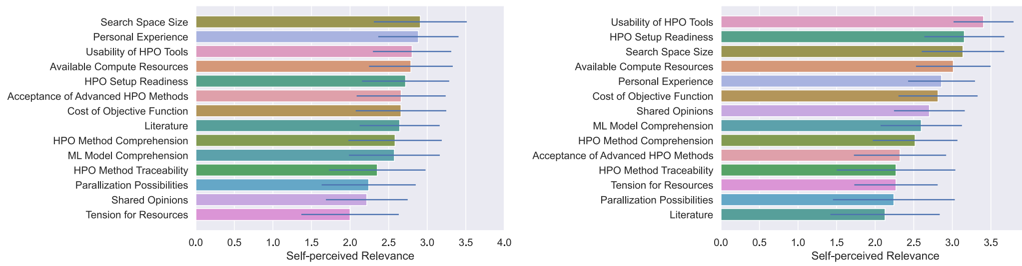
Notwithstanding the high sample efficiency of Bayesian optimization, some practitioners prefer to use manual tuning to decrease the number of necessary computations. In particular, if practitioners assume that their ML model understanding is high, they expect to outperform Bayesian optimization. Yet, it is difficult to compare manual tuning to programmatic HPO methods due to its reliance on a mixture of explicit and implicit knowledge that often cannot be fully extracted from observations of practitioner actions. One way to leverage ML model understanding could be to integrate practitioner priors on the location of well-performing hyperparameter configurations into Bayesian Optimization effectively warm-starting optimizations [44, 58, 76].

In addition to well-established goals pursued in HPO, this study presents a multitude of goals pursued by practitioners that are less emphasized in research on AutoML. Practitioners have strong motives for HPO beyond improving ML model performance. For example, the results of this study show that various practitioners are interested in better understanding their subject of work, including HPO methods and ML models, prior to using them. To better understand ML models, most practitioners chose HPO methods they understood over methods they would have to study first. This led practitioners to opt for manual tuning instead of programmatic HPO methods. Many participants perceived programmatic HPO tools as unsuited to increase ML model understanding. Although numerous software packages for advanced HPO methods are available for out-of-the-box use without requiring an understanding of their internal workings (e.g., Optuna [1] and SMAC3 [55]), practitioners appear reluctant to adopt HPO methods they do not fully understand. Practitioners tend to rely on their own knowledge rather than giving up control to insufficiently understood ML models and HPO tools.

Various software tools have been designed to help practitioners increase ML model understanding. Such tools, predominantly HPO tools in AutoML, mainly focus on supporting measurements of influences of hyperparameter configurations on ML model performance (e.g., [5, 41, 62, 71]). With a focus on HPO methods, especially tools for visual analytics are envisioned to support a better understanding of internal behaviors of HPO methods by visualizations (e.g., [6, 34, 69, 92]). Despite the existence of such tools, practitioners tend to prefer to use manual tuning, which may have different reasons. The first reason may be that practitioners are unaware of HPO tools that can help increase ML model understanding. Another reason may be that HPO tools do not fulfill the information needs of practitioners to increase ML model understanding because HPO tools mainly focus on performance of ML models, which, as shown in this study, is only one practitioner motive in HPO. A third reason may be that the functioning of HPO tools themselves is hard to comprehend for practitioners (e.g., because such HPO tools implement unfamiliar and complex HPO methods), which leads practitioners to prefer HPO methods they are familiar with.

The results presented in this study provide an aggregated view of practitioner motives in using HPO methods. Although our analysis did not reveal distinct personas with significantly different usage patterns, our results highlight nuanced variations in how practitioners engage with HPO methods. Specifically, we observed that the importance of goals and the use of HPO methods remained relatively consistent among practitioners, regardless of their experience, age, or education level. However, our findings indicate that contextual factors play a more prominent role in shaping the approaches of practitioners from academia versus those from industry (see Figure 9). Contrary to practitioners from industry, for example, academics tend to avoid using HPO methods that are difficult to integrate into workflows or require prior training. This especially applies when using complex HPO methods, according to the study participants.

To make the achievements of techno-centric HPO research more actionable in practice and to support development of novel and more user-centered HPO tools, it is essential to integrate practitioners' motivations for HPO into the tool development process. In response to calls for more human-centered AutoML [51, 56, 78, 87], we propose improvements to programmatic HPO methods



(a) Academia (27 participants). (b) Industry (22 participants).

Fig. 9. Relevance of the identified contextual factors in academia and industry.

based on our findings to complement technological advances in AutoML, which primarily focus on conventional performance metrics, by incorporating social aspects.

Increase ML Model Understanding. The inspection of frequent HPO method and goal combinations revealed that increasing ML model comprehension is the only goal with a strongly higher association with manual tuning than programmatic HPO methods (see Figure 3). A potential explanation could be that established HPO tools usually focus on finding the best-performing hyperparameters and do not provide details of other interesting hyperparameter values tested [87, 92]. To overcome this limitation, HPO tools should generate reports on the behavior of different ML models, for example, the importance of individual hyperparameters. For the generation of such reports, many methods are already available, such as functional ANOVA [41], ablation [5], importance of local parameters [6], partial dependence plots [62], and symbolic regressions [71]. Alternatively, HPO tools could provide additional insights about ML model behavior. Especially for more complex search spaces used for building complete ML pipelines, information about the transformation of input data can additionally help increase ML model understanding [92]. Including such reports in HPO tools can facilitate leveraging the benefits of advanced HPO methods (e.g., high sample efficiency) while still helping practitioners to *increase ML model understanding*.

Explain HPO Method Internals. The results of this study show that a key barrier to the adoption of advanced HPO methods is the perceived lack of transparency (in terms of HPO method traceability) and interpretability (in terms of HPO method comprehension), which affects the use of programmatic HPO tools (see Figure 6(a)). HPO tools should provide more support explaining their internal behavior to make them better comprehensible to practitioners. An easy approach would be simple visualizations of the hyperparameter values that were evaluated using parallel coordinates plots [34]. More sophisticated approaches could present information about the internals of their optimizers, for example, the surrogate model in Bayesian optimization [6]. Such measures could help educate practitioners about HPO methods and increase practitioners' confidence in the functioning and merits of programmatic HPO tools.

Integrate ML Model Understanding of Practitioners in HPO. Figure 6(c) shows that ML model comprehension is an important contextual factor for preferring manual tuning instead of a programmatic HPO method. Similarly, multiple interviewees mentioned that they prefer manual tuning when they are confident in predicting the impact of hyperparameters on the ML model behavior. To harness this knowledge, HPO tools should enable practitioners to incorporate their relevant knowledge of the behaviors of ML models into HPO tools prior to HPO on a case-by-case basis, tailored to goals like *improve ML model performance*. For example, practitioners could specify their perceived hyperparameter importance and influence between hyperparameters. Furthermore, ML models comprehension of practitioner could be directly incorporated into the search strategy of HPO methods, for example, to reduce search spaces [87] and better use available computing resources (the most important contextual factor for Bayesian optimization in Figure 6(a)). Promising work in this direction includes methods for integrating prior knowledge into Bayesian optimization. This can be achieved by directly specifying priors about the location of the optimum [44, 52, 67, 76], or structural priors, for example, in the form of log-transformations of hyperparameters [42], monotonicity constraints [53], or hyperparameter warping [75].

5.2 Contributions

The primary goal of this research is to inform the AutoML community and the **human-computer interaction (HCI)** community about practitioners' motives for using HPO methods. By bridging human-centered perspectives from the HCI community with technical advancements from the ML community, this work makes three key contributions to the development of more effective AutoML tools.

First, we offer a conceptual foundation that outlines why practitioners use HPO methods. This foundation comprises six core goals (e.g., improving ML model performance, aligning with a target audience) and 14 contextual factors (e.g., compute resource availability, traceability of HPO methods) that influence choices of practitioners. From the perspective of HCI, the conceptual foundation enables more user-centered HPO research by supporting better understanding of practitioner motives. It supports research on human-in-the-loop ML by helping define information needs (e.g., related to improving transparency in HPO tools) and designing better practitioner engagement strategies for different HPO methods. From a technical perspective, researchers focusing on HPO tool development can leverage these insights to build HPO methods and HPO tools that extend beyond performance optimization. Future HPO tools could be designed to be more context-sensitive, improving their adaptability and utility. Additionally, the identified goals can inform design of benchmarks for evaluating HPO tools in terms of compute resource efficiency and automation levels, rather than focusing ML model performance.

Second, we present a mapping between goals, HPO methods, and contextual factors, offering insights into why practitioners choose specific HPO methods. This mapping helps align HPO tool development with real-world practitioner needs. From the perspective of HCI, this work supports better understanding of the decision-making process for HPO methods, enabling the design of more intuitive, goal-driven HPO tools. This is useful to develop tailored automation features that cater to specific goals and contextual factors. From a technical perspective, the mapping presents key input parameters (e.g., priority of goals and contextual factors) that can be leveraged in development of HPO tools. For instance, specialized HPO tools could be designed to optimize specific contextual factors rather than using one-size-fits-all approaches.

Third, we present an overview of how practitioners perceive the success of different HPO methods in different contexts. The overview highlights areas where existing tools meet expectations and where improvements are needed. From the perspective of HCI, understanding practitioners' perceived success helps inform the design of better decision-support systems for HPO. It also enables development of novel interaction concepts that improve how practitioners select and use HPO tools. From a technical perspective, by analyzing self-reported success across different goals and contextual factors, we support a better understanding of the strengths and weaknesses of HPO methods. This can guide the development of new HPO methods tailored to specific workflows, practitioner goals, and technical constraints.

5.3 Limitations and Future Work

We performed semi-structured interviews in a qualitative and explorative research approach. Interview results strongly rely on interviewees' knowledge, perceptions, and capabilities to verbalize responses to questions—common sources for biases. We aimed to decrease biases in data gathering and data analysis by reaching out to a variety of practitioners with different levels of experience and work fields. Moreover, we aimed to decrease bias in the analysis of gathered data as multiple analysts independently coded the interview transcripts and discussed their results to agree on a shared understanding. However, despite these efforts, we cannot fully guarantee exhaustiveness and elimination of biases. Additional goals and contextual factors may be relevant to practitioners. Future research could extend this work to uncover additional goals and contextual factors that were not mentioned by the participants of this study (e.g., size of training datasets).

The results presented in this work hint at possible conflicts between goals and contextual factors. Practitioners must resolve such conflicts to succeed in HPO, which entails prioritizing goals and assessing the relevance of contextual factors. A possible conflict can arise when practitioners aim to "decrease necessary computations" and "increase ML model performance." Practitioners may attempt to find a Pareto-optimal achievement of both goals based on a clear prioritization. This work offers a foundation of goals and contextual factors that can lead to tradeoffs and call for prioritization of goals. Future work should investigate relationships between goals and contextual factors to uncover such tradeoffs and investigate how practitioners resolve them (e.g., in terms of prioritization).

Future investigations into human decision-making in HPO represent a promising research direction that could improve AutoML by incorporating human knowledge [40, 78, 86]. The practitioners we interviewed described actions they took during HPO that were independent of the specific HPO methods used—such as selecting a promising subset of hyperparameters to tune and defining appropriate search ranges.

Most interviewees followed similar procedures when choosing HPO methods, selecting hyperparameters, and configuring tuning strategies. Given that they reported successfully achieving their goals, this consistency suggests the existence of best practices for HPO-related tasks. Since many participants reported making these decisions largely unconsciously yet still achieved satisfactory outcomes, identifying the cognitive heuristics used in practitioners' decision-making [28, 29] holds great promise for advancing the automation of HPO within AutoML.

Understanding the functional heuristics practitioners use in HPO could lead to deeper insights into their decision-making processes. This, in turn, could inform the development of AutoML systems that are capable of automating complex tasks by emulating human reasoning in a resource-efficient way. Furthermore, such heuristics could contribute to the design of more effective and efficient HPO tools.

6 Conclusion

While programmatic HPO methods, such as Bayesian optimization and evolutionary algorithms, achieve high efficiency of the HPO process, practitioners sometimes opt for efficiency-wise inferior HPO methods, such as grid search and manual tuning. To understand practitioner motives for using HPO methods, we performed a two-step research approach consisting of semi-structured interviews and a survey based on an online questionnaire. We identified six principal goals pursued by practitioners in HPO, such as *decrease practitioner effort*, *decrease necessary computations*, and *increase ML model understanding*. Moreover, we extracted fourteen contextual factors that influence practitioners' decisions for using HPO methods, such as *available compute resources*, *HPO method traceability*, and *parallelization possibilities*.

By bridging the gap between technological advancements and practitioner motives to use HPO methods, this work contributes to enhancing HPO practices and tools in the context of AutoML. In particular, the results of this study can guide development of more user-centered HPO methods and HPO tools that cater to practitioner motives.

This work calls for more user-centered research on HPO, particularly on exploring purposeful ways to involve practitioners in programmatic HPO methods, decision-support systems for HPO, and enhancing transparency and comprehensibility of programmatic HPO methods. We will build on the findings presented in this work and seek to identify functional human heuristics [28] applied in HPO. After identifying human heuristics (e.g., [33]), we aim to implement them in algorithms for AutoML and evaluate those algorithms in comparison to the performance of human decision-making and less human-centered HPO tools.

Acknowledgments

We thank all study participants for their time and valuable contributions, which formed the foundation for this work. In particular, we thank Mikael Beyene and Benjamin Sturm for their fruitful input.

References

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis (Eds.), ACM, 2623–2631.

[2] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems*, Vol. 24, 2546–2554.

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

[4] James Bergstra, Brent Komer, Chris Eliasmith, Dan Yamins, and David D. Cox. 2015. Hyperopt: A Python library for model selection and hyperparameter optimization. *Computational Science & Discovery* 8, 1 (2015), 014008–014024.

[5] Andre Biedenkapp, Marius Lindauer, Katharina Eggensperger, Frank Hutter, Chris Fawcett, and Holger Hoos. 2017. Efficient parameter importance analysis via ablation with surrogates. *Proceedings of the AAAI Conference on Artificial Intelligence* 31, 1 (Feb. 2017), 773–779.

[6] André Biedenkapp, Joshua Marben, Marius Lindauer, and Frank Hutter. 2018. Cave: Configuration assessment, visualization and evaluation. In *Proceedings of the International Conference on Learning and Intelligent Optimization*, 115–130.

[7] Martin Binder, Julia Moosbauer, Janek Thomas, and Bernd Bischl. 2020. Multi-objective hyperparameter tuning and feature selection using filter ensembles. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, 471–479.

[8] Bernd Bischl, Martin Binder, Michel Lang, Tobias Pielok, Jakob Richter, Stefan Coors, Janek Thomas, Theresa Ullmann, Marc Becker, Anne-Laure Boulesteix, et al. 2023. Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *WIREs Data Mining and Knowledge Discovery* 13, 2 (2023), e1484.

[9] Xavier Bouthillier and Gaël Varoquaux. 2020. *Survey of Machine-Learning Experimental Methods at NeurIPS2019 and ICLR2020*. Research Report, Inria Saclay Ile de France.

[10] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative Research in Psychology* 3, 2 (2006), 77–101.

[11] Virginia Braun and Victoria Clarke. 2012. Thematic analysis. In *APA Handbook of Research Methods in Psychology, Vol 2: Research Designs: Quantitative, Qualitative, Neuropsychological, and Biological*. American Psychological Association, 57–71.

[12] Eric Brochu, Vlad M. Cora, and Nando De Freitas. 2010. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. arXiv:1012.2599. Retrieved from https://arxiv.org/abs/1012.2599

[13] Yutian Chen, Aja Huang, Ziyu Wang, Ioannis Antonoglou, Julian Schrittwieser, David Silver, and Nando de Freitas. 2018. Bayesian optimization in AlphaGo. arXiv:1812.06855. Retrieved from https://arxiv.org/abs/1812.06855.

[14] Anamaria Crisan and Brittany Fiore-Gartland. 2021. Fits and starts: Enterprise use of AutoML and the role of humans in the loop. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. Article 601, 15 pages.

[15] Tobias Domhan, Jost Tobias Springenberg, and Frank Hutter. 2015. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In *Proceedings of the 24th International Conference on Artificial Intelligence*, 3460–3468.

[16] Samuel Dooley, Rhea Sanjay Sukthanker, John P. Dickerson, Colin White, Frank Hutter, and Micah Goldblum. 2022. On the importance of architectures and hyperparameters for fairness in face recognition. In *Proceedings of the Workshop on Trustworthy and Socially Responsible Machine Learning (NeurIPS '22)*.

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

[18] Salijona Dyrmishi, Radwa Elshawi, and Sherif Sakr. 2019. A decision support framework for AutoML systems: A meta-learning approach. In *Proceedings of the 2019 International Conference on Data Mining Workshops*, 97–106.

[19] Katharina Eggensperger, Matthias Feurer, Frank Hutter, James Bergstra, Jasper Snoek, Holger Hoos, and Kevin Leyton-Brown. 2013. Towards an empirical foundation for assessing Bayesian optimization of hyperparameters. In *NIPS Workshop on Bayesian Optimization in Theory and Practice*, Vol. 10.

[20] David Eriksson, Michael Pearce, Jacob R. Gardner, Ryan Turner, and Matthias Poloczek. 2019. Scalable global optimization via local Bayesian optimization. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019 (NeurIPS)*. Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.), 5497–5508.

[21] Stefan Falkner, Aaron Klein, and Frank Hutter. 2018. BOHB: Robust and efficient hyperparameter optimization at scale. In *Proceedings of the 35th International Conference on Machine Learning*, 1437–1446.

[22] Matthias Feurer and Frank Hutter. 2019. *Hyperparameter Optimization*. Springer, Cham, Switzerland, 3–33.

[23] Matthias Feurer, Matthias Klein, and Frank Hutter. 2016. Winning the AutoML challenge with auto-sklearn. Retrieved from https://www.kdnuggets.com/2016/08/winning-automl-challenge-auto-sklearn.html

[24] Peter I. Frazier. 2018. A tutorial on Bayesian optimization. arXiv:1807.02811. Retreived from https://arxiv.org/abs/1807.02811

[25] Patricia Fusch and Lawrence Ness. 2015. Are we there yet? Data saturation in qualitative research. *Qualitative Report* 20 (2015), 1408–1416.

[26] Steven Gardner, Oleg Golovidov, Joshua Griffin, Patrick Koch, Wayne Thompson, Brett Wujek, and Yan Xu. 2019. Constrained Multi-Objective optimization for automated machine learning. In *Proceedings of the 2019 IEEE International Conference on Data Science and Advanced Analytics*, 364–373.

[27] Roman Garnett. 2023. Introduction. In *Bayesian Optimization*. Cambridge University Press, Cambridge, UK, 1–14.

[28] Gerd Gigerenzer and Henry Brighton. 2009. Homo heuristicus: Why biased minds make better inferences. *Topics in Cognitive Science* 1, 1 (2009), 107–143.

[29] G. Gigerenzer and R. Selten. 2002. *Bounded Rationality: The Adaptive Toolbox*. The MIT Press, Cambridge, MA.

[30] Pieter Gijsbers, Marcos L. P. Bueno, Stefan Coors, Erin LeDell, Sébastien Poirier, Janek Thomas, Bernd Bischl, and Joaquin Vanschoren. 2023. AMLB: An AutoML Benchmark. arXiv:2207.12560. Retreived from https://arxiv.org/abs/2207.12560

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

[32] Varun Godbole, George E. Dahl, Justin Gilmer, Christopher J. Shallue, and Zachary Nado. 2023. Deep Learning Tuning Playbook. Version 1.0. Retrieved from http://github.com/google-research/tuning_playbook

[33] V. Godbole, G. E. Dahl, J. Gilmer, C. J. Shallue, and Z. Nado. 2023. Deep Learning Tuning Playbook. Version 1. Retrieved from http://github.com/google-research/tuning_playbook

[34] Daniel Golovin, Benjamin Solnik, Subhodeep Moitra, Greg Kochanski, John Karro, and D. Sculley. 2017. Google vizier: A service for black-box optimization. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1487–1495.

[35] Raymond L. Gorden. 1975. *Interviewing: Strategy, Techniques, and Tactics* (Rev. ed.). Dorsey Press, Homewood, Ill.

[36] Greg Guest, Arwen Bunce, and Laura Johnson. 2006. How many interviews are enough?: An experiment with data saturation and variability. *Field Methods* 18, 1 (Feb. 2006), 59–82. DOI: https://doi.org/10.1177/1525822X05279903

[37] Yihui He, Ji Lin, Zhijian Liu, Hanrui Wang, Li-Jia Li, and Song Han. 2018. AMC: AutoML for model compression and acceleration on mobile devices. In *Proceedings of the European Conference on Computer Vision*, 784–800.

[38] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. 2018. Deep reinforcement learning that matters. In *AAAI '18/IAAI '18/EAAI '18*, Vol. 32, Article 392, 8 pages.

[39] Frank Heyen, Tanja Munz, Michael Neumann, Daniel Ortega, Ngoc Thang Vu, Daniel Weiskopf, and Michael Sedlmair. 2020. ClaVis: An interactive visual comparison system for classifiers. In *Proceedings of the Conference on Advanced Visual Interfaces*, 1–9.

[40] Keita Higuchi, Shotaro Sano, and Takeo Igarashi. 2021. Interactive hyperparameter optimization with paintable timelines. In *Proceedings of the 2021 ACM Designing Interactive Systems Conference (DIS '21)*, 1518–1528. DOI: https://doi.org/10.1145/3461778.3462077
[41] Frank Hutter, Holger Hoos, and Kevin Leyton-Brown. 2014. An efficient approach for assessing hyperparameter importance. In *Proceedings of the 31st International Conference on Machine Learning*, Vol. 32, Bejing, China, 754–762.
[42] Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. 2011. Sequential model-based optimization for general algorithm configuration. In *Learning and Intelligent Optimization*. Carlos A. Coello Coello (Ed.), Lecture Notes in Computer Science, Vol. 6683, Springer, Berlin, Germany, 507–523.
[43] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. 2018. *Preface*. Springer, Cham, Switzerland, IX–XI.
[44] Carl Hvarfner, Danny Stoll, Artyr Souza, Marius Lindauer, Frank Hutter, and Luigi Nardi. 2022. piBO: Augmenting acquisition functions with user beliefs for Bayesian optimization. In *Proceedings of the International Conference on Learning Representations*, 1–15.
[45] Kevin Jamieson and Ameet Talwalkar. 2016. Non-stochastic best arm identification and hyperparameter optimization. In *Artificial Intelligence and Statistics*, 240–248.
[46] Liu Jiang, Shixia Liu, and Changjian Chen. 2019. Recent research advances on interactive machine learning. *Journal of Visualization* 22, 2 (Apr. 2019), 401–417. DOI: https://doi.org/10.1007/s12650-018-0531-1
[47] Arlind Kadra, Marius Lindauer, Frank Hutter, and Josif Grabocka. 2021. Well-tuned simple nets excel on tabular datasets. In *Advances in Neural Information Processing Systems*. A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (Eds.), Vol. 34, 23928–23941.
[48] Kirthevasan Kandasamy, Karun Raju Vysyaraju, Willie Neiswanger, Biswajit Paria, Christopher R. Collins, Jeff Schneider, Barnabás Póczos, and Eric P. Xing. 2020. Tuning hyperparameters without grad students: Scalable and robust Bayesian optimisation with dragonfly. *Journal of Machine Learning Research* 21 (2020), 81:1–81:27.
[49] Florian Karl, Tobias Pielok, Julia Moosbauer, Florian Pfisterer, Stefan Coors, Martin Binder, Lennart Schneider, Janek Thomas, Jakob Richter, Michel Lang, et al. 2023. Multi-objective hyperparameter optimization in machine learning—an overview. *ACM Transactions on Evolutionary Learning and Optimization* 3, 4, Article 16 (2023), 1–50.
[50] Thanh Tung Khuat, David Jacob Kedziora, and Bogdan Gabrys. 2023. The roles and modes of human interactions with automated machine learning systems: A critical review and perspectives. *Foundations and Trends in Human–Computer Interaction* 17, 3–4 (2023), 195–387.
[51] Doris Jung-Lin Lee and Stephen Macke. 2020. A human-in-the-loop perspective on AutoML: Milestones and the road ahead. *IEEE Data Engineering Bulletin*. Retrieved from https://par.nsf.gov/biblio/10161752
[52] Cheng Li, Sunil Gupta, Santu Rana, Vu Nguyen, Antonio Robles-Kelly, and Svetha Venkatesh. 2020. Incorporating expert prior knowledge into experimental design via posterior sampling. arXiv:2002.11256. Retrieved from https://arxiv.org/abs/2002.11256
[53] Cheng Li, Santu Rana, Sunil Gupta, Vu Nguyen, Svetha Venkatesh, Alessandra Sutti, David Rubin, Teo Slezak, Murray Height, Mazher Mohammed, et al. 2018. Accelerating experimental design by incorporating experimenter hunches. In *Proceedings of the IEEE International Conference on Data Mining*, 257–266.
[54] Lisha Li, Kevin G. Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. 2018. Hyperband: A novel Bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research* 18 (2018), 6765–6816.
[55] Marius Lindauer, Katharina Eggensperger, Matthias Feurer, André Biedenkapp, Difan Deng, Carolin Benjamins, René Sass, and Frank Hutter. 2022. SMAC3: A versatile Bayesian optimization package for hyperparameter optimization. *Journal of Machine Learning Research* 23, 54 (2022), 1–9.
[56] Marius Lindauer, Florian Karl, Anne Klier, Julia Moosbauer, Alexander Tornede, Andreas Mueller, Frank Hutter, Matthias Feurer, and Bernd Bischl. 2024. Position: A call to action for a human-centered AutoML paradigm. arXiv:2406.03348. Retrieved from https://arxiv.org/pdf/2406.03348
[57] K. Louise Barriball and Alison While. 1994. Collecting data using a semi-structured interview: A discussion paper. *Journal of Advanced Nursing* 19, 2 (1994), 328–335.
[58] Neeratyoy Mallik, Edward Bergman, Carl Hvarfner, Danny Stoll, Maciej Janowski, Marius Lindauer, Luigi Nardi, and Frank Hutter. 2023. PriorBand: Practical hyperparameter optimization in the age of deep learning. In *NIPS '23: Proceedings of the 37th International Conference on Neural Information Processing Systems*. Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (Eds.), 7377–7391. Retrieved from https://dl.acm.org/doi/10.5555/3666122.3666445
[59] Michele J. McIntosh and Janice M. Morse. 2015. Situating and constructing diversity in semi-structured interviews. *Global Qualitative Nursing Research* 2 (2015), 2333393615597674.
[60] Gábor Melis, Chris Dyer, and Phil Blunsom. 2018. On the state of the art of evaluation in neural language models. In *Proceedings of the International Conference on Learning Representations*.
[61] Douglas C. Montgomery. 2017. *Design and Analysis of Experiments*. Wiley.

[62] Julia Moosbauer, Julia Herbinger, Giuseppe Casalicchio, Marius Lindauer, and Bernd Bischl. 2021. Explaining hyperparameter optimization via partial dependence plots. In *Advances in Neural Information Processing Systems*, 2280–2291.

[63] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. 2020. A metric learning reality check. In *Proceedings of the European Conference on Computer Vision*, 681–699.

[64] Michael D. Myers and Michael Newman. 2007. The qualitative interview in IS research: Examining the craft. *Information and Organization* 17, 1 (Jan. 2007), 2–26. DOI: https://doi.org/10.1016/j.infoandorg.2006.11.001

[65] Randal S. Olson and Jason H. Moore. 2019. TPOT: A tree-based pipeline optimization tool for automating machine learning. In *Automated Machine Learning*. Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren (Eds.), Springer International Publishing, Cham, 151–160. Retrieved from https://link.springer.com/book/10.1007/978-3-030-05318-5

[66] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollar. 2020. Designing network design spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).*

[67] Anil Ramachandran, Sunil Gupta, Santu Rana, Cheng Li, and Svetha Venkatesh. 2020. Incorporating expert prior in Bayesian optimisation via space warping. *Knowledge-Based Systems* 195 (2020), 105663.

[68] Jan N. Van Rijn and Frank Hutter. 2018. Hyperparameter importance across datasets. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2367–2376.

[69] René Sass, Eddie Bergman, André Biedenkapp, Frank Hutter, and Marius Lindauer. 2022. DeepCAVE: An interactive analysis tool for automated machine learning. *ICML Workshop on Adaptive Experimental Design and Active Learning in the Real World (ReALML).*

[70] Barry Schwartz. 2004. *The Paradox of Choice: Why More Is Less* (1st ed.). Ecco, NY.

[71] Sarah Segel, Helena Graf, Alexander Tornede, Bernd Bischl, and Marius Lindauer. 2023. Symbolic explanations for hyperparameter optimization. In *AutoML Conference 2023.*

[72] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. 2016. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE* 104, 1 (2016), 148–175.

[73] Sebastian Simon, Nikolay Kolyada, Christopher Akiki, Martin Potthast, Benno Stein, and Norbert Siegmund. 2023. Exploring hyperparameter usage and tuning in machine learning research. In *Proceedings of the 2023 IEEE/ACM 2nd International Conference on AI Engineering–Software Engineering for AI*, 68–79. DOI: https://doi.org/10.1109/CAIN58948.2023.00016

[74] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. 2012. Practical Bayesian optimization of machine learning algorithms. In *Proceedings of the 25th International Conference on Neural Information Processing Systems, Volume 2*, 2951–2959.

[75] Jasper Snoek, Kevin Swersky, Rich Zemel, and Ryan Adams. 2014. Input warping for Bayesian optimization of non-stationary functions. In *Proceedings of the International Conference on Machine Learning*, 1674–1682.

[76] Arthur Souza, Luigi Nardi, Leonardo Oliveira, Kunle Olukotun, Marius Lindauer, and Frank Hutter. 2021. Bayesian optimization with a prior for the optimum. In *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference (ECML PKDD '21).* Nuria Oliver, Fernando Pérez-Cruz, Stefan Kramer, Jesse Read, and Jose A. Lozano (Eds.), Lecture Notes in Computer Science, Vol. 12977, 265–296. Retrieved from https://link.springer.com/chapter/10.1007/978-3-030-86523-8_17

[77] Thilo Spinner, Udo Schlegel, Hanna Schäfer, and Mennatallah El-Assady. 2020. explAIner: A visual analytics framework for interactive and explainable machine learning. *IEEE Transactions on Visualization and Computer Graphics* 26, 1 (2020), 1064–1074.

[78] Yuan Sun, Qiurong Song, Xinning Gui, Fenglong Ma, and Ting Wang. 2023. AutoML in the wild: Obstacles, workarounds, and expectations. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23).* New York, NY, Article 247, 15 pages. DOI: https://doi.org/10.1145/3544548.3581082

[79] Kevin Swersky, Jasper Snoek, and Ryan Prescott Adams. 2014. Freeze-thaw Bayesian optimization. arXiv:1406.3896. Retrieved from https://arxiv.org/pdf/1406.3896

[80] Ryan Turner, David Eriksson, Michael McCourt, Juha Kiili, Eero Laaksonen, Zhen Xu, and Isabelle Guyon. 2021. Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020. In *Proceedings of the NeurIPS 2020 Competition and Demonstration Track*. Hugo Jair Escalante and Katja Hofmann (Eds.), Vol. 133, 3–26.

[81] Joaquin Vanschoren. 2019. Meta-learning. In *Automated Machine Learning*. Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren (Eds.), Springer International Publishing, Cham, Switzerland, 35–61. Retrieved from https://link.springer.com/chapter/10.1007/978-3-030-05318-5_2

[82] Chi Wang, Qingyun Wu, Markus Weimer, and Erkang Zhu. 2021. FLAML: A fast and lightweight AutoML library. In *Machine Learning and Systems*, 434–447.

[83] Dakuo Wang, Josh Andres, Justin D. Weisz, Erick Oduor, and Casey Dugan. 2021. AutoDS: Towards human-centered automation of data science. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. New

York, NY, Article 79, 12 pages.

[84] Dakuo Wang, Q. Vera Liao, Yunfeng Zhang, Udayan Khurana, Horst Samulowitz, Soya Park, Michael Muller, and Lisa Amini. 2021. How much automation does a data scientist want? arXiv:2101.03970. Retrieved from https://arxiv.org/pdf/2101.03970

[85] Dakuo Wang, Justin D. Weisz, Michael Muller, Parikshit Ram, Werner Geyer, Casey Dugan, Yla Tausczik, Horst Samulowitz, and Alexander Gray. 2019. Human-AI collaboration in data science: Exploring data scientists' perceptions of automated AI. *Proceedings of the ACM on Conference on Human-Computer Interactions* 3, CSCW, Article 211 (Nov. 2019), 24 pages.

[86] Qianwen Wang, Yao Ming, Zhihua Jin, Qiaomu Shen, Dongyu Liu, Micah J. Smith, Kalyan Veeramachaneni, and Huamin Qu. 2019. ATMSeer: Increasing transparency and controllability in automated machine learning. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. New York, NY, 1–12. DOI: https://doi.org/10.1145/3290605.3300911

[87] Doris Xin, Eva Yiwei Wu, Doris Jung-Lin Lee, Niloufar Salehi, and Aditya Parameswaran. 2021. Whither AutoML? Understanding the role of automation in machine learning workflows. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. New York, NY, Article 83, 16 pages. DOI: https://doi.org/10.1145/3411764.3445306

[88] Li Yang and Abdallah Shami. 2020. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing* 415 (Nov. 2020), 295–316. DOI: https://doi.org/10.1016/j.neucom.2020.07.061

[89] Baohe Zhang, Raghu Rajan, Luis Pineda, Nathan Lambert, André Biedenkapp, Kurtland Chua, Frank Hutter, and Roberto Calandra. 2021. On the importance of hyperparameter optimization for model-based reinforcement learning. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, 4015–4023.

[90] Marc-André Zöller and Marco F. Huber. 2021. Benchmark and survey of automated machine learning frameworks. *Journal of Artificial Intelligence Research* 70 (2021), 409–472.

[91] Marc-André Zöller, Fabian Mauthe, Peter Zeiler, Marius Lindauer, and Marco F. Huber. 2023. Automated machine learning for remaining useful life predictions. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, 2907–2912.

[92] Marc-André Zöller, Waldemar Titov, Thomas Schlegel, and Marco F. Huber. 2023. XAutoML: A visual analytics tool for understanding and validating automated machine learning. *ACM Transactions on Interactive Intelligent Systems* 13, 4, Article 28 (2023), 1–39, 39 pages.

Appendix

A Interview Guide

The following presents the interview guide that we used in the semi-structured interviews.

A.1 Briefing

Dear study participant,
Thank you for supporting our research on HPO in ML with your time and expertise. The interview will take about 30 minutes of your time. All data collected will be treated confidentially and reported only in aggregated form. Your responses will not be linked to your identity in any future publications.

HPO is an increasingly important topic in ML research, as it helps enhance model performance and enables objective comparisons of ML methods. In this study, we define HPO as the process of iteratively improving hyperparameter configurations during ML model training. A hyperparameter is a parameter whose value controls the learning process.

Various tools and ML libraries (e.g., Keras Tuner) support automated hyperparameter optimization. However, it remains unclear why practitioners often do not use advanced automated HPO,[3] and in which contexts manual HPO may even outperform automated optimization and vice versa. Our goal in this study is to understand the motives of practitioners for choosing different HPO methods.

[3]Xavier Bouthillier and Gaël Varoquaux. Survey of machine learning experimental methods at NeurIPS2019 and ICLR2020. [Research Report] Inria Saclay Ile de France. 2020. https://hal.science/hal-02447823/document.

All questions in this interview relate to your personal experience in ML, including the HPO methods you have used for hyperparameter optimization. To obtain detailed information on your work and its context, we ask you to refer to one ML model you developed for productive use in academia or industry. If possible, please provide a link to the publication or associated repository (e.g., GitHub).

The remainder of the interview is structured into three sections:

(1) *Hyperparameter Optimization in Machine Learning*. We will discuss the ML project you will reference in your responses and seek to understand your choice of HPO methods.
(2) *Participant Background and Personal Experiences*. We will ask about your demographics, background, and expertise in HPO and ML.
(3) *Debriefing*. We will summarize the key findings from the interview and outline the next steps.

While we have prepared several questions, we welcome any additional insights you wish to share.

We appreciate your participation and encourage you to share your experiences and perspectives on HPO approaches. There are no wrong answers.

Thank you again for your time and participation!

A.2 Hyperparameter Optimization in Machine Learning

This section is divided into two subsections. First, we will ask you to select one of your previous ML projects to focus on during the interview and to describe the HPO methods you used. Second, we will ask about your reasons for choosing these methods.

A.2.1 Hyperparameter Optimization in a Selected ML Project.

Details on Machine Learning Model. In the interview, we would like to refer to one of your previous ML projects. Please select an ML project that was meant for productive use in academia or industry and where you optimized hyperparameters.

(1) *Optional*: To which repository refer your following answers? Please provide the link to the repository of your model.
URI:____
(2) In case your repository includes multiple models: please provide information to which exact ML model your following answers will refer.
____
(3) Was you ML model empirically evaluated (e.g., in the form of a benchmark)?
☐ Yes ☐ No

Optimization Approach. Next, we want to learn more about how you tuned hyperparameters of the selected ML model.

> *Interviewer Instruction*: Question 4 must be answered "Yes." Otherwise, the interviewee must not participate in the study.

(4) Did you optimize the hyperparameters of the model selected for this interview?
☐ Yes ☐ No

(5) How did you tune your hyperparameters? If you have used a combination of different HPO methods, name all methods used.
Exemplary answers are Bayesian optimization, grid search, manual tuning, and random search.
____

A.2.2 Motives to Use the Selected Hyperparameter Optimization Methods.

> *Interviewer Instruction*: This section builds on the responses from the previous section. The used HPO methods determine whether the questions need to be inverted by using "*(not)*," as indicated in the following questions.

In the previous section, you indicated the HPO methods you used for your ML model: `[LIST OF USED HPO METHODS]`. We are now interested in your reasons for choosing these methods, as well as the advantages and disadvantages you experienced. We value your personal experiences and will not judge your responses—there are no wrong answers.

(6) Which goals (e.g., increase ML model comprehension) did you try to reach through HPO by the chosen methods? A goal is an idea of the future or desired results where the achievement of the idea is decidable. If multiple HPO methods have been used, please explain in which order you used them and explain the individual goals you tried to achieve.
____
____
____

(7) What potential advantages and disadvantages regarding the achievement of the described goals are you aware of with respect to the (combination of) HPO methods you have chosen? What potential advantages and disadvantages did you encounter in achieving these goals with the selected HPO methods? If you used multiple HPO methods, please specify which method each advantage and disadvantage refers to.
____
____
____

(8) Why did you *(not)* choose random search? Please consider contextual factors (e.g., limited compute) and goals (e.g., improve model comprehension).
____
____
____

(9) Why did you *(not)* manual tuning for HPO? Please consider contextual factors (e.g., limited compute) and goals (e.g., improve model comprehension).
____
____
____

(10) Why did you *(not)* grid search for HPO? Please consider contextual factors (e.g., limited compute) and goals (e.g., improve model comprehension).
____
____
____

(11) Why did you *(not)* choose Bayesian optimization for HPO? Please consider contextual factors (e.g., limited compute) and goals (e.g., improve model comprehension).

____
____
____

(12) Which tools did you use for HPO? Exemplary tools are HPBandster, Hyperopt, Ray, spearmint, and Ax.

____
____
____

(13) How could future research support you in optimizing hyperparameters?

____
____
____

(14) After you have helped us with your expertise, we would be grateful if you would share your thoughts on the contexts in which automated HPO may outperform manual HPO and vice versa.

____
____
____

A.3 Participant Background and Personal Experiences

In this section, we will ask you questions about your experience in AI to better understand the context of your previous answers. Your personal information will not be disclosed in any way that could identify you.

(15) What area(s) of AI do you focus on?
Exemplary answers are automatic speech recognition, computer vision, and NLP.

____

(16) What is your main activity in AI?
Exemplary answers are consulting, development of statistical methods, library development, software solution development (i.e., using existing libraries and methods), and use case development.

____

(17) What is/are your area(s) of expertise?
Exemplary answers are AutoML, parallel computing, or (un-)supervised learning.

____

(18) In which field do you work?
Exemplary answers are academia, automotive, finance, IT support and services, and pharma.

____

(19) How many years of experience do you have in ML?
☐ <2 ☐ 2–4 ☐ 5–7 ☐ 8–10 ☐ 11–15 ☐ >15

(20) What is your highest educational level?
Exemplary answers are Bachelor of Engineering, Bachelor of Science, Master of Science, and PhD.

____

(21) What is your profession?
Exemplary answers are big data engineer, developer, data scientist, and ML engineer.

____

(22) What is your position in your organization?
Exemplary answers are professor, PhD student, or lead software architect.

____

(23) What is your age?
☐ <20 ☐ 20–25 ☐ 26–30 ☐ 31–35 ☐ 36–40 ☐ 41–45 ☐ 46–50 ☐ 51–55 ☐ 56–60 ☐ 61–65 ☐ > 65

(24) In which country are you primarily working or employed?

____

(25) How many people are employed at your organization?
☐ <10 ☐ 10–50 ☐ 51–150 ☐ 150–500 ☐ 501–1,000 ☐ 1,000

A.4 Debriefing

This is the final section of the interview. Thank you again for supporting our research with your valuable time and expertise.

> *Interviewer Instruction*: Please summarize the key findings from the interview.

(26) Would you like to receive a summary of the study results?

(27) Do you know colleagues who might be interested in participating in this study? If yes, could you please connect us with them?

> *Interviewer Instruction*: Outline the next steps.

We have now reached the end of the interview. Thank you very much for your participation and support!

Received 23 September 2024; revised 9 May 2025; accepted 21 May 2025