Artificial Intelligence Review (2025) 58:268
https://doi.org/10.1007/s10462-025-11266-y

Synergistic integration of metaheuristics and machine learning: latest advances and emerging trends

Ruining Zhang[1] · Jian Wang[2] · Chanjuan Liu[1] · Kaile Su[3] · Hisao Ishibuchi[4] · Yaochu Jin[5]

Accepted: 12 May 2025 / Published online: 4 June 2025

Abstract
Metaheuristic algorithms (MH) and machine learning (ML) are important components of artificial intelligence (AI). The synergy between MH's optimization search capabilities and ML's data analysis strengths has proven to be highly effective, providing a powerful combination for delivering high-quality solutions across diverse fields. Particularly in real-world applications such as autonomous driving and healthcare, the integration of MH and ML can significantly enhance the intelligence level and decision-making efficiency of systems, addressing the urgent societal and industrial demands for high-efficiency and high-precision solutions. This paper clarifies the logic behind the surge in research on MH-ML hybrid algorithms and addresses the gaps in current reviews regarding their timeliness and breadth of perspective. We begin by elucidating the fundamental concepts underpinning MH and ML, followed by a comprehensive classification framework that categorizes and synthesizes the latest research findings systematically. The paper concludes with an exploration of the challenges inherent in MH-ML hybrid algorithms and proposes future research directions. The analysis of the collected literature demonstrates that the integration of MH and ML generally enhances the performance of algorithms in specific problems. Despite progress from simple combinations to deeper integrations, challenges such as theoretical lag and interpretability remain. Future research will focus on solving these challenges by exploring further integration, using open-source tools, and adapting across diverse domains to expand the use of MH-ML hybrid algorithms.

Keywords Metaheuristic algorithms · Machine learning · Hybrid algorithms

1 Introduction

As the Internet, big data, and artificial intelligence (AI) technologies advance rapidly, they are breaking down silos and knitting once-discrete systems into interconnected complex systems (McGill et al. 2021). The complexity of these systems has amplified the need for sophisticated techniques to address the challenges they present (Gill et al. 2022). AI, in particular, has risen to the forefront of societal, economic, and technological evolution, with its

Extended author information available on the last page of the article innovations propelling both commercial and societal progress (Ng et al. 2021). Within this context, metaheuristic (MH) and machine learning (ML) algorithms have become the two mainstream approaches of AI, excelling in tasks such as pattern recognition, prediction, and decision-making within advanced intelligent systems.

1.1 Research background and significance

1.1.1 Overview of metaheuristics

MH algorithms are a suite of algorithms designed to tackle intricate optimization challenges, seeking to identify the best or near-best solutions via search and optimization processes. The essence of MH lies in its iterative approach to narrowing down to the optimal solution by exploring the solution space (Halim et al. 2021). Over its evolution, MH has undergone significant enhancements and transformations. In the mid-20th century, pioneers began experimenting with heuristic searches to address real-world complexities (Winch 1947; Newell et al. 1957). The 1980 s marked a surge in MH's prominence, fueled by leaps in computer technology that enabled the assessment of numerous solution candidates, thus facilitating the discovery of optimal or near-optimal outcomes. In this era, several foundational MH algorithms emerged, including Particle Swarm Optimization (Kennedy and Eberhart 1995), Genetic Algorithm (Holland 1973), Simulated Annealing (Kirkpatrick et al. 1983), and Ant Colony Algorithm (Dorigo et al. 1991). Since the beginning of the 21st century, MH saw an exponential growth in development. Researchers have been at the forefront of discovering innovative heuristics and search strategies, thereby bolstering the performance and adaptability of MH algorithms. The expansion of computing power and the advent of large-scale parallel computing have rendered the parallel and distributed deployment of MH a critical research area (Cahon et al. 2004). Moreover, the integration of MH with other disciplines has given rise to many innovative methodologies and models, such as hybrid optimization strategies and ensemble learning techniques. Overall, MH algorithms and their hybrid variants hold vast potential for application across various fields. Choosing the right algorithm requires a meticulous evaluation of both the available computational resources and the specific problem to be solved.

1.1.2 Overview of machine learning

ML is a pivotal technology in realizing Artificial Intelligence (AI), enabling the analysis of data patterns and structures to facilitate learning, reasoning, and decision-making autonomously Tiwari et al. (2023). Over the past few decades, ML has evolved from its roots in symbolic learning and decision trees to the sophisticated neural networks and deep learning (DL) methodologies of today. The resurgence of neural networks in the 1980 s signaled the start of a new era in ML history. Since the dawn of the 21st century, ML has made remarkable strides in various domains, notably in speech recognition and image classification. Within the realm of ML, significant advancements have been achieved in Reinforcement Learning (RL) (Ladosz et al. 2022) and unsupervised learning (Sindhu Meena and Suriya 2020), with the latter playing a crucial role in handling data without explicit labels. Recently, transfer learning (TL) (Zhuang et al. 2021) and self-supervised learning (Wang et al. 2022) have emerged as research hotspots, fueled by the surge in big data and the availability of substantial computing resources. These advancements have been instrumental in tackling complex tasks such as image generation, with the introduction of cutting-edge models like Generative Adversarial Networks (GANs) (Gui et al. 2023) marking a significant milestone.

The application spectrum of ML has expanded beyond its traditional domains in finance and healthcare, now penetrating frontier fields like autonomous driving and game AI, underscoring ML's transformative potential and value (Bertolini et al. 2021). As the demand for interpretability, adaptability, privacy protection, and ethical considerations in ML models grows, future research is expected to prioritize not only technical innovation but also the practical applicability and societal impact of ML solutions.

1.2 Main contributions and our approach

1.2.1 Gaps and opportunities: limitations of current MH-ML reviews

The synergistic combination of MH and ML algorithms has recently garnered significant interest in the field of artificial intelligence. This partnership is mutually beneficial: ML leverages the data to uncover actionable insights that refine decision-making processes. These insights lead to substantial improvements in the quality of solutions, the swiftness of convergence, and the overall performance of MH algorithms. Conversely, MH plays a crucial role in optimizing the computational efficiency and cost of ML models, streamlining the process of establishing accurate predictive models. Moreover, by fine-tuning the parameters of ML models, MH algorithms can elevate the accuracy and reliability of their output.

This reciprocal inspiration between MH and ML is paving the way for innovative research avenues, particularly in addressing the intricate challenges posed by complex systems. To effectively tackle these challenges, a thorough grasp of the latest breakthroughs in MH-ML integration is essential. Such knowledge will catalyze the development of hybrid algorithms and frameworks that not only offer enhanced performance but also meet the demands of solving complex, real-world problems with greater efficacy and intelligence.

The combination of MH and ML offers unique advantages in solving complex system problems. Complex systems are typically characterized by high dimensionality, nonlinearity, and multimodality, making them difficult to address with traditional single-algorithm approaches. The MH-ML hybrid algorithm leverages the global search capabilities of metaheuristics and the local optimization capabilities of machine learning to achieve a balance between search efficiency and precision, while also demonstrating strong adaptability and self-learning abilities. In contrast, other hybrid approaches, such as ML for quantum computing (ML-QC) or the combination of MH with other AI paradigms, have their own strengths but may fall short in certain aspects. For instance, ML-QC has the potential to handle exponentially complex problems and offers computational advantages for large-scale problems, but its hardware and algorithms are still immature and complex to implement (Cerezo et al. 2022). Gradient-based ML optimization methods converge quickly in convex optimization problems but are prone to local optima in the non-convex, nonlinear problems common in complex systems, and are sensitive to initial values and learning rates (Zhu et al. 2024). The MH-ML hybrid algorithm also excels in the integration of data-driven and model-driven approaches, maintaining high performance even with scarce data or high noise levels.

Some review articles have synthesized the research on the synergistic integration of MH and ML algorithms, contributing to a more comprehensive view of this interdisciplinary domain. These reviews serve multiple critical functions in advancing research and consolidating understanding as we tackle challenges in complex systems. Previous work on the integration of MH and ML prior to 2019 has been comprehensively summarized and analyzed in existing reviews, providing readers with a systematic retrospective of the early developments in this field (Memeti et al. 2018; Houssein 2019; Talbi 2021; Oliva et al. 2021).

Regarding the currency of related research, the last half-decade has witnessed remarkable progress in technologies such as RL, TL, and GANs. This progress is complemented by the swift evolution of neural architecture search (NAS), evolutionary reinforcement learning (ERL), and large language models, which collectively introduce novel opportunities and challenges to the integration of MH and ML (Donatti et al. 2024), Ojha et al. (2017), and Birattari and Kacprzyk (2009). Consequently, staying abreast of the latest research developments and technological trends is essential for maintaining relevance and impact in this dynamic field. Over the past five years, studies concerning these advanced technologies have demonstrated increasing practical value, owing to their remarkable efficacy in addressing real-world challenges (Ng et al. 2023). By concentrating on the research conducted within this timeframe, we can not only synchronize with the rapid pace of technological advancement but also gain a profound comprehension of the current state and future trajectory of research in MH and ML. This focus offers invaluable insights and direction for driving innovation and practical application across related domains.

When it comes to the scope of review focus, most existing literature reviews tend to exclusively explore the research landscape from either the ML-assisted MH or MH-assisted ML perspective. In contrast, a select few offer a more holistic examination of the synergistic enhancements between MH and ML, considering both ML-assisted MH and MH-assisted ML dimensions. Overall, the existing reviews can be divided into three distinct categories.

The first category, which adopts the ML-assisted MH viewpoint, primarily concentrates on optimization challenges, as exemplified by studies such as Costa Oliveira et al. (2023), Karimi-Mamaghan et al. (2022) and Szénási and Légrádi (2024). Nevertheless, only a few of these reviews consider how ML and MH work together.

The second category, including reviews like Akhter et al. (2019) and Saha et al. (2022), takes the MH-assisted ML perspective, highlighting discoveries and progress within particular domains. Yet, these reviews are often limited in scope, covering a narrow range of MH and ML types, as seen in works such as Chiroma et al. (2020), Elaziz et al. (2021), Akay et al. (2022), Abdallaoui et al. (2022), Devikanniga et al. (2019), José-García et al. (2023), Kaleybar et al. (2023), and Wang et al. (2024). Given the anticipated shift towards more problem-oriented and interdisciplinary hybrid algorithm research, there is a pressing need for reviews that offer a wider and more integrated perspective to foster scientific progress.

The third category, represented by reviews such as Calvet et al. (2017), Alkabbani et al. (2021), and Li et al. (2024), stands out by acknowledging both ML-assisted MH and MH-assisted ML synergies. The synthesis and critical analysis of literature within these reviews not only deepen our understanding of the field's evolution but also give practitioners invaluable insights and direction for real-world applications. However, such reviews often fail to cover all types of MH and ML algorithms, and the scope of the included literature is usually limited to specific domains. Table 1 summarizes the categories and application fields of the representative reviews under the proposed classification method.

Table 1 Summary of the included reviews

Focus	Scope	Review	Year	Application	Standardized metrics
ML-assisted MH	General	Szénási and Légrádi (2024)	2024	Unbounded domains	None
ML-assisted MH	General	Costa Oliveira et al. (2023)	2023	Unbounded domains	None
ML-assisted MH	General	Karimi-Mamaghan et al. (2022)	2022	Combinatorial optimization problems	None
MH-assisted ML	Only evolutionary computation and GANs	Wang et al. (2024)	2024	Evolutionary-based GANs	IS, FID, MMD, SWD
MH-assisted ML	Only genetic algorithm and neural networks	Kaleybar et al. (2023)	2023	Rail vehicle systems	None
MH-assisted ML	Only Biclustering	José-García et al. (2023)	2023	Unbounded domains	bSIZE, VAR, MSR, SMSR, ACF, ACV, VE, CVF
MH-assisted ML	General	Saha et al. (2022)	2022	Sensor-based human activity recognition systems	None
MH-assisted ML	General	Devikanniga et al. (2019)	2022	Hybrid artificial neural networks applied to classification and prediction	MSE, SSE, RMSE
MH-assisted ML	General	Abdallaoui et al. (2022)	2022	Path planning for autonomous vehicles in navigation tasks	None
MH-assisted ML	General	Akay et al. (2022)	2022	Optimization of deep neural networks	None
MH-assisted ML	Only swarm intelligence, evolutionary computing and deep neural networks	Elaziz et al. (2021)	2021	Optimization employed to enhance deep neural networks performance	None
MH-assisted ML	Only nature inspired algorithms and DL	Chiroma et al. (2020)	2020	Application of natured inspired MH algorithms in DL	None
MH-assisted ML	General	Akhter et al. (2019)	2019	Forecasting of photovoltaic power generation	None
Both	Only evolutionary algorithms and RL	Li et al. (2024)	2024	Evolutionary reinforcement learning	None
Both	General	Alkabbani et al. (2021)	2021	Forecasting of renewable power	None
Both	General	Calvet et al. (2017)	2017	Combinatorial optimization problems with dynamic inputs	None
Both	General	Ours	2025	Unbounded domains	None

While most reviews offer valuable insights from a single viewpoint, there is a clear demand for comprehensive reviews that encapsulate both the ML-assisted MH and MH-assisted ML perspectives.

Firstly, focusing solely on one aspect of the research can result in a partial understanding of the symbiotic relationship between MH and ML. Each approach possesses distinct strengths and weaknesses, and a one-sided summary might inadvertently exaggerate the benefits or drawbacks of one approach while downplaying the other's contributions. This imbalance hinders a fair assessment of their combined performance and impact within hybrid algorithms.

Secondly, adopting a dual-perspective review ensures the alignment and uniformity of information, which is crucial for readers to easily grasp and contrast the varying integrations of MH and ML. This approach prevents the pitfalls associated with duplicating or conflicting information that can arise from merging two single-perspective reviews.

Furthermore, enhancing the classification methods employed in existing reviews is essential for delivering a more precise overview of emerging combination methodologies and developmental trajectories. By refining these methods, we can enhance our collective understanding of new technologies and trends, enabling reviews to stay attuned to the rapidly evolving landscape of research.

1.2.2 Literature retrieval and statistics

To compile the literature on pertinent topics, we conducted a comprehensive search using various online databases. The literature's relevance to our paper's subject matter was determined by assessing the frequency of predefined keywords within titles, abstracts, keyword lists, and the main text. Our review is concentrated on articles published from 2019 to 2025. Informed by initial search results, we utilized a set of targeted keywords to filter the literature in databases such as Google Scholar and esteemed journals including *IEEE Transactions on Neural Networks and Learning Systems*, *Information Sciences*, *Applied Soft Computing*, *Journal of Artificial Intelligence*, *IEEE Transactions on Cybernetics*, *IEEE Transactions on Evolutionary Computation*, *IEEE Transactions on Fuzzy Systems*, *Artificial Intelligence Review*, *Swarm and Evolutionary Computation*. The keywords we focused on included combinations of "improved metaheuristic", "metaheuristic AND machine learning", as well as terms like "artificial bee colony", "particle swarm optimization", "ant colony optimization", "genetic algorithm", "simulated annealing", and "tabu search", each paired with "machine learning" and "neural networks" as well as "linear regression", "logistic regression", "linear discriminant analysis", "support vector machine", "naïve Bayes", "gradient boosting", "decision tree", "random forest", "k-nearest neighbor", "artificial neural network", "K-means", "shared nearest neighbor clustering", "density-based clustering algorithm", "self-organizing mapping", "principal component analysis", "multiple correspondence analysis", "association rules", "semi-supervised learning", "Q-learning", and "learning automaton", each paired with "metaheuristic". These specific search terms were designed to pinpoint the most relevant and cutting-edge research at the intersection of metaheuristics and machine learning.

Throughout the literature collection process, 184 relevant papers were selected based on whether they clearly and explicitly described the integration of MH and ML, as well as the Chinese Academy of Sciences zone. The gathered literature was categorized and tallied according to the year of publication, as illustrated in Fig. 1. It can be observed that prior to 2020, the number of studies on the combination of MH and ML was relatively limited. From 2020 to 2021, with the rapid development of MH and ML technologies, research on hybrid methods based on MH and ML gradually increased, showing a steady upward trend in the number of related publications. However, in 2022, the research enthusiasm in this field slightly declined. The number of related publications peaked in 2023, followed by a decline in 2024, although it remained at a relatively high level. Due to the incomplete collection of data for 2025, the current statistics show a lower number of publications for that year.

This paper presents a comprehensive summary and analysis of the above literature, categorizing hybrid algorithms of MH and ML into two distinct sections: Machine Learning-Assisted Metaheuristics (ML-assisted MH) and Metaheuristics-Assisted Machine Learning (MH-assisted ML). In the ML-assisted MH approach, MH forms the core algorithm, with ML techniques employed to enhance its performance. Conversely, in the MH-assisted ML approach, ML is the primary algorithm, augmented by MH strategies to boost its effectiveness. Figure 2 organizes the selected papers based on whether they utilize ML-assisted MH or MH-assisted ML methodologies, with a subset of papers employing both approaches. The distribution of these papers is further broken down by publication year. As shown in Fig. 2, in recent years, research on utilizing MH algorithms to assist ML has garnered more attention compared to studies focusing on leveraging ML to enhance MH algorithms.

Relevant literature has demonstrated that hybrid algorithms combining MH and ML approaches exhibit superior performance compared to standalone MH or ML algorithms, as evidenced by a range of evaluation metrics. This paper summarizes the evaluation metrics used in the included studies, with the six most frequently employed metrics being: accuracy, root-mean-squared error (RMSE), mean absolute error (MAE), coefficient of determination (R^2), error, and computational efficiency. These metrics reflect the performance of algorithms from various perspectives. Relevant literature has experimentally validated that the integration of MH and ML approaches demonstrates significant advantages across these

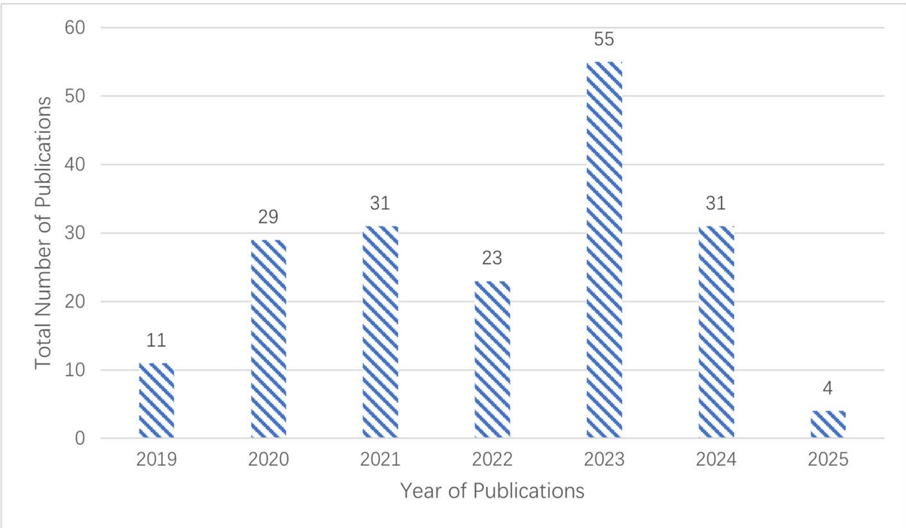

Fig. 1 Timeline analysis of MH and ML integration research (2019-2024)

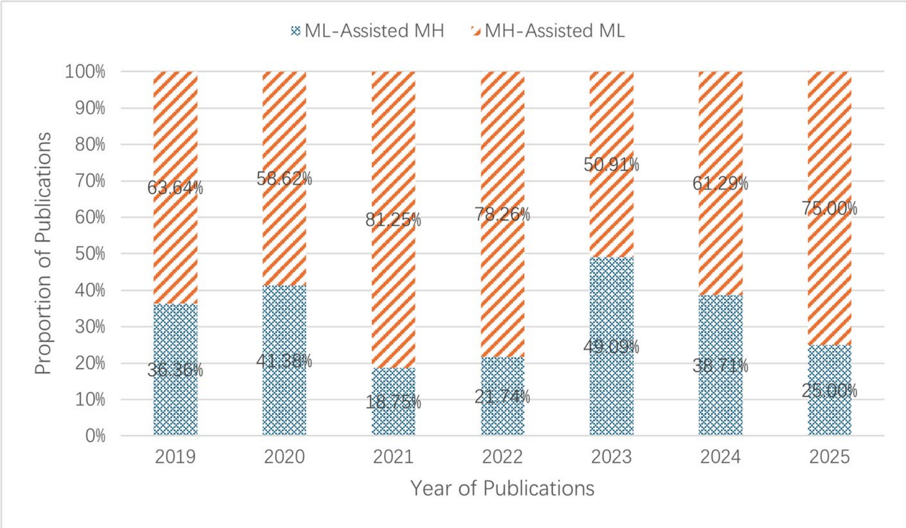

Fig. 2 Proportion of the included literature's category

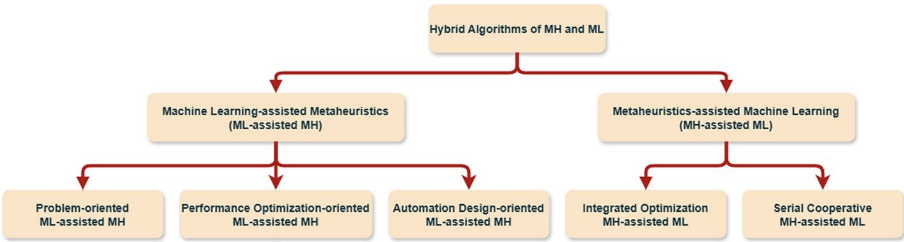

Fig. 3 Hierarchical classification of MH and ML hybrid algorithms

evaluation metrics, thereby proving the superiority of hybrid algorithms in solving complex problems.

1.2.3 Hierarchical classification

Responding to the need for in-depth analysis in complex systems that leverage hybrid algorithms of MH and ML algorithms, this paper offers an exhaustive examination of the diverse algorithms within these domains. It categorizes and synthesizes both domain-specific and general hybrid algorithms that have emerged over the past five years, examining them from the dual perspectives of ML-assisted MH and MH-assisted ML.

In doing so, the paper introduces a more holistic and universal classification framework. This framework facilitates the systematic analysis of ML-assisted MH and MH-assisted ML methods, with ML-assisted MH being divided into three distinct categories and MH-assisted ML into two. The stratification of classification methods used throughout this paper is visually represented in Fig. 3, which will be thoroughly elucidated in Sects. 3 and 4. This figure serves as a roadmap for understanding the organizational structure of the hybrid algorithms discussed, providing clarity on their interrelationships and applications.

Our systematic classification and synthesis of the current literature reveal that the synergistic collaboration between MH and ML significantly bolsters the performance of algorithms when confronting dynamic environments and intricate system challenges. The integrated MH-ML hybrid algorithms not only yield superior solution quality but also outperform individual methodologies in terms of optimization prowess and solution efficiency. This integrated approach addresses complex and nonlinear challenges, uncovers obscured patterns and rules, and fosters self-adaptation and intelligent responses within sophisticated systems, suggesting a wide range of potential applications. The existing body of research has pushed the integration of ML and MH beyond mere amalgamation to a more profound level of deep fusion. This paper provides readers with more comprehensive information and new insights and directions for problems in more fields and applications.

1.3 Layout of this review

The rest of this paper is organized as follows: Section 2 provides an introduction and a detailed classification of both MH and ML. Sections 3 and 4 delve into the specifics of how ML can enhance MH and vice versa, offering a thorough summary of the relevant scholarly work in these areas. Section 5 presents the challenges of hybrid algorithm research, outlines future research prospects, and summarizes the paper.

2 Main concepts

2.1 Metaheuristic (MH) algorithms

MH combines random and local search algorithms derived from heuristic algorithms. MH intelligently combines different concepts, develops and utilizes the search space with heuristic strategies through iterative generation, and thus obtains the approximate optimal solution (Hussain et al. 2019).

MH is usually a general strategy that does not target a specific problem and thus enables a wider range of applications. When solving problems, it is necessary to determine feasible solutions and make improvements to obtain the globally optimal solution. It is challenging to determine the exact global optimal solution for complex and resource-limited optimization problems (Bandaru and Deb 2016). The approximate optimal solution generated by MH can be used to solve this problem. MH can be used to solve nonlinear and non-convex optimization problems (Toscano and Lyonnet 2010). For combinatorial optimization problems, MH can generate more suitable solutions than simple greedy heuristics (Blum and Roli 2003). For optimization problems involving multi-objective functions with nonlinear constraints, population-based MH algorithms usually have advantages over single-solution-based MH algorithms since several non-dominated solutions can be obtained by a single run of population-based MH (Zavala et al. 2014).

MH can be classified from many perspectives, such as natural and non-naturally inspired (Dokeroglu et al. 2019), individual search and population-based (Agushaka and Ezugwu 2022), dynamic and static objective functions (Alba et al. 2013), different neighborhood structures (Santos et al. 2018), etc. According to the number of parallel solutions in each iteration of the search process, MH is divided into single-solution-based metaheuristics (single-solution-based MH) and population-based metaheuristics (population-based MH). The above two methods update the solution iteratively and select the optimal solution satisfying the conditions according to the predefined rules. An iteration can be divided into two phases: a generation phase and a replacement phase.

Single-solution-based metaheuristics (single-solution-based MH)

Single-solution-based MH generates and replaces a single solution to obtain the optimal or approximate optimal solution (Jaddi and Abdullah 2020). Each iteration starts from only one solution, and in the generation phase, a set of candidate solutions is generated by the local transformation of the current solution. In the replacement phase, the current solution is replaced with a new solution selected from the candidate solutions or remains unchanged. The iteration stops when a preset termination criterion is met. The generation and replacement of the iterative process can be performed only based on the current solution, or the search history can be exploited to generate candidate solutions and select new solutions. The currently popular single-solution-based MH algorithms include Simulated Annealing (SA) (Bertsimas and Tsitsiklis 1993) and Tabu Search (TS) (Glover 1990).

Population-based metaheuristics (population-based MH)

Population-based MH obtains the optimal or approximate optimal solution from a set of relatively independent parallel solutions (Beheshti and Shamsuddin 2013). This process is an iteration of the population of solutions. Each iteration starts from an initial population of solutions. In the generation stage, a candidate population is generated from the current population. In the replacement stage, the appropriate solutions are selected from the current and candidate populations to form a new population to replace the current population. The iteration stops when a preset termination criterion is met. The characteristics of the population can be used to guide the search of the solution space. The population update by population-based MH focuses on exploiting unknown areas, has more global exploration ability, and generates more diverse solutions in the whole solution space. The commonly used population-based MH algorithms include evolutionary algorithms (EA) (Bartz-Beielstein et al. 2014), ant colony optimization (ACO) (Dorigo and Stützle 2003), differential evolution (DE) (Price 2013), particle swarm optimization (PSO) (Kennedy and Eberhart 1995), and artificial bee colony algorithm (ABC) (Karaboga et al. 2014).

2.2 Machine learning (ML)

As a branch of artificial intelligence, ML obtains a model through automatic data analysis and then uses the model to predict unknown data (Mahesh et al. 2020). ML has extensive applications in diverse domains such as data mining, computer vision, natural language processing, biometrics, search engines, medical diagnosis, detection of credit card fraud, securities market analysis, DNA sequence sequencing, speech and handwriting recognition, strategy games, and robotics (Jordan and Mitchell 2015).

Many ML methods are available, and three main classification criteria are used in the existing research work. The first approach divides ML into supervised, unsupervised, semi-supervised, and reinforcement learning according to whether it learns under human supervision. The second approach divides ML into online learning and batch learning according to whether it can be dynamically incremental (Burlutskiy et al. 2016). The third approach divides ML into example-based learning (Sung and Poggio 1998) and model-based learning (Bishop 2013), depending on whether ML simply matches the input data with the known data or performs model analysis on the training data to build a prediction model. These three classification criteria categorize ML from different perspectives, yet they are not mutually exclusive but rather complementary. The learning paradigm (supervised, unsupervised, semi-supervised, reinforcement learning) determines the data labeling requirements and learning objectives; the learning mode (online, batch learning) dictates the model update mechanism and data processing approach; the learning strategy (instance-based, model-based) defines the prediction mechanism of the model. Collectively, these classification criteria form a comprehensive understanding of ML, aiding researchers in selecting appropriate ML techniques for specific tasks. This paper adopts the first ML classification criterion, which classifies ML into four types based on the amount and type of supervision it receives during training: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning.

Supervised learning

In supervised learning, the training data set must contain both input vectors and their corresponding output vectors to support the model in learning the mapping relationship between input and output (Hastie et al. 2009). Supervised learning is ML with independent and dependent vectors, where the input vector is the independent vector and the output vector is the dependent vector. The output vector is the label of the input vector to which it is matched, and both are known information to the algorithm. During training, supervised learning will establish a relationship between the input and output vectors, which will be used to predict the output vector corresponding to the input vector in the test data. Supervised learning can be used for classification and regression. Common supervised learning includes linear regression (LR), logical regression (LogR), linear discriminant analysis (LDA), support vector machine (SVM), naïve Bayes (NB), gradient enhancement (GB), decision tree (DT), random forest (RF), k-nearest neighbor (k-NN), artificial neural network (ANN), TL, and so on.

Unsupervised learning

The training data for unsupervised learning is neither classified nor labeled, and the algorithm learns automatically without guidance. The training data of supervised learning has only independent variables without dependent variables. Input variables as independent variables are known information, and there is no output variable as the dependent variable. Unsupervised learning is trained to build a latent model in the training data (Ghahramani 2003).

Unsupervised learning is mainly used for clustering, dimensionality reduction, visualization, and association rule learning. The clustering algorithm divides the data set into different groups according to specific criteria so that the data within the same group is as similar and the data between different groups is as different as possible. Several clustering algorithms are commonly used, such as the K-means algorithm, shared nearest neighbor clustering (SNNC), density-based clustering algorithm (DBSCAN), and maximum expectation algorithm. Another task in unsupervised learning is dimensionality reduction and visualization. For example, dimensionality reduction for feature extraction merges multiple features into a single feature, thus simplifying the data while minimizing the loss of information. The algorithms commonly used for visualization and dimensionality reduction include self-organizing mapping (SOM), principal component analysis (PCA), kernel principal component analysis (KPCA), and multiple correspondence analysis (MCA). Association rules (AR) learning is also a typical application of unsupervised learning. Its common algorithms include the Apriori algorithm and the Eclat algorithm.

Semi-supervised learning

The training set used for semi-supervised learning consists of a small amount of labeled data and a large amount of unlabeled data. Due to the high cost of obtaining data labels, semi-supervised learning can avoid the waste of data and resources and has great practical value. Semi-supervised learning usually combines supervised learning with unsupervised learning while solving problems such as the weak generalization ability of supervised learning models and the imprecision of unsupervised learning models (Van Engelen and Hoos 2020). For example, deep belief networks (DBNs) are constructed by stacking multiple layers of unsupervised restricted boltzmann machines (RBMs). Semi-supervised learning is mostly based on certain assumptions about the data. When the assumptions about the data are invalid, the semi-supervised algorithm cannot show its normal performance, which limits the further application of semi-supervised learning.

Reinforcement learning

RL algorithms mimic humans' trial-and-error learning processes to achieve their goals. RL uses the reward and punishment mode to process the data, learning from the feedback of each operation. By continuously enhancing the operations that contribute to achieving the goal during the learning process and penalizing the operations that lead to deviation from the goal, RL can determine the optimal processing path to achieve the goal (Kaelbling et al. 1996). As the optimal global processing path may require local sacrifice, the strategy of RL learning includes penalty operations, resulting in partial detours in the search path, thereby achieving delayed satisfaction. The autonomous learning ability of RL enables it to obtain the best output results in unknown environments.

The main components of RL are agents and environments. Agents represent the RL itself, while the environment represents the scenario in which it operates. Agents can perceive reward signals from the environment. Reward signals can reflect whether the current state contributes to achieving goals. Agents perform operations to maximize the cumulative reward, known as return. RL accomplishes its goal through the iterative learning of agents. The agent interacts with the environment in each iteration to obtain the state of its environment. Based on the state information, agents use their knowledge to decide the next operation to perform in response to that state. The environment will change with the agents' operation, and the reward signal generated by it will be fed back to the agents. Agents evaluate the last operation and update the agents' knowledge based on the reward signal. RL stops iterating when the agents sense a termination state from the environment (Ding et al. 2020).

Common RL includes Q-learning (QL), learning automaton (LA), opposition-based RL (OPRL), Monte Carlo-based RL (Monte Carlo), SARSA, and deep reinforcement learning (DRL). Classical RL can be classified from multiple perspectives. The main classification methods include model-based and model-free RL, value-based and policy-based RL, and actor-critic, which combines the two; Monte Carlo and temporal-difference RL with model-free learning; and on-policy and off-policy RL. In this paper, according to whether an agent can learn the model of the environment completely, different RL is divided into model-based learning and model-free learning. The environment is modeled as a function that predicts state transitions and rewards.

Model-based learning is usually used when the environment is clearly defined, unchanged, and difficult to test in the real environment (Moerland et al. 2023). The agent first constructs a model of the environment, then simulates the operation sequence according to the probability of the maximum cumulative reward, and then assigns weights to the operation sequence. In the process, the agent generates different policies based on the pre-planning results used to achieve the final goal. The advantage is that the agent can think ahead, and when making each operation, it can try out possible future choices in advance and then choose from the candidates. Model-based learning also has the drawback that the agent usually cannot access a realistic environment model. There are differences between the model built by the agent and the true model, which may cause the agent to have good performance only in the learned model but unsatisfactory performance in the real environment.

Model-free learning is typically utilized in scenarios where the environment is vast, intricate, and challenging to describe (Huang 2020). Model-free learning does not build a model of the environment. Consequently, it does not have the potential gains in the sample efficiency of model-based learning. As an alternative, model-free learning employs a trial-and-error approach in the environment. Model-free learning constructs state-operation pairs and their sequences, and by scoring and recording them, model-free learning generates a policy for selecting operations. Compared with model-based learning, model-free learning is easier to implement and adjust.

3 Machine learning-assisted metaheuristics (ML-assisted MH)

Based on the specific roles of ML in the optimization process, ML-assisted MH can be categorized into three types: problem-oriented ML-assisted MH, performance optimization-oriented ML-assisted MH, and automation design-oriented ML-assisted MH. In the problem-oriented ML-assisted MH, ML serves as a tool to model and analyze specific problems, thereby enhancing MH's problem-solving capabilities. In performance optimization-oriented ML-assisted MH, ML improves the internal mechanism of MH algorithms to improve the operation efficiency and the quality of the solution. In automation design-oriented ML-assisted MH, ML automatically generates or optimizes the algorithm framework, operation strategy, and core components. The classification hierarchy of ML-assisted MH is shown in Fig. 4.

The classification method in this section forms a complete logical chain from problem modeling to performance optimization and then to automated design, comprehensively covering the different roles of ML in MH optimization. This approach facilitates researchers and practitioners in selecting appropriate ML-assisted methods based on practical needs. Compared to the classification of ML-assisted MH into six categories including algorithm selection, fitness evaluation, initialization, evolution, parameter setting, and collaboration (Karimi-Mamaghan et al. 2022), the classification method in this section is more concise and logically structured, avoiding overly fragmented divisions. Additionally, compared to the classification of ML-assisted MH into local-level and global-level hybridizations (Calvet et al. 2017), the method in this section is more specific and better reflects the diverse functionalities of ML in optimization. The classification method in this section not only aligns closely with current research trends but also provides a clear framework and guidance for the research and application of ML-assisted MH.

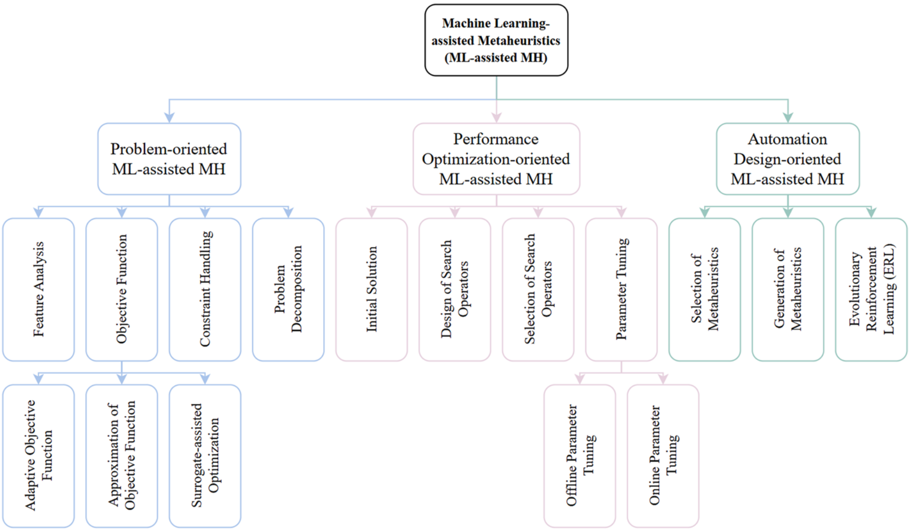

Fig. 4 Classification of ML-assisted MH

3.1 Problem-oriented ML-assisted MH

In the research of ML-assisted MH, optimization according to the problem characteristics is the key to achieving the efficiency and adaptability of the algorithm. The problem-oriented ML-assisted MH methods emphasize that in different stages of the optimization process, ML is fully used to enhance the understanding and modeling of the problem to improve the ability of MH algorithms to solve complex problems. Specifically, optimization problems usually contain multi-dimensional complex characteristics, such as high-dimensional nonlinearity of features, complexity of objective functions, diversity of constraints, and complexity of problem structure. Therefore, problem-oriented ML-assisted MH is divided into four core aspects: feature analysis, objective function, constraint handling, and problem decomposition. These aspects cover several essential links in the process of solving optimization problems.

3.1.1 Feature analysis

Feature analysis is extracting and identifying key features from a dataset. This is done to eliminate any redundant information and provide effective input for model or data analysis. In the field of MH, feature analysis is necessary to help identify the key features of a problem, simplify its complexity, and guide the design and improvement of the algorithm. By delving into data features, MH can search the solution space more efficiently, improve the speed and accuracy of the solution, and thus better solve the problem.

According to the stage during which MH utilized feature-related knowledge, the related work that used ML to complete the feature analysis of MH can be divided into two categories. The first category involves using ML during the execution process, while the second category involves providing the feature-related knowledge as input data to MH before the execution.

Feature-related knowledge used during execution is often used to improve the performance of MH. Lucas et al. (2020) proposes an MH for solving combinatorial optimization problems. The algorithm uses DT to define promising regions in the feature space of TS. By doing so, the search is mainly focused on these promising regions, which minimizes the search time and improves the quality of the solutions found.

Once the feature analysis has been completed before execution, the knowledge related to the features obtained is used as the input data for MH. At this point, the whole process of completing a given task can be divided into two phases. The first phase is a feature analysis based on ML methods, while the second phase is a metaheuristic algorithm. The feature-related knowledge obtained from the first phase helps to improve the performance of the overall method for a given problem. When (El-Kenawy et al. 2021) classifying images, a CNN model named ResNet-50 is applied to feature learning and extraction. Then, the improved MH algorithm selects features from the features extracted by the ResNet-50 model. To solve a multi-objective optimization problem (MOO) with nonlinear constraints, Zhang et al. (2021) first used the Back Propagation Neural Network (BPNN) model to model the relationship between objective functions and feature attributes on the dataset. Then, a multi-objective MH algorithm based on individual intelligence is used to search for the optimal combination of target variables under the defined constraints. Momenitabar et al. (2022) aims to design an efficient and sustainable bioethanol supply chain network (SBSCN). To achieve this, three popular machine learning methods, RF, Extreme Gradient Boosting (XGBoost), and the Adaboost algorithm in ensemble learning, are utilized to predict demand and use it as input for the proposed mathematical model to find the optimal values of strategic, planning, and tactical decision variables. To tackle large and medium-sized problems, two MH algorithms are employed to determine the Pareto optimal solution.

3.1.2 Objective function

In ML-assisted MH, the modeling and optimization of the objective function is the core of the optimization process of MH algorithms, which directly affects the efficiency and performance of the algorithm. The objective function of optimization problems is usually complicated and expensive to calculate. This challenge is especially significant in high-dimensional, multi-objective, or dynamic environments. The introduction of ML can significantly improve the processing capacity of the objective function, reduce the computational complexity, and improve the optimization efficiency and accuracy. To systematically analyze the auxiliary effect of ML on the objective function, this paper divides the cooperative modes acting on the objective function into three categories: adaptive objective function, approximation of objective function, and surrogate-assisted optimization. This division reflects the different angles from which ML improves the objective function. This systematic division and analysis can more fully reveal the multi-level application of ML in objective function optimization and its enhancement to MH algorithms.

Adaptive objective function

The adaptive objective function utilizes the knowledge related to the given problem to adjust the methods or parameters automatically and forms the estimation model of the original objective function. The adaptive objective function usually adopts online learning to extract relevant features and relationships from the states accessed by the MH algorithm during the search process. RL can be used to construct an adaptive objective function.

(Bayliss et al. 2020) proposes a feedback loop that integrates MH algorithms with ML methods. Firstly, the optimization module composed of the MH algorithm proposed a promising candidate solution based on the decision cost predicted by RL. Then, the candidate solution is tested in the reality module by calculating its true cost and accurate data about the decision-making cost. Finally, accurate data on decision-making costs are used for RL training. As part of the feedback loop, RL provides the MH algorithm with a fast prediction of the decision-making cost required for the search.

Approximation of objective function

To find the optimal solution, MH methods require a large amount of computation of the objective function. However, calculating the true objective value of each solution is often expensive and time-consuming, which limits the application of MH methods in practical problems. To mitigate this issue, using ML to approximate the actual value of the objective function can effectively reduce the computational cost of MH. For example, Shukla et al. (2019) combines two MH algorithms using an SVM classifier as its fitness function. SVM can estimate the predictive fitness of the optimal gene subset and then select informative genes from high-dimensional data.

Surrogate-assisted optimization

A surrogate model based on the objective functions and constraints of a given problem can be used as an alternative to replacing the actual objective function under specified conditions to guide the MH algorithm through the search space.

The use of ML-based surrogate models to assist MH methods requires consideration of the following three issues in turn: which ML method should be used to construct the surrogate model, under what circumstances the surrogate model should be embedded in the MH method, and when and how the surrogate model should be updated. For example, Ahmadi et al. (2019) uses three MH algorithms for feature selection, thus establishing a prediction model to evaluate the risk of absenteeism. Three different classifiers, DT, RF, and NB, provide fitness function values for the above MH algorithms. Zhang et al. (2023) uses the optimized ANN as the fitness function of GA to evaluate the fitness of individuals. Xue et al. (2023) constructed four different ANN models to predict different geothermal productivities. These ANN models are used as the objective model of the DE algorithm, substituting the numerical model of the objective function in the DE algorithm. The study in Zhang et al. (2024) treats each particle in PSO as an independent LA. In each iteration, each LA interacts with the environment to adaptively select actions based on its current position, thereby updating the particle's position. The LA dynamically adjusts the action probabilities according to the environmental feedback on the action rewards, promoting adaptive exploration of particles in the search space.

3.1.3 Constraint handling

A given problem usually has a set of constraints associated with it. The handling of constraints is crucial for the construction of MH methods. In addition to the approximation strategy already applied to the objective function, the handling of constraint conditions can also consider transforming them. For more or higher complexity of the constraint, the ML methods can simplify the set of constraints. Kakarash et al. (2023) extends Density Peak Clustering (DPC) to k-nearest Neighbors (KNN) and uses the improved clustering KNN algorithm to assist the MH algorithm in solving the multi-label feature selection problem.

Firstly, the feature space is transformed into a KNN graph, and similar features are grouped by the graph-based DPC method, and the decision graph is used to identify the cluster center. Then, the MH algorithm is used on the cluster graph to make the individuals move between the cluster centers and weigh the features. Finally, the MH algorithm ranks the features according to their pheromone values and filters them with low redundancy. Through KNN, the nonlinear multi-label constraints are transformed into cluster centers that ACO can identify.

3.1.4 Problem decomposition

Smaller scale problems tend to be more beneficial for MH algorithms to solve. ML can assist MH methods in decomposing large-scale problems into relatively small-scale problems. There are two main ideas for decomposing the problem. When the problem-solving process can be carried out step by step, the problem can be decomposed layer by layer with the deepening of the solution. Another decomposition idea is to decompose the original problem into several independent sub-problems by using a similar idea of divide-and-conquer, and the solutions of each sub-problem obtained through parallel mode can be integrated to solve the original problem further. For example, for the urban emergency rescue planning task, Yang et al. (2023) designed a clustering method to identify the area of each rescue location and then used the improved MH algorithm to determine the rescue path of each vehicle. Based on the idea of divide and conquer, clustering transforms multi-objective path planning into single-objective path planning, which simplifies the task of rescue location planning. Li et al. (2023) designed an ML-assisted evolutionary computing framework. The framework identifies the attention subspace by learning a low-rank representation of a problem with limited samples. Evolutionary computation methods explore the attention subspace so as to reliably avoid local optima.

3.2 Performance optimization-oriented ML-assisted MH

Performance optimization-oriented ML-assisted MH refers to improving the operation efficiency and optimization performance of MH algorithms through ML. These methods focus on improving the internal mechanism of the algorithm, such as initialization, search operator design, dynamic selection of search strategy, and parameter optimization. The goal is to improve the global search ability, convergence speed, and solution quality of the algorithm, which is suitable for scenarios with a wide range of optimization problems and obvious algorithm performance bottlenecks. Through a data-driven approach, ML learns the rules in the operation of MH algorithms and helps to dynamically adjust the algorithm's behavior to make it more efficient on complex problems.

The quality of the initial solution or population is crucial to the efficiency and solution performance of MH algorithms. Through ML, high-quality initial solutions can be generated to shorten the search time and improve the convergence speed of the algorithm. Search operators, such as crossover and mutation operators in GA, are the core components of MH algorithms. ML can automatically design new search operators or optimize the operation logic of existing operators to enhance the searchability of the algorithm. The choice of search operator has an important impact on the search effect. In MH algorithms, the effect of the search operator usually varies depending on the type of problem or the stage of optimization. ML can dynamically select the operator that best fits the current state, thus enabling a more intelligent search. The parameters of MH algorithms (such as crossover rate, mutation probability of genetic algorithm, and temperature decay rate of SA) have a significant impact on the performance of the algorithms. Traditional manual parameter tuning is inefficient and depends on experience, while ML provides automatic and dynamic tuning methods. The four specific directions of performance optimization-oriented ML-assisted MH each target different aspects of the MH algorithm, helping the algorithm to achieve more efficient and stable performance in a wider range of scenarios.

3.2.1 Initial solution

MH methods update iteratively based on the initial solution, and the quality of the initial solution has an important influence on the subsequent search process of MH. A suitable initial solution is helpful for MH methods to obtain a higher-quality final solution with lower calculation costs. Initial solutions of MH are generally generated randomly within the search space. The randomly generated solutions are usually of poor quality and cannot guarantee the diversity of their search space and the rationality of the given problem. Since ML methods can extract knowledge from the existing information to generate better-quality solutions, many research works use ML to assist in the initialization process of MH.

Yang and Sutrisno (2020) applies the MH algorithm to solve high-dimensional optimization problems with only one control parameter. The design of the MH algorithm is inspired by the symbiotic interactions between organisms in the ecosystem. To improve the search efficiency of the MH algorithm, k-means clustering with adaptive K-value is used to create the initial subpopulation, thus simplifying the cluster distribution. In Sun et al. (2021), since the initial population of MH algorithm is randomly generated within the search space, MH algorithm is relatively sensitive to the state of the initial population. Dynamic opposition-based learning (DOL) can assist the initialization of the MH algorithm population, reduce the sensitivity of MH algorithm to the initial population state, and avoid premature convergence of the algorithm. Li (2023) maps the initial positions of individuals in the MH algorithm through Local Reverse Learning. To prevent all initial populations from generating their own inverse solutions and thus prolong the iterative search time, the improved MH algorithm constructs a local inverse learning model based on local inverse knowledge. Local OBL improves the initialization process of the MH algorithm, which makes the distribution of the initial population more uniform, enriches the diversity of the initial population, and improves the convergence speed of the algorithm. The study in Premkumar et al. (2024) employs the K-means algorithm to cluster data points in the dataset before applying a population-based MH algorithm, generating initial solutions closer to the potential optimal solution. This approach helps guide the population toward more promising regions of the search space at the early stages, thereby improving the convergence efficiency of the algorithm.

3.2.2 Design of search operators

ML can be used to construct search operators in MH. Search operators can be classified into three categories according to how new solutions are obtained: constructor operators, local search operators, and multivariate operators. Constructor operators are used to generate new candidate solutions directly based on the characteristics of the problem and heuristic rules, usually without involving modifications of existing solutions. The local search operators search in the neighborhood of the current solution to find the local optimal solution. These operators explore the solution space by modifying or perturbing the current solution in a small range and then evaluate the quality of the new solution. The multivariate operator aims to increase the diversity of solutions and avoid the algorithm falling into local optima. It uses the combination of multiple solutions to explore a wider search space. This strategy helps the algorithm jump out of the local optimal solution and find the global better solution. Various MHs may use different terms to describe the above operators or combine the functions of multiple operators into a single operator. In practical applications, it is necessary to select appropriate operators for combination and optimization based on the characteristics of the problem.

Constructor operators

The constructor operator is used to generate new solutions for the MH directly. For offline ML methods, some research works employ unsupervised learning to extract and merge information and use it for new solution generation. Supervised learning, such as ANN, can directly provide new solutions for MH. For example, Rohaninejad et al. (2022) uses the local learning based on regression neural network to explore the neighborhood of each given solution obtained by crossover and mutation operators in each iteration of GA to determine the local optimal solution and improve the quality of the final solution. The data set required for training adaptive local search is generated by the k-means clustering algorithm and non-dominated sorting.

For online ML methods, most of the existing works use RL to construct the constructor operator. Sadeg et al. (2019) uses Q-learning to replace the local search in the MH algorithm. Q-learning enables each individual to learn from its own experience and that of other individuals during the search process, thus improving the quality of the new solutions generated. Seyyedabbasi (2023) combines value-based Q-learning with MH algorithm and controls the fitness function through a parameter R and q value. Parameter R controls the q-valued control, which is responsible for making the most reasonable exploration decisions. Combining the two achieves a balance between utilizing known information and further exploring unknown information, thereby enabling MH algorithm to make more reasonable decisions in generating new solutions. Zhang et al. (2023) uses NNs to provide candidate solutions for the non-dominated sorting genetic algorithm (NSGA-II) and finally obtains the global optimal solution.

Local search operator

The local search operator in MH is mainly used to control the MH to search within a suitable neighborhood. The optimization of the local search operator by ML mainly focuses on two aspects: adjusting the range of the neighborhood and improving the way of sampling in the neighborhood.

On the one hand, ML can improve the efficiency of MH searching in the neighborhood or the quality of new solutions generated by MH by adjusting the size of the neighborhood or the number of nodes contained in the neighborhood. Song et al. (2020) uses neural networks to model the dynamics and characteristics of pheromones in the MH algorithm. The pheromone evaporation model can guide the updating of neural network outputs so that the evolution of the neural network can better simulate the characteristics of pheromones. Chen et al. (2020) constructs a minimum cost heuristic and uses it to insert nodes removed by random order into feasible locations with the least detour, thus solving large-scale combinatorial optimization problems such as the Vehicle Routing Problem (VRP). Hierarchical Recurrent Graph Convolutional Network (HRGCN) provides anchor nodes and path coefficients for the MH algorithm. The study in (Farahmand-Tabar and Ashtari 2024) employs a SOM to identify more promising search regions from the extracted information during the iterative process of a MH algorithm, thereby reducing the search space and optimizing the search strategy of the MH algorithm. To optimize a ruin-and-recreate-based metaheuristic algorithm, the study in (Chen and Wang 2024) applies Q-learning to determine the ruin strategy for disrupting the current solution, set parameters, and select operators during the variable neighborhood descent (VND) process. Numerical experiments demonstrate the effectiveness of Q-learning in enhancing the algorithm's performance.

On the other hand, ML can guide MH in generating a set of high-quality candidate solutions in the neighborhood, avoiding generating candidate solutions randomly in the neighborhood or all the candidate solutions in the neighborhood range. The jumps generated by Sun et al. (2021) using Dynamic Opposite Learning (DOL) can give MH algorithm a greater chance to jump out of local optima and thus obtain better quality solutions. To automate the antenna design, Fu et al. (2022) uses two ML models, RBF network and simplified Kriging, to screen candidate solutions and generate new solutions to guide the update of the PSO. By replacing the computationally expensive electromagnetic simulation with the ML model, the number of electromagnetic simulations in the particle swarm iteration process is effectively reduced, and the convergence speed of the algorithm is improved. In Wang et al. (2023), QL has a fixed Q-table when it meets the given conditions, which can dynamically determine the way and quantity of individuals to select food sources in the MH algorithm. This approach enables the distributed assembly flow shop scheduling problem to achieve a shorter completion time and total delay time. The study in Truong et al. (2025) proposes an enhanced k-Nearest Neighbor Comparison (k-NNC) model to optimize the computational efficiency of MH algorithms. This model compares the k nearest neighbors of a new solution with those of a selected solution, and directly rejects the new solution if the majority of its neighbors are of lower quality than the compared solution. Doing so avoids the actual computation of the objective function for unpromising candidate solutions.

Multivariate operator

ML-assisted multivariate operators are used in MH to generate new solutions from multiple solutions, which can be realized by recombining multiple solutions or selecting new solutions from multiple candidate solutions.

Recombination is one of the effective ways to generate new solutions in Evolutionary Algorithms (EA). Tinós (2020) assisted EA in calculating the recombination solution through the Radial Basis Function Network (RBFN). The successful reorganization cases that EA has obtained in the implementation process provide knowledge for training RBFN online. Liu et al. (2023) uses RL to assist Genetic Algorithm (GA) in selecting suitable genetic loci. To solve the combinatorial optimization problem, GA interacts with the problem by selecting the genetic locus of the crossover operator, and RL decides to punish or reward the genetic locus based on the execution result of the crossover operator. The crossover operator guided by RL has a greater probability of producing excellent individuals. In addition to EA, other MH methods can also realize solution reorganization through multivariate operators constructed by ML methods. For example, in order to solve the problem of minimum weight independent dominating set, Wang et al. (2020) designed the repair process of the solution based on RL. After obtaining the initial solution, RL assigns a reward value to each node.

After executing the local search, the corresponding reward value is adjusted according to the information obtained from the feedback. Finally, the original solution is destroyed and reconstructed based on RL, which helps the local search algorithm break through the cycle and further search other spaces.

In the process of MH selecting new solutions, the main role of ML is to provide guidance information for selection based on known information. García and Maureira (2021) introduces the concept of KNN into the search operator of the MH algorithm. The modified search operator is used to update the binary solution of the MH algorithm, thus solving the multidimensional knapsack problem. Song et al. (2023) combines a scale-free network with MH algorithm, and MH algorithm embeds the sorted nodes into the network model after each round of search, thus improving the diversity of populations. The scale-free nature of the network can assist the updating mechanism of the MH algorithm in making a trade-off between exploiting the information of excellent individuals and exploring new nodes. Monteiro and Sau (2023) improves the search process of SA and guides the selection of solutions during iteration. This is achieved through the relationship function between state and feedback value constructed by Support Vector Regression (SVR) of the linear kernel and artificial neural network. This function helps to solve the problem of the calculation cost of the fitness function of a given problem being large. RL implements the generalization of the search range from the search space where the node state is known to the search space where the node state is unknown. To obtain the most useful set of new solutions for the EA algorithm, Hu et al. (2023) utilizes DRL to evaluate the potential future rewards of each population based on the current evolutionary state and selects the most suitable population and the corresponding archive as parents to generate offspring.

3.2.3 Selection of search operators

There are various design schemes for search operators, and operator selection plays a crucial role in the efficiency of MH in solving problems. ML can assist MH in selecting the most suitable search operator.

An analysis of the included literature reveals that most existing studies use online learning methods such as RL to select search operators. O Costa et al. (2020) proposes a strategy gradient algorithm and uses it to train deep reinforcement learning. Deep reinforcement learning learns a stochastic policy that can obtain promising new solutions by using the current most frequently visited solutions and historical information of the search process. By doing so, the algorithm can select an appropriate 2-opt operation given the current solution. Tapia et al. (2021) uses QL to select the discretization scheme for two MH algorithms, that is, which transfer function to use and which binarization technique to use in each iteration. The improved algorithm proposed by Durgut et al. (2021) applies QL to the standard MH algorithm, thus enabling the algorithm to dynamically select appropriate operators from a set of operator pools to generate new candidate solutions. With the help of feedback from each operation, the improved algorithm maps the state of the target to the operator. This way, the search operator can learn how to select the best operator online according to the given state. The improved algorithm combined QL with clustering to estimate the distance between the input data and the fine-tuned cluster centers. It then uses it as a reference index when selecting operators.

Zhou and Zhao (2023) used the deep QL network to select the appropriate neighborhood search operator for MH algorithm to expand the search range reasonably. Zhao et al. (2023) adopts inverse reinforcement learning (IRL) with QL mechanism to improve the MH algorithm. The improved MH algorithm builds a policy pool that stores a variety of operators. The QL mechanism selects the appropriate operator through the relevant historical data of each operator in the strategy pool, and the selected four operators are used to update the positions of individuals. The reward function of each operator is learned by IRL from samples provided by experts. To solve large-scale real-valued optimization problems, Zhao et al. (2023) uses a QL-based multi-agent central controller to guide the search of the MH algorithm. The multi-agent central controller established the optimal policy pool through the QL mechanism: in the training phase, the multi-agent central controller trained the agents; in the testing phase, the multi-agent central controller assigns policies from the optimal policy pool to each agent. Yi et al. (2023) proposes two RL-based methods, namely, a deep q-network-based method and a proximal policy optimization-based method, for formulating a range of different metaheuristics in a unified way. The general MH framework uses general algorithm components to automatically select appropriate heuristic and evolutionary operators. Ni et al. (2024) uses QL to complete the policy selection of the MH algorithm. The QL can indirectly select the most suitable policy for the current population through the probability value. Li et al. (2023) uses the combination of AdaBoost and decision tree and BP neural network to realize the adaptive adjustment of the mutation operator and parameters of the DE algorithm. The study in (Shao et al. 2024) designs several problem-specific search operators for exploring and exploiting the solution space, and dynamically selects these operators through a meta-learning-based multi-objective search framework. This framework consists of two phases: the meta-training phase trains a Q-learning model using the search operators, while the adaptation search phase automatically selects the optimal search operators via the trained model to enhance the quality of solutions. The study in Wang et al. (2024) proposes a Q-learning-based MH algorithm for solving mixed integer programming (MIP) problems. The algorithm dynamically selects metaheuristics from three MH algorithms during the iterative process using Q-learning, thereby optimizing the search strategy and improving the solution process. The study in (Yüksel et al. 2024) applies QL to both the Iterated Greedy (IG) algorithm and the Block Insertion Heuristic (BIH) algorithm, enabling dynamic selection of the most suitable perturbation operators, destruction size (dS), and block size (bS), thereby enhancing the algorithms' exploration capabilities in the solution space.

3.2.4 Parameter tuning

Many parameters need to be set in MH, and the rationality of these parameters is related to the performance of MH. Traditional MH requires manual adjustment of parameters based on expert experience or empirical rules to determine the most appropriate parameter settings. However, manual parameter tuning is usually inefficient. ML provides a powerful tool for automating the parameter-tuning process. ML-assisted parameter tuning can be divided into two categories: offline or static parameter tuning and online or dynamic parameter tuning. Offline or static parameter tuning determines the most appropriate parameter settings before the MH algorithm is executed. Online or dynamic parameter tuning dynamically adjusts parameter settings based on feedback information during the execution of the MH.

Offline parameter tuning

The optimal parameter settings of the MH algorithms are closely related to the given problem. Most of the existing works use supervised learning to predict the optimal parameter settings for a given problem. The parameter optimization process of supervised learning applied to MH is implemented in two steps. First, the ML model is trained. The ML model builds its training set by solving a given problem with multiple initial parameter values. ML can learn the potential relationship according to the algorithm performance fed back by different parameter values, thus establishing the relationship model between parameter values and algorithm performance. Second, the trained ML model can guide other given problems efficiently searching parameter settings.

Many ML models have been applied to the parameter optimization process of different MH. Harrison et al. (2019) uses ML techniques to predict the performance of different control parameter configurations of PSO on arbitrary benchmark problems. The modified PSO selects control parameter configurations that predict good performance on the current benchmark problem. Wang and Schafer (2020) uses DT as a prediction model to generate hyperparameters for the three metaheuristics. Ma et al. (2022) uses an Extreme Learning Machine (ELM) to calculate the output weight of the improved MH in all iterations and then estimates the output value. After completing the training of ELM, the improved MH minimizes the error by determining the fitness function. McDevitt et al. (2023) uses the trained classifier to determine the parameter configuration with the best predictive performance for the PSO. The selected classifiers include KNN, tree ensemble classifier, and SVM trained in MATLAB by the functions fitcknn (KNN), fitcsemble (ENS), and fitcecoc (ECO), respectively. Yin et al. (2023) designed a parameter adaptation method based on RL to improve the convergence of PSO. The proposed method controls the parameters of PSO through an Actor-Critic Algorithm. In each iteration, the optimal parameters are generated by the actor-network. Before the algorithm runs, the actor-network completes its training with either the target function or the test function. Han et al. (2023) utilizes Q-learning to design the reference point activation mechanism and reference point adaptation method for the newly proposed multi-strategy and multi-crossover DE optimizer. The study in (Tunga et al. 2024) employs a MH algorithm to solve combinatorial optimization problems (COPs) for non-dominated solutions, aiming to obtain the Pareto front in the solution space. A RL approach enhanced by a Bayesian inference-based learning automaton (BILA) is used to determine the optimal parameters for the MH algorithm. The BILA learns the optimal action policy based on the actions executed by the agent and their corresponding feedback.

Online parameter tuning

The online learning model can extract and learn information during the search process while adaptively adjusting the parameter values. At present, the ML model used for online learning is mainly RL. Fallahi et al. (2022) has designed a parameter-adaptive algorithm based on QL to assist DE and PSO in selecting the optimal parameters in each iteration. Li et al. (2023) proposed a velocity vector generation strategy based on RL. The velocity update model constructed by it provides guidance information for PSO to select particles in each iteration. Yu and Zhou (2023) uses Q-learning to adjust the key parameters and mutation operators in the DE algorithm and applies the adjusted key parameters to the mutation and crossover stages.

3.3 Automation design-oriented ML-assisted MH

Automation design-oriented ML-assisted MH uses ML to generate, design, or optimize MH algorithms automatically. This category of approaches aims to reduce the reliance on human intervention through data-driven approaches, enabling automated development, innovative design, and even adaptive improvement of MH algorithms. This direction emphasizes generating or optimizing the algorithm framework, operation strategy, and core components of MH algorithms to solve different problems more efficiently by learning from existing algorithms, historical optimization experience, and problem characteristics through ML.

As an advanced strategy or framework, metaheuristics are used to guide algorithms on how to efficiently search the solution space to find an approximately optimal solution to the problem. The selection of metaheuristics utilizes ML to automatically select the most suitable combination of algorithms for solving a particular problem from an existing library of metaheuristics. The focus is quickly matching problem requirements with algorithm capabilities by analyzing problem characteristics and algorithm performance history. The generation of metaheuristics refers to generating a new algorithm framework or improving the existing algorithm structure based on the existing MH by adjusting low-level search operators, operator combinations, or parameter configurations. This process focuses on automating the design of new search strategies or operators to improve optimization capabilities. ERL combines the RL and evolutionary mechanism. By learning the dynamic feedback signal during the operation of the algorithm, the evolutionary process of algorithms is optimized, or the behavior of algorithms is dynamically adjusted. The core goal is to make the MH algorithm flexible to adapt to the needs of different optimization stages during operation through dynamic optimization. In recent years, ERL has gradually become an important research direction in RL. Through the combination of the three, ML-assisted MH for automation design can realize the automatic design of the whole process from selection to generation and then to optimization, providing efficient and intelligent solutions for complex optimization problems.

3.3.1 Selection of metaheuristics

Even for the same MH algorithm, different metaheuristics will make the MH algorithm produce different performance. Determining suitable metaheuristics is an important part of designing efficient MH methods for a given problem. Many studies have been proposed on metaheuristic selection using ML methods. Both offline and online learning methods can be used to select metaheuristics.

Offline learning

Offline learning learns relevant information from historical records and applies it to solve other problems. To realize the selection of metaheuristics, the method based on offline learning trains and builds an ML model before constructing the MH method. The model then predicts the performance of each metaheuristic, and the one with the best-predicted performance is used to construct the MH. For example, in Chen et al. (2023), a deep neural network (DNN) is used to guide the selection of branches at each step of the tree search. In each iteration, an initial solution is updated by a set of perturbation and repair heuristics. In addressing the 0-1 Multidemand Multidimensional Knapsack Problem (MDMKP), the study in Scherer et al. (2024) investigates the relationship between the performance of MH algorithms and the problem structure. The research constructs an interpretable DT model to select the optimal algorithm from four MH alternatives for a given instance based on the meta-features of the problem. Experimental results demonstrate that the problem structure significantly influences the performance of MH algorithms.

Online learning

Online learning can be learned during the execution of the MH method to achieve the adaptation of the algorithm, but it usually incurs additional overhead. The current online learning method commonly used for metaheuristic selection is RL. To solve the multi-objective optimization problem (MOPs), Li et al. (2023) has improved the population-based Multi-objective Optimization Evolutionary Algorithm (MOEA). RL is used for the iterative update mechanism. As a heuristic for selecting heuristics, the hyperheuristic based on learning automata can guide the switch between low-level heuristics according to the advantages of different heuristics in different stages of the search, thus improving the versatility of the algorithm. Two RL-based methods designed by Yi et al. (2023) provide guidance information for automatically selecting heuristics and designing new general swarm intelligence algorithms. Zhao et al. (2023) adopts value-based QL as an advanced hyper-heuristic approach, gaining experience from the historical record to select the appropriate low-level MH. Each low-level MH targets a specific problem domain. The study in Lin et al. (2024) employs QL to enhance three MH algorithms, including GA, PSO, and DE, for solving multi-objective scheduling problems. The research designs six problem-specific local search operators (LSOs) for use during the iterative process of the MH algorithms. The QL-based strategy is utilized to select the appropriate MH algorithm for generating high-quality new solutions and to choose the optimal LSO from the six options to accelerate convergence to the optimal solution.

3.3.2 Generation of metaheuristics

In some specific problems, MH methods that automatically generate metaheuristics according to the specified problem can perform better than MH methods that manually construct metaheuristics. In addition, the automated design process can save manpower and improve efficiency. Various ML models have generated metaheuristics suitable for a given problem. For example, Akyol et al. (2023) uses SVM and KNN to calculate the fitness values of two improved MH algorithms.

3.3.3 Evolutionary reinforcement learning (ERL)

ERL combines the advantages of EA and RL and aims to optimize policies or models in RL through an evolutionary process. By simulating processes such as natural selection, crossover, and mutation, EA can search globally in a large policy space, while RL continuously optimizes the policy by interacting with the environment.

The core idea of ERL is to combine the global optimization ability of EA and the local policy update of RL to accelerate agents' learning processes in complex tasks and improve their performance and adaptability. Specifically, EA in ERL is often used to optimize the policy, network structure, or hyperparameters of an RL agent, while RL adjusts the agent's behavior based on feedback from the environment. An important feature of ERL is that EA can enhance the diversity of policies and help solve local optima in RL, thus improving the agents' performance in dynamic and complex environments.

As an emerging research field, ERL has gained a lot of attention and advanced significantly in recent years. Majumdar et al. (2020) has designed multi-agent ERL, which combines gradient-based and gradient-free optimization to handle two agent-specific and team-specific objectives, respectively. A gradient-free optimizer is an EA that maximizes the team objective via neuroevolution. Gradient-based optimizers train policies to maximize dense local rewards for each agent. Liu et al. (2020) proposed an ERL-based multi-agent path planning method. The decentralized evolution method based on the A2C algorithm can be applied to any number of agent training processes, gradually eliminating low-performance policies during the training process to improve training efficiency and performance. To address the scalability problem of GAs in ERL, Bodnar et al. (2020) uses learning-based mutation operators to compensate for the simplicity of genetic representation. The proposed two backpropagation-based genetic operators are integrated into the new framework, which uses an interactive hierarchy between evolution and learning. To address the limitations on individual exploration capabilities when existing approaches treat actor networks as individuals of EA, Zhu et al. (2024) maintains a population of complete RL agents and divides the whole learning process into two distinct phases. In the initial phase, individuals independently learn the actor-critic network, which is alternately optimized by RL and PSO. In the subsequent stages, the best individual performs further optimization, and the remaining individuals continue with PSO-based optimization.

Although ERL faces challenges in terms of computational overhead, convergence speed, and algorithm complexity, it is expected to be more widely used in multi-agent systems, high-dimensional tasks, and automated tasks with the application of algorithm optimization, hardware acceleration, and emerging technologies. Future research will pay more attention to improving search efficiency, reducing computational overhead, and enhancing generalization ability, making ERL an important technique in RL.

3.4 Summary and discussion

As a major class of methods combining MH and ML, ML-assisted MH has unique advantages and limitations. Its advantage is that ML can intelligently adjust the parameters and strategies of MH according to historical data and information in the search process, thus improving the search efficiency and accuracy of the algorithm. As the problem size and amount of data grow, ML can continuously learn and optimize the search strategy, making MH better adapted to the complex and dynamic environment. Furthermore, ML models can easily integrate new data and knowledge, enabling algorithms to handle more diverse problems. The literature in this section proves that ML can reduce unnecessary searches and improve the search efficiency of MH algorithms, thus accelerating the convergence process of algorithms and improving the quality of solutions. Additionally, ML can assist MH algorithms in adaptively adjusting parameters and strategies, enhancing their generalization capabilities.

However, such methods also have some limitations. First, the characteristics of the original MH algorithms may change. When ML is used to optimize the parameters or execution process of MH, some characteristics of the original MH algorithms may be inadvertently changed. These characteristics may be the key to the excellent performance of original MH algorithms on specific problems. For example, the mutation and crossover operations of GA are an important source of diversity in its search. If ML makes inappropriate adjustments to these operations, it may reduce the searchability of MH. Second, the stability of the MH algorithms may fluctuate. MH may have some instability, and its performance may vary for different instances of the problem. This instability may be further exacerbated when ML models are introduced for optimization. In addition, the difficulty of implementing hybrid algorithms may increase. The implementation of ML-assisted MH can be more complex than MH or ML alone. This includes algorithm integration, parameter adjustment, and code optimization. The complexity of implementation may increase the difficulty of algorithm development and maintenance, thereby limiting its promotion and popularization in practical applications.

Figure 5 shows the distribution characteristics of the specific methods adopted by the included literature regarding the categories to which they belong. In this paper, the methods of ML-assisted MH are classified into three categories: problem-oriented ML-assisted MH, performance optimization-oriented ML-assisted MH, and automation design-oriented ML-assisted MH. These three methods show obvious differences in the number of literature and their proportion.

Performance optimization-oriented ML-assisted MH has the largest number of literature, indicating that the current research focuses more on designing the internal mechanism of MH algorithms. This may be because research on internal mechanisms is more fundamental, easier to implement and apply, or has broader applications in data processing, model optimization, and other areas. In addition, performance optimization-oriented ML-assisted MH can provide more flexibility and control. However, the literature on automation design-oriented ML-assisted MH is relatively the least, which indicates that the research in this direction is still in the initial stage or challenging, and this kind of method is still in a relatively marginal position in the current research. This also means that automation design-oriented combination methods have greater research space and development poten-

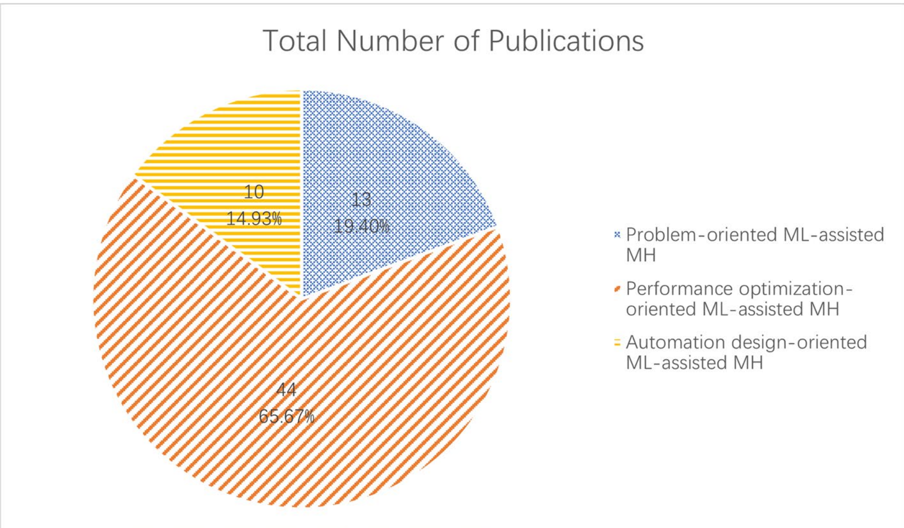

Fig. 5 Category analysis of the methods used in the included literature on ML-assisted MH

tial in the future. For problem-oriented ML-assisted MH, although the number of literature is relatively small, it still occupies a significant proportion. This may be because the problem-oriented ML method pays more attention to practical applications and solving specific problems, so it may have more advantages in some specific fields or applications. This class of methods is of certain importance in the methods of ML-assisted MH and has a certain potential for development in future research.

Judging from the number distribution of literature in the three methods, there is a certain imbalance in the research. The amount of literature using performance optimization-oriented ML-assisted MH is significantly higher than that of the other two types of methods. This could be due to the uneven distribution of research resources or the difference in research difficulty.

4 Metaheuristics-assisted machine learning (MH-assisted ML)

By combining with ML methods, MH improves ML to form a more effective solution to a given problem. According to the different combination methods of MH and ML, the hybrid algorithms of MH-assisted ML can be divided into integrated optimization MH-assisted ML and serial cooperative MH-assisted ML. Integrated optimization MH-assisted ML uses different technologies to collaborate with each other to solve problems. Serial cooperative MH-assisted ML uses different technologies one by one. The input of each technology is the output of the previous technology. The classification hierarchy of MH-assisted ML is shown in Fig. 6.

The two types of hybrid algorithms mainly differ in their working methods and degree of interaction. In terms of working mode, since each algorithm may be good at different aspects, integrated optimization MH-assisted ML makes full use of their respective advantages through cooperation to improve the overall performance. Serial cooperative MH-assisted ML makes each algorithm responsible for optimizing different aspects of the model or providing solutions at different stages, forming a relaying process. In terms of the degree of interaction, the algorithms of integrated optimization MH-assisted ML have closer interaction with each other and may need to run simultaneously and communicate with each other to work collaboratively. The interaction between algorithms of serial cooperative MH-assisted ML is relatively simple and is usually achieved by passing solutions or intermediate results. Each algorithm usually only focuses on the task of its specific stage. Both types of hybrid algorithms utilize MH to improve the performance and efficiency of ML models. They have certain flexibility and can be combined to form a more comprehensive and efficient optimization strategy in some cases.

The classification scheme presented in this paper categorizes MH-assisted ML into two distinct types, which is designed to eliminate redundancy and reduce the complexity of the taxonomic system through a clear classification rationale and comprehensive scope. The integrated optimization MH-assisted ML achieves a synergy between global and local optimization by leveraging multiple techniques concurrently, making it suitable for high-dimensional and nonlinear problems. In contrast, the serial cooperative MH-assisted ML handles multi-stage optimization tasks through a chained transfer of techniques, fitting for scenarios that require sequential processing. Compared to classification frameworks based on task types or technological hierarchies, the dual-type framework proposed in this paper more

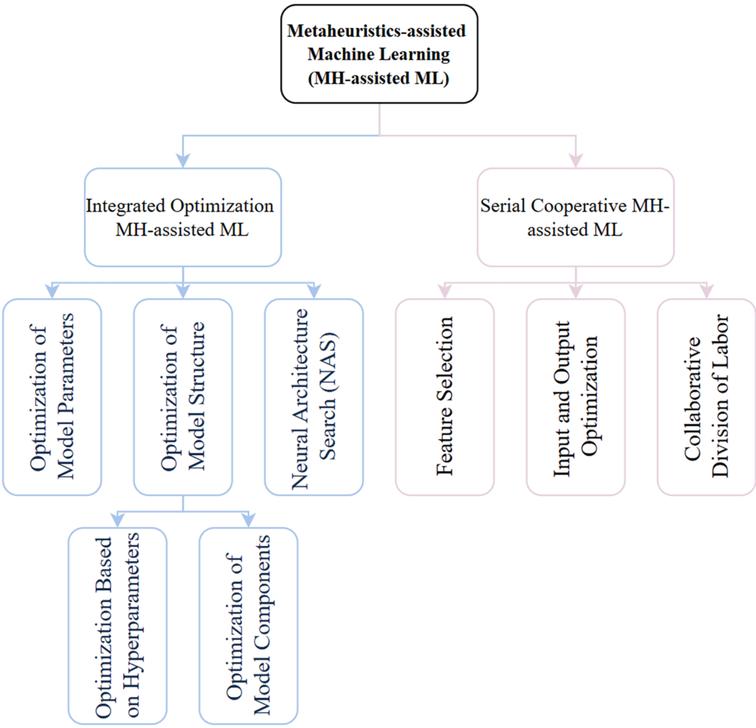

Fig. 6 Classification of MH-assisted ML

directly reflects the intrinsic nature of the integration between MH and ML. It combines logical rigor with practical applicability, facilitates algorithm design and optimization, and offers efficient and adaptable methodological support for solving complex system problems.

4.1 Integrated optimization MH-assisted ML

Through the cooperation of two or more algorithms in the execution process, integrated optimization MH-assisted ML integrates the respective advantages of different algorithms to improve its ability to solve a given problem. MH is usually used to optimize the parameters and structure of ML models.

Moreover, NAS has significant advantages as an important tool for automating the design of ML models. NAS can automatically discover the network architecture suitable for a specific task, reducing the time and experience dependence of manual design. NAS is developing rapidly and shows great potential in improving model accuracy, adapting to multitasking, and hardware friendliness.

4.1.1 Optimization of model parameters

MH methods can adjust the weight, bias, or loss function of ML models to get the best performance. MH methods can optimize the model parameters to achieve the selection of optimal parameters, initialization of parameters, and treatment of local minimum and overfitting.

Selection of optimal parameters

Most works use the MH methods in the training process of ML so as to obtain the optimal parameter configurations of ML models. There have been reviews on obtaining the optimal parameter configuration of ML models by MH methods, and the related research works have been summarized. For example, Eshtay et al. (2019) reviews the optimization of ELM networks by MH methods from three directions: the optimization of input weights and hidden biases, the selection of hidden neurons, and the optimization of activation functions.

In most related studies, researchers use an MH to search for the optimal configuration of the weights and biases of the neural network. Moayedi et al. (2020) predicts soil shear strength by using the Multilayer Perceptron (MLP) with the Tangent Sigmoid Function (Tansig). The connection weights between existing layers and the threshold value assigned to each node are optimized by four MH methods. Yuan and Moayedi (2020) determines the optimal values of the weights and biases of the MLP neural network through six evolutionary methods to achieve the prediction of landslide incidence. Ng et al. (2021) utilizes the Stochastic Gradient Descent (SGD) algorithm, Adaptive Moment Estimation (Adam) algorithm, and PSO to train a neural network model as an intelligent agent. Zhou et al. (2021) determines the most suitable weights and biases of the Multilayer Perceptron Neural Network (MLPNN) through three MH algorithms. Zhang et al. (2021) optimized the weights and biases of the ELM model through GA and PSO, reducing the error of the ELM model in predicting copper prices. Wu et al. (2022) optimizes the weights and biases of the MLP by two MH algorithms. Kardani et al. (2022) used the improved MH to optimize the bias and weights of ELM and ANN, respectively, so as to predict the petrophysical properties of carbonate reservoirs. Si et al. (2022) uses MH algorithm to search the weights and biases of the MLP so that the Minimum Mean Squared Error (MSE) of medical data classification is minimized. Chen et al. (2023) uses the MH algorithm to optimize the initial weights and biases used by CNN in the initial training period and realizes the pre-training of the CNN model. For the strength prediction of carbon fiber reinforced plastics (CFRP), Kaveh and Khavaninzadeh (2023) compares the different effects of four MH algorithms when they are applied to the training functions of Feed-Forward Backpropagation (FFB) and Radial Basis Function (RBF). The MH methods optimize network weights and biases for different layers. Liu et al. (2023) introduces evolutionary theory into the graph structure learning (GSL) method. Through EA's optimization of GNN model parameters, the improved GSL can adapt to environmental changes. Li et al. (2023) optimizes the structure of the BP neural network by using a Bayesian algorithm to determine the number of neurons in a single hidden layer before training the prediction model. After the structure optimization, the model parameters are optimized by GA, and the BP neural network with more accurate prediction is obtained. The study in Sangjinda et al. (2024) integrates ANNs with four MH algorithms to construct hybrid predictive models. The MH algorithms are employed to optimize the weights and bias parameters of the ANNs, thereby enhancing the predictive accuracy and computational efficiency of the models.

It is worth noting that there are still a few studies that focus on using a certain kind of MH to adjust only the weights of neural networks. These studies focus on exploring the specific impact of weight optimization on neural network performance, aiming to develop more efficient and targeted optimization strategies to adapt to various application requirements and data characteristics. Altan et al. (2019) constructs a digital currency prediction model that can reflect the nonlinear characteristics of time series. The Long Short-Term Memory (LSTM) neural network identifies each Intrinsic Mode Function (IMF) obtained by the Empirical Wavelet Transform (EWT) decomposition technique, and the MH algorithm is used to optimize the weight coefficients of each output. Behjat et al. (2020) employs a scalable backpropagation method and PSO for handling multimodal loss functions to train neural network weights to predict the behavior of the physical system. El-Kenawy et al. (2021) uses improved MH algorithm to optimize the connection weights of the MLP neural network. This resulted in the highest accuracy in the classification of the input chest X-ray case images. Panda and Majhi (2021) uses a MH algorithm to train a kind of feedforward neural network, namely PSNN neural network. PSNN can be used to solve data classification problems. The MH algorithm provides random weights to the PSNN and selects the parameter values that reduce the RMSE value fed back from the PSNN. After sufficient iterations, the MH algorithm can produce the best parameter configuration for PSNN to achieve the best classification accuracy. The study in Jiadong et al. (2024) utilizes five MH algorithms to fine-tune the weights of ANNs, aiming to enhance the predictive accuracy of ANN-based models.

In the field of ML, in addition to neural networks, many other models also make extensive use of a certain MH to optimize their key parameters. To solve the non-convex optimization problem of base station (BSs) deployment, Dai and Zhang (2020) uses multi-objective GA and greedy algorithms to optimize the parameters of different ML technologies, including KNN, RF, SVM with different kernels, and MLP. Xie et al. (2021) uses MH algorithm to optimize the key parameters of Gradient Boosting Machine (GBM), SVM, Gaussian Process (GP), and ANN, respectively. Petrović et al. (2022) uses MH algorithm to optimize the AdaBoost algorithm to solve the task of credit card fraud detection. To improve classification performance in an industrial environment, Neelakandan et al. (2022) uses MH algorithm to optimize the parameters of the Stacked Sparse AutoEncoders (SSAE) model for classifying garbage. Yonar and Yonar (2023) determines the premise parameters and result parameters of the Adaptive Neuro-Fuzzy Inference System (ANFIS) model by GA, PSO, and DE and then realizes the prediction of air pollution. The study in He et al. (2024) constructs a self-organizing neural network (SONN) for prediction, which consists of two layers of weights: the first layer of weights, between the input layer and the hidden layer, is determined using an evolutionary computation method; the second layer of weights, between the hidden layer and the output layer, is optimized by three MH algorithms, respectively. Experimental results demonstrate that the SONN optimized by the three MH algorithms significantly improves prediction accuracy.

In recent years, researchers have actively explored the integration of multiple traditional MH into hybrid optimization algorithms to optimize the parameters of ML models. This method aims to combine the advantages of each algorithm and shows a broad application prospect in the field of ML, which provides a new way to improve the performance of the model. Wu et al. (2020) uses the coupled ELM model to predict pan evaporation and obtains the optimal parameter configuration of ELM by two MH algorithms. Combining GWO and PSO, El-Hasnony et al. (2020) optimizes both the result and prerequisite parameters of ANFIS. Ansari et al. (2020) combines the Magnetic Optimization Algorithm (MOA) and PSO to train the Feedforward Neural Network (FFNN) and reduces the time complexity of training the weights of FFNN in bankruptcy prediction. Shariati et al. (2021) defines two sets of parameters for ANFIS, namely result parameters and premise parameters. Its optimization process is completed by PSO and GA. Ahmadianfar et al. (2022) uses the ANFIS model to predict electrical conductivity (EC). The proposed adaptive hybrid algorithm combines DE with PSO to determine the linear relationship coefficients between the optimal Membership Functions (MFs) and IF-THEN Fuzzy Rules (FRs) for the input parameters of the ANFIS model. The study in (Moosavi et al. 2024) integrates the Transductive Transfer Learning (TTL) mechanism with DNNs to construct a predictive model. A hybrid MH algorithm, formed by combining two MH algorithms, is employed to optimize the weights and biases of the DNN, thereby enhancing the accuracy and convergence speed of the hybrid model.

Initialization of parameters

The MH methods can provide suitable initial values for the model parameters and optimize the training process of the ML model. Jiang et al. (2019) uses PSO to improve MH algorithm and then adjusts the initial weight and threshold of ELM through the improved MH algorithm.

Treatment of local minima and overfitting

The problems of local minima and overfitting are common to ML methods and have a significant impact on the final performance of the model. Embedding MH into ML models provides an effective solution to the problems of local minima and overfitting. Various modified MH methods have been used in existing works to deal with local minima and overfitting problems. In terms of local minimum, Tran-Ngoc et al. (2021) combines two MH algorithms to form a hybrid MH algorithm to solve the local minimum problem of ANN. When the parameters of the ANN fall into a local minimum, the hybrid MH algorithm transforms the training parameters into a single vector and optimizes it as the initial population of the GA. In terms of overfitting, Tavasoli et al. (2021) uses an improved MH algorithm to optimize the parameters of SVM whose kernel function is RBF for solving the optimization problem during gene selection. The improved MH algorithm can adjust the parameters of the RBF, thus preventing overfitting with the least number of features.

4.1.2 Optimization of model structure

For the same ML model, the difference in model structure can have a crucial impact on the final performance of the model. The performance of ML models can be improved by using MH methods to optimize the structure of ML models.

4.1.2.1 Optimization of hyperparameters

The hyperparameters of an ML model are the variables used to tune its structure. In the existing research works, MH methods mainly optimize the hyperparameters related to the model structures to improve the ML models.

(1) Optimization using a single MH algorithm

In recent years, the research on optimizing neural network hyperparameters has dominated the research on hyperparameter tuning in the ML field. The research work of optimizing neural network hyperparameters can be divided into two categories. One is to use a single

MH to optimize hyperparameters. The other category applies multiple MHs to the hyperparameter optimization process separately. These two types of methods, respectively, focus on the performance optimization of a single algorithm and the performance comparison between multiple algorithms, which provide different strategies and perspectives for the effective adjustment of neural network hyperparameters.

On the one hand, literature on optimizing neural network hyperparameters using a single MH usually conducts in-depth research on specific MH algorithms and verifies their effectiveness in optimizing neural network hyperparameters through rigorous experimental design and performance evaluation. The learning rate is one of the important hyperparameters that affect the performance of the DNN model. Khan et al. (2021) used the MH algorithm to optimize the learning rate of DNN. Rajesh and Kumar (2022) uses DE to determine the most appropriate network structure parameters and hyperparameters for deep CNN. Eid et al. (2022) used LSTM to predict confirmed cases of monkeypox and optimized its network parameters by the MH algorithm. Pandey et al. (2022) uses the ANN optimized by GA for classification prediction of heterogeneous reservoirs and estimation of related reservoir parameters. Reservoir model identification is a classification problem, while reservoir model representation is a regression-based task. By selecting features and optimizing the number of neurons in the first three hidden layers of the classifier and regressor, GA improves the prediction accuracy of the model. Mamoudan et al. (2023) combines CNN with Bidirectional Gated Recurrent Unit (BiGRU) and determines its optimal hyperparameters through FA to improve the performance of the neural network. Zhang et al. (2023) proposed a cost-sensitive DL method based on attention TCN. The class costs and hyperparameters of the network are optimized by an adaptive top-k differential evolution (ATDE) algorithm.

On the other hand, the literature that applies various MH to the hyperparameter optimization process of neural networks usually systematically discusses the performance of different MH in the optimization of neural network hyperparameters. Through comparative experiments and performance evaluation, the advantages and disadvantages of various algorithms are analyzed in-depth. This kind of literature pays more attention to the comprehensive performance evaluation of the algorithm, such as optimization effect, efficiency, stability, and interpretability. It provides researchers with in-depth insights on how to choose and use MH to achieve the best optimization effect through rigorous experimental design and data analysis. Passos and Papa (2020) uses different MH methods to adjust the main hyperparameters of Deep Boltzmann Machines (DBMs), namely the number of hidden units, learning rate, weight decay, and momentum. Gaspar et al. (2021) uses MH to optimize the hyperparameters of CNN. When optimizing CNN hyperparameters, firstly, MH is used to determine the number of hyperparameters that need to be optimized, and the MH algorithm is initialized. Then, the images that need to be fed into the CNN are normalized. Next, the CNN is encapsulated in a function to be called when evaluating the objective function. Kumar and Singh (2021) constructs a three-layer neural network to predict the workload in the cloud environment and optimizes its input neurons by swarm intelligence algorithm, EA, nature-inspired algorithm, and physically-inspired algorithm. Nayak et al. (2022) uses three MH algorithms to determine the optimal input parameters of the CNN model to minimize the fitness function value. These input parameters include training size, learning rate, and number of epochs. Using two MH algorithms, Hakim et al. (2022) fine-tunes the hyperparameters of the CNN model to improve the accuracy of the model prediction. Zhao et al. (2022) uses ANN to establish and analyze the nonlinear relationship between compressive strength (CS) and concrete composition. Firstly, nine common algorithms are used to train the neural network, and the algorithm with the best performance is selected as the training algorithm of CNN. Then, two MH algorithms are used to optimize network parameters of CNN to overcome its computational shortcomings. Khan et al. (2022) applies various MH methods to select the appropriate number of hidden layers and associated neurons for ANN. Chen and Song (2023) uses the CNN model for spatial explicit prediction of landslide susceptibility and adjusts the hyperparameters of the CNN model through two MH algorithms to improve the prediction accuracy of the model. The study in Xu et al. (2024) integrates RF with six MH algorithms to construct hybrid predictive models. Experimental results demonstrate that optimizing the hyperparameters of RF using MH algorithms significantly enhances both the predictive accuracy and computational efficiency of the models.

In addition to focusing on optimizing hyperparameters of neural networks, there exists a large literature devoted to optimizing hyperparameters of other ML models. These studies cover a wide range of model types, including SVM, DT, RF, etc. They explore how to effectively tune the hyperparameters of different types of ML models to achieve model performance improvement. According to the different research objectives and scope, the research work on hyperparameter optimization of ML models other than neural networks can be divided into two categories. One is to optimize the hyperparameters of a single ML model to maximize its performance on a specific task. The other is to extensively compare and optimize the respective hyperparameters of multiple ML models to identify the model configuration that performs best in different scenarios or datasets. These two types of research aim to improve model performance by tuning model parameters. The former can assist in gaining a deeper understanding of the performance characteristics of a particular model, while the latter can help to make more informed model selections in different application scenarios.

First, for the literature focusing on hyperparameter optimization of a single ML model, the salient feature is to explore the hyperparameter space of a particular model in depth, aiming to find the optimal configuration to improve the model performance. In the literature, researchers usually elaborate on the selection basis, adjustment strategy, and performance evaluation method of hyperparameters and verify the effectiveness of the proposed optimization strategy through comparative experiments. In addition, this kind of literature also focuses on analyzing the relationship between hyperparameters and model performance, providing theoretical support and practical guidance for improving model performance.

Hoang (2019) constructs a Least Squares Support Vector Machine (LSSVM) model to detect asphalt pavement patches. To assist the construction of the LSSVM model, the MH algorithm is used to optimize the hyperparameters of the LSSVM model. Zhou et al. (2021) optimizes the hyperparameters "C" and "gamma" of SVM through three MH algorithms to more accurately predict the tunneling rate of TBM under hard rock conditions. Fadhillah et al. (2021) uses two MH algorithms to determine the optimal hyperparameters of SVR. The optimized SVR is used to establish the probability model for Gangneung-si groundwater. The hyperparameters that need to be set in the SVR model include the kernel function (kernel), the penalty parameter (C), and the loss feature ε(epsilon). Fattahi and Hasanipanah (2021) combines the Relevance Vector Regression (RVR) model with two MH algorithms which can optimize the hyperparameters of RVR. Zhou et al. (2021) combines six MH algorithms with the XGBoost algorithm to optimize the hyperparameters of XGBoost. Tao et al. (2021) uses GA to optimize the hyperparameters of the XGBoost model. Luo and

Paal (2021) uses Coupled Simulated Annealing (CSA) to optimize the hyperparameters that affect the prediction ability of the Least Squares Support Vector Machine Regression (LS-SVMR) model to enhance the generalization ability of the lateral strength prediction of reinforced concrete columns. Ngo et al. (2022) uses MH to optimize the hyperparameters of SVR and finally realizes the energy consumption prediction in the building sector. Paryani et al. (2022) uses two MH algorithms to determine the optimal parameter settings for SVR. He et al. (2023) searches the solution space for the optimal hyperparameters of RF, namely n estimator, maximum depth, and minimum sample split, through three MH algorithms. The optimized RF reduces the prediction generalization error. Li et al. (2023) uses three MH algorithms, respectively, so as to determine the optimal hyperparameter configuration for SVM to improve its ability to predict the stability of hard rock pillars. To detect the severity of traffic accidents, Pérez-Sala et al. (2023) proposed a framework based on 1-D and 2-D CNNs. Before the Boosting algorithm calculates the weight of features, GA optimizes the hyperparameters of the Boosting algorithm. The study in Wu et al. (2024) employs two MH algorithms to optimize the hyperparameters of Gradient Boosting Decision Trees (GBDT), aiming to enhance the predictive accuracy of GBDT-based models.

Secondly, the literature on optimizing and comparing the respective hyperparameters of various ML models is usually comprehensive and systematic, covering not only a variety of different machine learning models but also exploring the optimization strategy of hyperparameters of each model in depth. Zhang et al. (2020) divides the training process of ML into four stages: selection of ML algorithms, determination of optimal hyperparameters, improvement of model robustness and sensitivity analysis, and combines PSO with ML methods to determine its globally optimal hyperparameters. This paper comprehensively compares the prediction performance of Back Propagation Neural Network (BPNN), Generalized Regression Neural Network (GRNN), ELM, SVM, and RF optimized by PSO. To detect spam, Gibson et al. (2020) uses NB, SVM, RF, DT, and MLP to construct a classifier, and the hyperparameters of the given model are provided by PSO. Zhang et al. (2021) compares the performances of BPNN, ELM, SVM, RF, and Evolutionary Polynomial Regression (EPR) in predicting the compressibility Cc of reconstructed clay and uses GA to optimize the superparameters of five ML algorithms. In GA, the average prediction error of the 10-fold cross-validation (CV) set is used as a fitness function, which effectively enhances the robustness of the ML model and avoids the overfitting problem. Cao et al. (2021) uses the LogitBoost Classification and Regression Tree (CART) model to identify image samples. Additionally, MH is used to determine the most appropriate hyperparameter set of the LogitBoost Tree model, including learning rate, learning period, minimum number of leaves, and maximum number of splits. Ng et al. (2022) uses PSO to assist the training process of SVR and Feedforward Neural Network (FNN). PSO represents each weight or bias as a particle, and the number of particle swarms is the number of particle sets used in the training process. By adjusting the hyperparameters of SVR and FNN, PSO minimizes the value of the objective function, thereby improving the predictive ability of the ML model. In order to enhance the ability to predict the heating and cooling loads of residential buildings, Dasi et al. (2024) uses several MH algorithms to optimize two commonly used SVR and XGBoost. Ahmad et al. (2024) employs eight supervised ML models for prediction while using tree-based pipeline optimization tools and GA to optimize the input parameters of ML models. Razavi-Termeh et al. (2023) uses a GA algorithm to select the optimal hyperparameter values of two parallel integrated ML models, RF and bootstrap aggregation

(Bagging), so that these models have the best predictive ability. Han et al. (2024) applies GA to optimize the hyperparameters of XGBoost-based and neural network-based models, respectively. The study in Dakic et al. (2024) employs PSO to optimize the hyperparameters of both KNN and XGBoost classifiers, thereby enhancing the detection accuracy and computational efficiency of the attack detection systems.

(2) Optimization using hybrid MH algorithms

Elmasry et al. (2020) proposed a dual PSO algorithm for automatically selecting hyperparameters for DNN, Long Short-Term Memory Recurrent Neural Networks (LSTM-RNN), and Deep Belief Network (DBN). Zhang et al. (2020) introduces adaptive inertia weight and Levy flight into MH algorithm to realize dynamic adjustment of step length according to fitness value, thus avoiding MH algorithm falling into local optimum. In the process of training the Multi-Output Least Squares Support Vector Regression (MOLSSVR) model on the training dataset, the improved MH algorithm finds the local optimal hyperparameters corresponding to the minimum RMSE for MOLSSVR. Zhang et al. (2020) proposes a multi-objective PSO to search for the optimal hyperparameter of the ML algorithm to optimize the concrete mix ratio. Jovanovic et al. (2022) improves the traditional MH and uses it to optimize the hyperparameters of SVM, ELM, and XGBoost. Kaveh et al. (2023) combines three MH algorithms to optimize the hyperparameters of ANN, thus minimizing the Mean Square Error (MSE) during the training process. Navazi et al. (2023) optimizes the hyperparameters of ML by GA or grid search, thus realizing the prediction of early risk of diabetes by ML with optimal hyperparameters. The ML models used for prediction include DT, KNN, SVM, and MLP. Using PSO for feature selection achieves the optimization of feature dimension, which helps to find the optimal feature subset and avoids the prediction of all features. In order to reduce the control cost and achieve efficient and precise control of robots with complex dynamic characteristics, Zheng et al. (2023) designes a control method based on a fuzzy neural network (FNN). On the one hand, the adaptive differential evolution (ADE) method searches for the optimal parameters in the global scope and delimits the pseudo-global search scope. On the other hand, the memetic differential evolution (MDE) method searches for the optimal parameters in a pseudo-global scope. The probability factor can determine whether to use the backpropagation (BP) algorithm for online optimization.

(3) Optimization of model components

In the optimization of hyperparameters, there are related reviews on the application of MH methods. Eshtay et al. (2019) reviews related works on MH methods for optimizing different components of ELM networks. The parts that need to be optimized include the weights and biases that represent the connections between neurons in the input and hidden layers, the structure of the network, such as the number of neurons, and the values produced by the hidden neurons. Morales-Hernández et al. (2023) reviews the Multi-Objective Hyperparameter Optimization (HPO) methods for ML and classifies the related research using metaheuristics and metamodels into three categories: optimization algorithms based on metaheuristics, optimization algorithms based on metamodels and hybrid algorithms. In order to achieve fault section diagnosis, Xiong et al. (2023) designs a divisional method based on hierarchical extreme learning machines (HELM) with structure adaptation. DE optimizes the number of neurons in the hidden layer and the regularization factor of HELM so as to obtain the submodule with the optimal structure.

4.1.2.2 Optimization of model components

In addition to the hyperparameters of the model, MH methods can also optimize the structure of the ML model by directly adjusting its key components, thereby achieving better model performance. Połap and Woźniak (2021) uses classic ANN and CNN to perform image recognition tasks and trains neural networks through federated learning. After obtaining all the models, federated learning executes the aggregation mechanism to create a new model for the next round of federated learning. The MH algorithm is executed on the server to verify model quality. The best model is selected for further aggregation by analyzing the loss function value of the MH algorithm. Mahjoubi et al. (2021) automatically configures parameters and hyperparameters for ML models via a tree-based pipeline optimization tool (TPOT). To achieve multi-objective optimization, the non-dominated sorting genetic algorithm (NSGA) is used to optimize the objective function. Revin et al. (2023) designs EA-based optimizers for automatically creating and optimizing composite machine-learning pipelines with variable-shape structures. Wang et al. (2024) employs three multi-objective EAs to optimize the objective functions of six ML models for prediction. The DL model proposed by Poyatos et al. (2023) uses TL to initialize and replaces the final fully connected layers with sparse layers optimized by GA. This model can satisfy the EA for pruning or feature selection in this densely connected part of the neural network. Dogan et al. (2023) establishes a deep cascade forward neural network (DCFNN) for prediction, and the objective function value of the model is optimized by GA. Ma et al. (2023) designed the DL algorithm based on the SA framework. The algorithm replaces the state exchange process with a neural network, the energy function with a description function, the energy function with a description function, and the annealing criterion with the gradient optimization method. The structure parameters of the reference images are used to determine the number of network layers, the feature distribution of the reference images extracted by the description function is used as the objective function, and the gradient optimization method is used to optimize the designed network.

4.1.3 Neural architecture search (NAS)

NAS is an automated search technique for neural network architectures that aims to automatically discover the best or near-optimal neural network architecture according to the requirements of a specific task. Traditional neural network design usually relies on human experience and experimentation, but NAS automatically explores the space of network architectures through algorithms to improve model performance and save design time.

NAS can discover architectures that are better than manually designed networks, often improving models' performance. Especially on complex tasks, such as computer vision and natural language processing, NAS can discover innovative architectures that are not easily found by manual design, which provides new ideas to solve difficult problems in specific tasks. NAS can design the most appropriate architecture for different tasks (such as classification, detection, regression, etc.) so that the network can be more accurately adapted to various task requirements.

NAS technology has developed rapidly since 2017 and has made remarkable progress. The combination of MH and ML has become one of the important research directions of NAS. Xue et al. (2023) considers both accuracy and time consumption and proposes a multi-objective genetic algorithm based on probability stack (MOEA-PS). MOEA-PS stacks structural blocks through proxy models to generate deep neural networks. MOEA-PS accelerates the search speed of NAS while ensuring the accuracy of the final network structure. To realize Multi-objective Neural Architecture Search (MONAS), Luong et al. (2024) uses NSGA-II as a search algorithm to simultaneously optimize network accuracy and network complexity. To reduce the computational cost, training-free performance metrics (i.e., synflow, jacov, grasp, snip, grad norm, or fisher) are used as proxies for the network accuracy objective. Lin et al. (2024) proposed an RNN structure search method based on multi-objective evolutionary algorithms. This approach optimizes objectives related to the complexity of RNN architectures by using approximate network morphisms. The multi-objective optimization framework based on the NAS strategy proposed by Pujari et al. (2023) optimizes the hyperparameters in the Nonlinear Auto-Regressive (NAR) model, Wavelet Neural Networks (WNN), and LSTM model and reduces the computational burden by reducing the number of parameters. Han et al. (2024) proposed a self-adaptive differential evolution algorithm for neural architecture search (SaDENAS). SaDENAS integrates local and global vector information in a linear combination to balance exploration and exploitation. The self-adaptive scaling factor dynamically adjusts the balance between local and global search throughout the optimization process.

Although NAS still faces challenges such as high computational overhead and low search efficiency, the combination of MH and ML will play an increasingly important role in NAS research with algorithms' improvement, hardware adaptation, and computing resources. Future research will continue to focus on improving the search efficiency, reducing the computational cost, and improving the generalization ability of the models.

4.2 Serial cooperative MH-assisted ML

Serial cooperative MH-assisted ML uses different algorithms executed separately, one by one, taking the output of each stage as the input of the next stage, thus forming a hierarchy. When ML works serially with MH in a specific order, MH can select appropriate features for ML, optimize the input and output data of ML, or be responsible for optimizing or improving the solution in a specific stage of the whole hybrid algorithm.

4.2.1 Feature selection

Feature selection is a key step in ML. The utilization of various features can lead to different effects produced by the same ML model. In the feature selection stage, various methods have been used to minimize the dimensionality of the feature set while maintaining the performance of ML. Existing surveys have provided a comprehensive summary of feature selection methods based on metaheuristic algorithms. Specifically, the survey in Dokeroglu et al. (2022) offers an in-depth review of the most influential metaheuristic feature selection algorithms over the past two decades, analyzing their performance in key aspects such as exploration-exploitation balance, selection mechanisms, transfer functions, fitness evaluation, and parameter configuration. The study in Veeranjaneyulu et al. (2024) provides a comprehensive review of existing research on the use of swarm intelligence metaheuristic algorithms for optimizing feature subset selection in breast cancer datasets and applying the optimized feature subsets to ANNs. The review compares the effectiveness of different methods in classification tasks using various evaluation metrics, highlighting the potential of swarm intelligence metaheuristics in optimizing ANNs while also addressing their limitations.

MH methods have great potential in solving optimization problems. In recent years, many studies have applied MH methods to the feature selection problem of ML before the ML method is implemented. Sakhnini et al. (2019) uses three MH algorithms to determine the optimal feature subset for the supervised learning algorithm to obtain the best accuracy. The supervised learning methods used include SVM, KNN, and ANN. Sharma and Rani (2019) has constructed a cancer classification framework, thus minimizing the gene subset and improving the classification accuracy. Firstly, a filtering method removes irrelevant genes so that a smaller subset of relevant genes can be selected. Then, the multi-objective two MH algorithms are combined to select the most relevant and informative gene subset from the preprocessed gene subset. The fitness function of the hybrid MH algorithm can identify the fitness of gene subsets. Finally, SVM, KNN, Naive Bayes (NBY), and DT are used to classify and predict the identified optimal gene subsets. Elmasry et al. (2020) improves PSO by selecting an appropriate feature subset before training the ML model. The three DL models used include DNN, Long Short-Term Memory Recurrent Neural Network (LSTM-RNN), and DBN. Pham et al. (2020) combines adaptive PSO with other MH algorithms for feature selection. The selected features are used as the input of the MLP to achieve multi-class classification. To remove redundant or irrelevant features and reduce the dimension of feature space, Mazaheri and Khodadadi (2020) combines GA, PSO, and DE. A non-dominant sorting genetic algorithm (NSGA II) is used to perform multi-objective optimization to optimize the cost function and the number of selected features. Finally, the KNN, feedforward neural network (FF net), fitting neural network (Fit net), radial basis function neural network (RBFNN), and pattern recognition network (Pat net) are used to classify normal heart rhythm and six kinds of abnormal heart functions. Le et al. (2021) combines two MH algorithms to form a hybrid MH method. By reducing the number of required input attributes, it realizes the feature selection of preprocessed data, thus optimizing the structure of MLP and improving the classification accuracy of early diabetic patients. As a survey on solving feature selection problems for ML, Agrawal et al. (2021) summarizes the research between 2009 and 2019 that applied MH methods to solving feature selection problems for ML. MH adjusts the parameter values of ML to reduce the dimension of the feature set while maintaining good performance. Sai Sindhu Theja and Shyam (2021) combines the MH algorithm and Opposition-based Learning (OBL) method to search for the optimal solution in the feature space to select the basic features. Based on the selected features, the RNN classifier classifies the input data, thus realizing the detection of Denial of Service (DoS) attacks. Bacanin et al. (2021) solves the feature selection problem in ML by reducing the dimension of features through MH. Khan et al. (2022) selects the features of the input ANN by various MH methods. Yun et al. (2023) proposed a sequential two-stage best feature subset selection method for stock price prediction models. In the first stage, the wrapper method is combined with GA to select the respective important feature sets of the five tree-structured ML regression algorithms. In the second stage, the filter method uses the importance scores to filter out the more important features common to the five ML regression algorithms from the important features selected in the first stage. Kuo and Chiu (2024) proposes a hybrid algorithm that integrates two distinct MH algorithms, designed to efficiently search for the optimal feature subset of SVM and the optimal values of the two key parameters in the Radial Basis Function (RBF) kernel. The study in Zerouali et al. (2024) proposes an enhanced MH algorithm designed to address the challenges of handling discrete variables. This algorithm is applied to feature selection tasks in deep learning models, including RNNs, LSTMs, and GANs. Experimental results demonstrate that the proposed method significantly improves the accuracy of models in slope stability classification tasks.

To achieve intelligent crowd density classification, the study in (Ahmad 2024) first preprocesses the input images using Guided filtering (GF), then extracts image features using a neural architectural search network (NASNet), where the hyperparameters of NASNet are optimized by an improved MH algorithm. Finally, a multilayer extreme learning machine (MLELM) is employed to perform the crowd density classification task. The study in Dokeroglu and Kucukyilmaz (2025) employs a binary multi-objective MH algorithm to optimize feature selection and applies the resulting optimal feature subset to an adaptive KNN classifier. Experimental results demonstrate that the enhanced KNN classifier significantly reduces computational costs by decreasing the number of features while maintaining high classification performance. The study in Pandey and Pandey (2025) employs MH algorithms to select optimal feature subsets from extracted image features and inputs these features into five deep transfer learning classification models. Experimental results demonstrate that the use of MH algorithms for feature selection enhances both the classification speed and accuracy of the deep transfer learning framework.

4.2.2 Input and output optimization

MH methods have a wide range of applications in solving optimization problems, so performing MH methods before using the ML model can optimize the data input to the ML model. To enhance the ability of the Intrusion Detection System (IDS) to resist new attacks, Msika et al. (2019) iteratively generates antagonistic samples and uses them to train four ML methods that constitute IDS until the detection rate converges. The ML methods constituting the IDS include a 3-layer NN, RF, SVM, and naive Bayes classifier. Each iteration round consists of three phases, namely the generation phase, the evaluation phase, and the training phase. In the generation stage, the GAN is combined with the local search algorithm and GA to generate a new attack dataset. In the evaluation stage, the detection rate of newly generated attacks is calculated. In the training phase, the newly generated examples train the IDS to better cope with these attacks. Regarding the optimization of training data, Boubekeur et al. (2020) has designed an evaluation method to help teachers evaluate student submissions in model-driven engineering courses. Firstly, the literature designs a scoring heuristic to calculate the number of related concepts and generates a score for each concept, thus evaluating students' submissions. Then, the output data of scoring heuristics are used for the training of logistic regression (LR), NB, DT, RF, and KNN to realize the recognition of high-quality submissions and predict their approximate letter grades. In addition to optimizing the input data, MH methods can further adjust the output results of ML models to better solve the given problem. Sadrossadat et al. (2022) first builds models using ANNs and Gaussian Process Regression (GPR) and selects the model with the best performance from them. Then, PSO completed the design and optimization of multi-objectives based on the prediction results of the selected model. The study in Ruddick et al. (2025) utilizes state-of-the-art MH algorithms to generate optimal Energy Management Systems (EMS) and models the EMS as interpretable DT.

4.2.3 Collaborative division of labor

For large-scale and complex problems, the complexity of the problem can be effectively reduced by decomposing the problem into different tasks executed in series in logical order. The original problem after task decomposition requires a phased solution. Each stage corresponds to a subtask, and different methods can be used for different subtasks, and the division of labor and cooperation between different methods constitute a complete solution to the original problem.

Many research works apply MH and ML to different subtasks of the same problem to achieve different functions in different steps, thus eventually providing an ideal solution to the original problem. Tayal et al. (2020) completes the Sustainable Facility Layout Problem (SFLP) in four stages to maximize performance and minimize operating costs. Firstly, the dimension is reduced. Secondly, two kinds of MH algorithms are combined to complete the generation of simulation and Facility Layout Problem (FLP). Then, the data envelopment analysis and ML(DEA-ML) are combined to complete data fusion and prediction. Finally, K-Means completes data mining to determine the maximum number of SFLPs that meet the sustainability standards. To solve the batch-flow Flexible Job-Shop Scheduling Problem (FJSP-LS), Li et al. (2023) divided the FJSP-LS problem into two stages: determining the optimal scheduling scheme and determining the optimal scheduling scheme for the subsections. In the first phase, ABC is used to search for the best scheduling scheme. In the second stage, RL is used to search for the best alternative for the subplot. By solving the FJSP-LS problem in stages, the complexity of the solution space is reduced, and the search ability of the algorithm is improved. Wang et al. (2023) trains SVM and GPR to predict the three indicators of engines. For the three indicators obtained, the Pareto fronts are solved by the nondominated sorting genetic algorithm III (NSGA-III).

4.3 Summary and discussion

MH-assisted ML is another angle of combining MH and ML with certain advantages and disadvantages. The advantage of this approach is that MH can find better model parameters or structures by performing global optimization in the search space and avoiding getting trapped in local optimal solutions. The ability of MH to handle various types of problems and constraints, including non-convex, non-continuous, and multi-modal problems, provides ML with a broader search space. MH and ML work together and complement each other to improve the whole system's performance. The literature in this section proves that MH algorithms can improve ML models' accuracy and generalization ability by jumping out of local optima and optimizing ML model parameters. MH algorithms can improve ML models' accuracy and generalization ability by jumping out of local optima and optimizing ML model parameters.

However, such methods still have drawbacks. For example, MHs often require significant computational resources to run, which may limit their application on large datasets or complex models. The running time of the algorithm may also be long, which affects the efficiency and feasibility of practical applications. Just like ML models, MH also requires the adjustment of its parameters to achieve optimal performance. This may require a lot of experimentation and expertise, increasing the difficulty and complexity of the application. In some cases, MH may also exhibit unstable behavior. For example, it is difficult to converge on a satisfactory solution to some problems. This may affect the performance and stability of ML.

Figure 7 shows the distribution characteristics of the specific methods adopted by the included literature in terms of the categories to which they belong. The hybrid algorithms of MH-assisted ML can be divided into two categories: integrated optimization MH-assisted ML and serial cooperative MH-assisted ML. There is a significant difference in the amount of literature between the two categories.

The relatively large number of papers on integrated optimization MH-assisted ML indicates that integrated optimization MH-assisted ML is currently a hot topic and the main research direction. Integrated optimization MH-assisted ML has received extensive attention and research due to its effectiveness, flexibility, and wide range of application scenarios in practical problem-solving. Even though there has been a lot of research, there may still be room for improvement and optimization of integrated optimization MH-assisted ML. In the future, more attention can be paid to improving the performance of integrated optimization MH-assisted ML by improving the algorithm structure, optimizing parameter settings, or exploring new application scenarios. In addition, the combination with technology in other fields can also be considered to generate more innovation points.

Despite the relatively small amount of literature, serial cooperative MH-assisted ML still has a certain research basis and application prospect, and it is a research direction that cannot be ignored. Serial cooperative MH-assisted ML represents a different research idea or application scenario with its unique advantages and application value in some specific fields or problems. Serial cooperative MH-assisted ML has important application value in the analysis and processing of data streams, and more attention can be paid to its algorithm

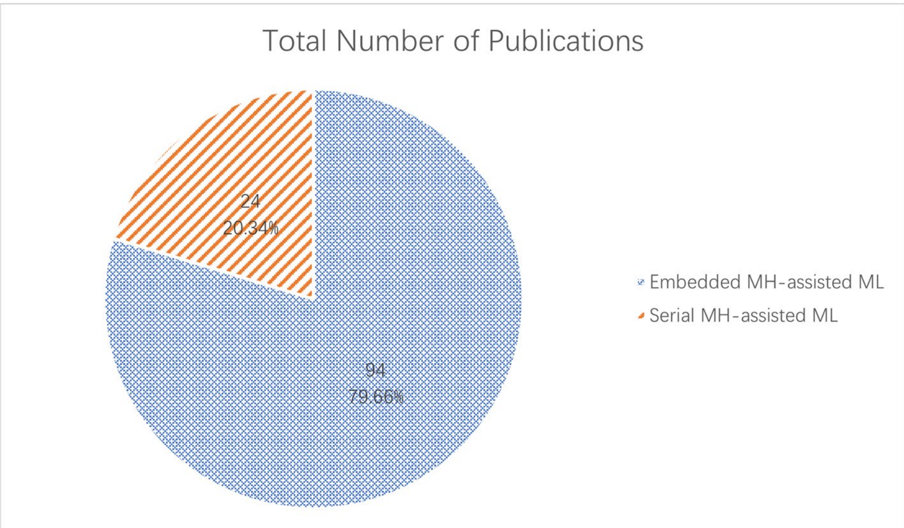

Fig. 7 Category analysis of the methods used in the included literature on MH-assisted ML

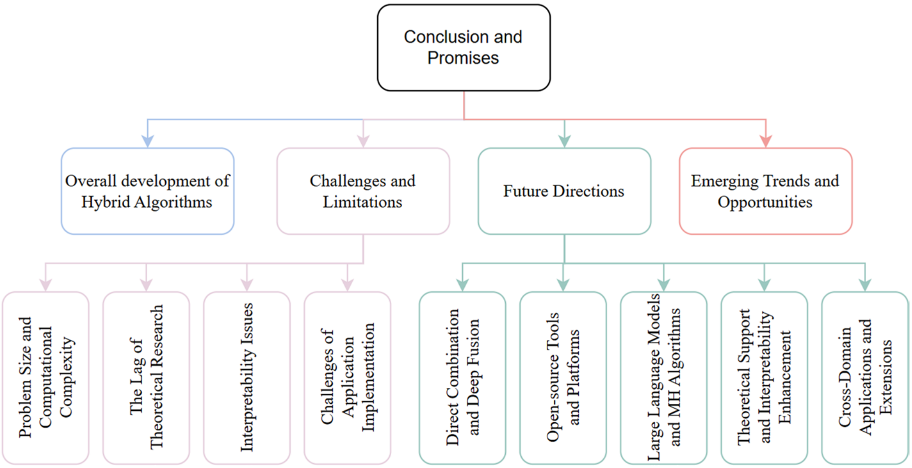

Fig. 8 Organization of the conclusion and promises

principle, application scenarios, and potential optimization methods in the future. Future research can build on the integrated optimization of MH-assisted ML, while the potential and value of serial cooperative MH-assisted ML should not be overlooked.

5 Conclusion and promises

The hybrid algorithm combining MH and ML has great potential in improving optimization efficiency and solving complex problems, but it still faces challenges such as high-dimensional computational complexity, theoretical lag, interpretability, and industry adaptation. Future development will focus on deep fusion, support for open-source tools and platforms, large language models, theoretical support and interpretability enhancement, and cross-domain adaptation to promote its wide application and sustainable development in practical applications. The organization of this section is shown in Fig. 8.

This classification approach systematically organizes the developmental trajectory of hybrid algorithms by integrating their overall progress, current challenges, future directions, and prospects for impact. It highlights the bottleneck issues in contemporary research while providing clear guidance for future endeavors. This method not only facilitates a clear understanding of the current state and future trends of hybrid algorithms for readers but also encourages researchers to explore different directions, thereby promoting the further development of hybrid algorithms.

5.1 Overall development of hybrid algorithms

MH and ML are pivotal in addressing challenges within complex systems. Their integration significantly amplifies the capabilities of hybrid algorithms, opening avenues for application across diverse domains. The two primary research directions in this integration-ML-assisted MH and MH-assisted ML-each possess unique developmental trajectories. In ML-assisted MH, the initial focus was on utilizing ML models to predict and assess the performance of metaheuristic searches, thereby refining the search strategy. As research progresses, the integration becomes more sophisticated, with ML and MH becoming closely intertwined to achieve dynamic adjustments in search strategies. This synergy allows for real-time optimization, enhancing the efficiency and effectiveness of the search process.

Conversely, MH-assisted ML initially centered on optimizing ML models' parameters and structure through metaheuristic techniques. However, as research deepens, the role of MH has expanded. It now encompasses not just hyperparameter tuning but also feature selection and model architecture design, all aimed at bolstering ML model performance. Cutting-edge research is now converging on the synergy between DL and these hybrid approaches, aiming to unlock an even broader spectrum of optimization potential. By harnessing the complementary strengths of MH and ML, these advanced techniques promise to revolutionize the way we tackle optimization challenges in complex systems.

The literature review suggests that there are two primary collaboration modes for ML and MH: direct combination and deep fusion. The direct combination approach emphasizes the independence of each method, maintaining distinct operational boundaries. This method aims to harness the unique strengths of ML and MH through synergistic collaboration, achieving a common goal. While the methods operate independently, they coordinate their efforts to complete tasks, preserving their individual characteristics post-combination. This approach is particularly adept at handling modular problems, which can be decomposed into stages or pipelines for complex task resolution. For instance, in ML-assisted MH, ML models serve as offline tools to fine-tune the parameters of MH algorithms or to analyze features, enhancing the search capabilities of MH. Conversely, MH-assisted ML employs MH algorithms to optimize the parameters of ML models or to select relevant features, improving the model's predictive accuracy.

The flexibility of direct combination allows for the integration of various ML models and MH algorithms, tailoring solutions to diverse task requirements. This approach is straightforward to implement and intuitive to manage, making it accessible for a wide range of applications. However, it's important to note that the direct combination may lead to increased computational costs and potentially diminished algorithmic efficiency due to the independent operations of each component.

Deep fusion integrates multiple methods or technologies into a cohesive whole through profound collaboration and mutual influence. In this approach, the boundaries between each method may become indistinct or even vanish entirely. The tightly coupled methods create a unified architecture that not only influences and enhances each other but also redefines their functionalities. For instance, in ML-assisted MH, ML models dynamically adjust the objective function or search operators of the MH algorithms, thereby influencing the algorithms' search processes. Conversely, MH-assisted ML embeds the MH algorithm within the training framework, allowing for real-time adjustments to the parameters of the ML model.

The synergy achieved through deep fusion can lead to significantly improved performance, often resulting in new algorithms or frameworks that can address complex problems that single approaches cannot solve. Systems developed through deep fusion typically exhibit greater coherence, minimizing interface complexity among different components and enhancing overall efficiency. However, implementing deep fusion presents challenges, as it requires a comprehensive understanding of the underlying principles and mechanisms of both ML and MH algorithms.

At present, the direct combination method predominates in engineering applications, owing to its straightforward implementation and adaptability. This approach allows for the straightforward integration of ML and MH techniques without the need for intricate fusion. However, with advancements in DL and RL, deep fusion is increasingly demonstrating its superiority, offering theoretical depth and enhanced performance capabilities.

Both collaboration methods-direct combination and deep fusion-find their place in ongoing research, chosen based on the specific demands of the application scenario and the nature of the tasks at hand. Direct combination excels in environments where resources are constrained or the tasks are well-defined, capitalizing on its simplicity and ease of management. On the other hand, deep fusion is particularly advantageous for tackling high-complexity tasks or in applications that necessitate advanced levels of automation, where the tight integration of ML and MH can lead to more sophisticated and efficient solutions.

5.2 Challenges and limitations

Although the combination of MH and ML has significant advantages, it also faces many technical and application challenges. The problem scale and computational complexity, the lag of theoretical research, the interpretability problem, and the difficulty of application landing have restricted the promotion of its practical application to a certain extent. The problem scale and computational complexity directly affect the efficiency and scalability of algorithms. The lagging theoretical research results in an insufficient understanding of the intrinsic mechanisms of hybrid methods. The interpretability problem restricts the application of models in high-risk decision-making scenarios. The difficulty in application landing is reflected in the adaptation and performance optimization in real-world scenarios. These challenges to some extent limit the promotion of practical applications. Therefore, future research should focus on addressing these key issues to facilitate further development and practical application of technologies in this field.

Problem size and computational complexity

MH algorithms are easy to fall into the “curse of dimensionality" when dealing with high-dimensional optimization problems. That is, with the increase of the dimension of the problem, the solution space expands rapidly, resulting in the decline of search efficiency and a sharp increase in computational complexity. The reason is that MH algorithms rely on random search strategies; although it can avoid local optimum, the number of solution evaluations and calculations increases exponentially in high-dimensional space, which affects the execution efficiency of the algorithm. In addition, ML models, especially DL models, require a large amount of data and computing resources during training. Training a model is even more computationally demanding for high-dimensional data, especially when dealing with large datasets, where the training process may involve billions of calculations. Combining MH algorithms with ML models may further increase computational complexity, especially when computing resources are limited, and may limit its efficiency and feasibility for practical applications.

The lag of theoretical research

Although MH and ML have rich theoretical foundations, respectively, the theoretical research on the combination of MH and ML is still in the preliminary stage and has not yet formed a systematic theoretical framework. A key issue is how to quantify the synergistic effect of the two and how to evaluate the effectiveness of different combination strategies. In addition, it is unclear under what conditions the combination of the two can be more effective than each other, which requires in-depth study of the mechanism of interaction between the algorithms. MH is essentially a heuristic search strategy that usually relies on randomness and local search, while ML performs pattern recognition through a data-driven learning process. After combining the two, it is still a challenge to rigorously analyze the convergence, optimality, and stability of the algorithm. Most of the existing research focuses on the exploration of application level and lacks in-depth theoretical analysis. Future research will concentrate on how to ensure combined algorithms' effectiveness and convergence in multiple problem backgrounds.

Interpretability issues

MH algorithms and complex ML models (e.g., DL) have their own "black box" characteristics; that is, their decision-making processes and optimization mechanisms are difficult to explain and understand for external users. MH algorithms can effectively find the global optimal solution through heuristic search strategy and stochastic process optimization problems, but their search path and results are difficult to predict and understand. On the other hand, ML models deal with the data complex by multi-layer nonlinear transformation, and their internal characteristics and decision-making process are highly abstract, which is difficult to understand intuitively. When these two are combined, the overall interpretability and transparency of the system may be further reduced, causing the decision-making process to become more difficult to trace and understand. This is particularly critical in critical domains such as healthcare and finance, which require the model's decisions to be clearly understood and validated. The lack of interpretability may lead to a decrease in user trust in the system and even hinder the adoption and application of the model, thus affecting its practical application. Therefore, improving the interpretability of hybrid algorithms, especially in high-risk domains, is a key challenge to promote the widespread application of these methods.

Challenges of application implementation

Optimization problems in different domains usually have specific constraints and application requirements, which makes the combination of MH and ML have to be designed with a lot of customization. For example, production scheduling problems in manufacturing may involve complex constraints on time, resources, and equipment, while risk assessment in the financial domain requires consideration of dynamic market changes and regulatory rules. To ensure the effectiveness and practical feasibility of the solution, these domain-specific constraints require targeted tuning and optimization of the algorithm during design. Therefore, several engineering challenges need to be overcome to efficiently combine the two methods and apply them to real industrial systems. First, hardware resource constraints, such as computing power and storage space, may limit large-scale data processing and complex model training, especially in real-time decision systems. Secondly, the integration and compatibility of software frameworks are also very prominent. Existing ML platforms and metaheuristic optimization tools often lack the ability to seamlessly connect, and how to efficiently cooperate between different frameworks and ensure the stability and scalability of the algorithm is still a big challenge in the implementation process. The solution to these problems is crucial to promote the practical application of the combination of the two.

5.3 Future directions

Hybrid methods that combine MH and ML have emerged as a popular research topic. Related research is expected to maintain its popularity and evolve with new development trends in deep fusion, open-source tools and platforms, large language models, theoretical support, cross-domain application, and expansion.

Direct integration and deep fusion significantly enhance model performance and robustness through multi-level, multi-modal data integration and optimization; the proliferation of open-source tools and platforms lowers research barriers, promoting algorithm sharing and community collaboration; the combination of large language models and MH algorithms strengthens the modeling and solving capabilities for complex problems, advancing intelligent decision-making and generative tasks; theoretical support and interpretability enhancement provides a solid mathematical foundation and transparent decision-making mechanisms, improving trustworthiness and practicality; cross-domain applications and extensions enable efficient generalization and performance optimization of methods in diverse scenarios. These directions collectively drive technological breakthroughs and application expansion of MH and ML hybrid methods, offering more efficient, reliable, and universally applicable solutions for future research.

Direct combination and deep fusion

In the hybrid algorithm combining ML and MH, the collaboration mode of the two is not only a direct combination but also deep fusion.

Direct combination highlights the independence of each method, with clear boundaries between them, aiming to leverage their respective strengths for achieving the goal through effective collaboration Karimi-Mamaghan et al. (2022). Methods are loosely coupled and run independently while collaborating to accomplish tasks Szénási and Légrádi (2024). After being directly combined, each method still retains independent characteristics. Direct conjunction suits modular problems such as solving complex tasks by pipelining or staging strategies. For example, ML-assisted MHs use ML models as an offline tool for tuning the parameters of MH algorithms or for feature analysis. MH-assisted MLs use MH algorithms to adjust the parameters of ML models or perform feature selection. The direct combination allows for the flexible integration of various ML models and MH algorithms to address different tasks, making it simple to implement and easy to understand and operate. The direct combination may increase additional computational costs and reduce algorithm efficiency.

Deep fusion integrates multiple methods or technologies to form an indivisible whole through deep collaboration or mutual influence. After deep fusion, the boundaries of each method may become blurred or even disappear. Methods are tightly coupled to form a unified architecture, influencing and enhancing each other and redefining functionality. For example, ML-assisted MHs utilize ML models to dynamically adjust the objective function or search operator of MH algorithms, which affects the search process of algorithms. MH-assisted MLs embed MH algorithms into the training process to adjust the parameters of ML models in real-time. After deep fusion, the synergy of each method may lead to improved performance. Deep convergence often results in new algorithms or frameworks capable of tackling problems that cannot be solved by a single approach. The system, after deep fusion, tends to be more coherent, which can avoid the interface complexity between different modules and improve efficiency. Implementing deep fusion is challenging and requires a deep understanding of the principles and mechanisms of ML and MH.

In ERL, deep integration is manifested in the organic combination of EA and RL. EA's genetic operations provide RL with diverse strategy exploration, while RL's feedback mechanism guides EA's optimization direction. This mutual penetration forms an adaptive strategy optimization loop. Under this deep integration, the population evolution of EA and the strategy learning of RL are no longer independent processes but are interwoven and co-evolving, thereby demonstrating performance that surpasses that of individual methods in complex decision-making tasks. This deep integration not only enhances the robustness and efficiency of the algorithm but also gives rise to novel solutions capable of addressing problems that are intractable for traditional methods. The deeply integrated system becomes more coherent, avoiding the complexity of interfaces between different modules and improving overall efficiency. However, the implementation of deep integration is challenging, requiring a deep understanding of the principles and mechanisms of ML and MH, as well as how to balance the exploration of EA with the exploitation of RL in ERL, ensuring that both components maximize their efficacy within the deeply integrated framework.

Currently, direct combination remains the primary method used in engineering practice due to its ease of implementation and flexibility. With the development of DL, RL, and other technologies, deep fusion has gradually occupied advantages in theory and performance. There are two collaboration methods in current research, and the specific one depends on the application scenario and task requirements. The direct combination is better suited for scenarios with limited resources or explicit tasks, while deep fusion is more suitable for high-complexity tasks or applications requiring a greater level of automation.

Open-source tools and platforms

In recent years, the hybrid algorithms combining MH and ML have received extensive attention in academic research and industrial applications, and their development has benefited from the promotion of more and more open-source tools and platforms (such as TensorFlow, PyTorch, and their optimization plugins) Van Thieu and Mirjalili (2023). These platforms significantly lower the threshold for combining MH with ML through modular design and highly flexible interfaces. For example, TensorFlow provides a powerful computational graph mechanism and optimization tools (such as TensorFlow Probability), which allow users to embed optimization tasks into machine learning models. PyTorch helps researchers quickly implement hybrid algorithms and develop them iteratively with its dynamic computation graph and rich community resources. In addition, the open-source platform also integrates with several automated tuning tools (e.g., Optuna, Ray Tune), which can automatically adjust hyperparameters and model structure with MH algorithms, thus significantly improving development efficiency and model performance. At the same time, these tools generally support distributed computing and hardware acceleration, such as GPU and TPU, which makes it possible to solve large-scale and complex optimization problems. In practical applications, with the support of open-source tools, hybrid algorithms have been widely used in intelligent manufacturing, medical health, financial analysis, automatic driving, and other fields, providing efficient solutions for global optimization, real-time decision-making, and multi-objective optimization problems. In addition to promoting the technical innovation of hybrid algorithms, the continuous development of open-source tools accelerates their popularization and application in more fields, providing a broad prospect for the development of intelligent optimization technology.

Nevertheless, these open-source tools and platforms, while facilitating the combination of MH and ML, also present certain limitations. For instance, although TensorFlow and

PyTorch offer robust functionalities, their scalability may be constrained when dealing with large-scale data and complex models Rojas et al. (2021). Moreover, specialized MH libraries (such as DEAP) may encounter compatibility issues with mainstream open-source tools and platforms. For example, DEAP, a popular metaheuristic algorithm library, may face interface incompatibility or performance degradation when integrated with TensorFlow or PyTorch. Additionally, the high hardware resource demands of these tools (e.g., the use of GPUs and TPUs) may also restrict their accessibility for certain users or teams.

Large language models and MH algorithms

With their powerful natural language understanding and generation capabilities, large language models (LLMs), such as GPT, also show broad prospects in complex system optimization. Future combinations of LLM and MH algorithms will bring new solutions for complex system optimization. On the one hand, LLM can generate input formats or rules for MH algorithms, reduce the complexity of manual modeling, and assist with parameter tuning, setting initial values, or dynamically modifying search strategies in the optimization process Stein and Bäck (2024); Pattanashetty et al. (2024). Furthermore, by analyzing data during the optimization process in real-time and generating strategic recommendations (e.g., heuristic rule tuning or algorithm fine-tuning), LLMs can provide knowledge and experience-based heuristic direction Liu et al. (2024). On the other hand, MH algorithms can supplement the limitations of LLM in generating a single answer and improve the solution space exploration ability of models for complex optimization tasks (Huang et al. 2024). For example, in text mining and optimization, MH algorithms select the optimal text features or keywords, and then LLM performs semantic optimization on the generated results. Therefore, combining LLM and MH algorithms will provide a more intelligent, flexible, and efficient solution for complex system optimization.

Theoretical support and interpretability enhancement

With the wide application of hybrid methods, the demand for their theoretical support and interpretability is also increasing. By integrating the optimization ability of MH and the data-driven characteristics of ML, the hybrid method improves the efficiency and accuracy of the solution. However, their internal mechanisms and decision-making processes are often complex and difficult to explain. Therefore, future research will pay more attention to the in-depth discussion of the theoretical basis of the hybrid algorithm and establish a rigorous mathematical model and proof system, aiming at a deeper understanding of its internal mechanism and working principle (Szénási and Légrádi 2024). At the same time, enhancing interpretability has become a crucial aspect of developing hybrid methods. By introducing visualization technology, feature importance evaluation, and other means, the hybrid method will provide a more intuitive and understandable decision-making basis (Saraswat and Tyagi 2023). This will help to enhance the trust and reliability of hybrid methods, enhance their application trust in high-risk fields, and promote their wider application and development.

Cross-domain applications and extensions

The hybrid method combining MH and ML has many application prospects, and future research will pay more attention to its application and extension in different fields. For instance, in fields such as intelligent manufacturing (Fu et al. 2022; Neelakandan et al. 2022), financial risk management (Altan et al. 2019; Petrović et al. 2022), medical diagnosis (Eid et al. 2022; Navazi et al. 2023), and others, hybrid algorithms can significantly enhance the decision-making efficiency and intelligence level of the system. In optimization problems, image processing, natural language processing, and other fields, hybrid algorithms are expected to achieve breakthrough progress. With the continuous emergence of new technologies, such as quantum computing and edge computing, hybrid methods will also be combined with these new technologies to explore more innovative application modes. At the same time, cross-domain cooperation will become an important force in promoting the development of hybrid methods, and experts from different fields will jointly explore new application scenarios and optimization strategies by sharing knowledge and exchanging experiences. The establishment and improvement of knowledge-sharing mechanisms will facilitate the flow and complementarity of knowledge between different fields, promote the cross-domain expansion of hybrid methods, and accelerate the technical exchange and cooperation between industries.

5.4 Emerging trends and opportunities

The integration of MH and ML holds great promise for future applications and is expected to achieve breakthroughs in various domains. This hybrid approach leverages the global search capabilities of MH and the data-driven optimization capabilities of ML, thereby significantly enhancing adaptability in dynamic environments (Szénási and Légrádi 2024). For instance, in dynamic optimization problems, the hybrid method can rapidly adapt to environmental changes and avoid local optima through ML-assisted parameter tuning and initialization. Moreover, this approach has demonstrated remarkable performance in fields such as physical simulation, scheduling problems, and feature selection.

In terms of adaptive optimization mechanisms, future hybrid algorithms will feature more advanced adaptive parameter tuning mechanisms, which can automatically adjust algorithm parameters based on problem characteristics and data variations. This will enhance the generality and robustness of the algorithms. In the realm of decision-making, hybrid algorithms can provide strong technical support for intelligent decision systems, aiding decision-makers in making scientifically sound and rational choices in complex environments. For real-time optimization problems, hybrid algorithms, equipped with real-time learning and feedback mechanisms, can process real-time data streams to achieve online optimization and adaptive adjustments (Molokomme et al. 2024).

This hybrid approach not only drives innovation in algorithmic theory and technical research in academia but also promotes interdisciplinary integration and development (Szénási and Légrádi 2024). In application domains, it significantly enhances the level of intelligence, especially in scenarios requiring rapid response and optimization (Molokomme et al. 2024). Future research directions should focus on reducing computational costs, improving model transparency, and further optimizing the design of hybrid methods. Through these improvements, the combination of MH and ML will offer novel solutions for complex problem-solving and drive further development in computer science and related fields.

5.5 Conclusion

The relationship between MH and ML algorithms is becoming more intertwined, with significant implications for artificial intelligence. This paper provides an in-depth review of the strategies that leverage the synergistic effects of MH and ML, examining both MH-assisted ML and ML-assisted MH frameworks. We introduce a new classification system for these hybrid algorithms, offering a structured analysis from two distinct yet complementary viewpoints and synthesizing the findings from relevant research.

Our analysis reveals that the integrated application of MH and ML significantly bolsters an algorithm's adaptability, robustness, and generalization capabilities, particularly in dynamic environments and intricate system challenges. The MH-ML hybrid algorithms can generate superior solutions, outperforming individual methods in optimization strength and solution efficiency. This integrated approach is adept at tackling complex and nonlinear problems, uncovering obscured patterns, and facilitating self-adaptation and intelligent responses within complex systems. The research presented in this paper has traced the evolution of ML and MH integration from simple combinations to more profound and integrated fusions, showcasing the advancements and potential of these hybrid methodologies.

Building on analyzing the synergistic effects between MH and ML, this paper delves into the challenges researchers face when integrating these two domains. We scrutinize the obstacles encountered in the hybridization of MH and ML, offering insights into the complexities and potential pitfalls that arise from this interdisciplinary work. Furthermore, the paper outlines promising avenues for future research, equipping readers with a comprehensive understanding of the intricacies and potential of MH-ML hybrid algorithms. We aim to guide the scientific community in harnessing the full potential of MH-ML integration and to stimulate innovative approaches to complex problem-solving.

Author contributions J.W. and C.L. proposed ideas; K.S. guided research methods; R.Z. summarized the literature and wrote the main manuscript; H.I. and Y.J. revised the manuscript text. All authors reviewed the manuscript.

Data availability No datasets were generated or analysed during the current study.

Declarations

Competing interests The authors declare no competing interests.

References

Abdallaoui S, Aglzim EH, Chaibet A, Kribèche A (2022) Thorough review analysis of safe control of autonomous vehicles: path planning and navigation techniques. MDPI. https://doi.org/10.3390/en15041358

Agrawal P, Abutarboush HF, Ganesh T, Mohamed AW (2021) Metaheuristic algorithms on feature selection: a survey of one decade of research (2009–2019). IEEE Access 9:26766–26791. https://doi.org/10.1109/ACCESS.2021.3056407

Agushaka JO, Ezugwu AE (2022) Initialisation approaches for population-based metaheuristic algorithms: a comprehensive review. Appl Sci 12(2):896. https://doi.org/10.3390/app12020896

Ahmad S (2024) Intelligent crowd density classification using improved metaheuristics with transfer learning model on smart cities. SN Comput Sci 5(8):1064. https://doi.org/10.1007/s42979-024-03435-7

Ahmad A, Yadav AK, Singh A, Singh DK (2024) A machine learning-genetic algorithm based models for optimization of intensification process through microwave reactor: a new approach for rapid and sustainable synthesis of biodiesel from novel hiptage benghalensis seed oil. Fuel 374:132470. https://doi.org/10.1016/j.fuel.2024.132470

Ahmadi E, Garcia-Arce A, Masel DT, Reich E, Puckey J, Maff R (2019) A metaheuristic-based stacking model for predicting the risk of patient no-show and late cancellation for neurology appointments. IISE Trans Healthc Syst Eng 9:272–291. https://doi.org/10.1080/24725579.2019.1649764

Ahmadianfar I, Shirvani-Hosseini S, He J, Samadi-Koucheksaraee A, Yaseen ZM (2022) An improved adaptive neuro fuzzy inference system model using conjoined metaheuristic algorithms for electrical conductivity prediction. Sci Rep. https://doi.org/10.1038/s41598-022-08875-w

Akay B, Karaboga D, Akay R (2022) A comprehensive survey on optimizing deep learning models by metaheuristics. Artif Intell Rev 55(2):829–894. https://doi.org/10.1007/s10462-021-09992-0

Akhter MN, Mekhilef S, Mokhlis H, Shah NM (2019) Review on forecasting of photovoltaic power generation based on machine learning and metaheuristic techniques. Inst Eng Technol. https://doi.org/10.1049/iet-rpg.2018.5649

Akyol S, Yildirim M, Alatas B (2023) Multi-feature fusion and improved bo and igwo metaheuristics based models for automatically diagnosing the sleep disorders from sleep sounds. Comput Biol Med. https://doi.org/10.1016/j.compbiomed.2023.106768

Alba E, Nakib A, Siarry P (2013) Metaheuristics for dynamic optimization. Springer, Berlin

Alkabbani H, Ahmadian A, Zhu Q, Elkamel A (2021) Machine learning and metaheuristic methods for renewable power forecasting: a recent review. Front Media SA. https://doi.org/10.3389/fceng.2021.665415

Altan A, Karasu S, Bekiros S (2019) Digital currency forecasting with chaotic meta-heuristic bio-inspired signal processing techniques. Chaos, Solitons Fract 126:325–336. https://doi.org/10.1016/j.chaos.2019.07.011

Ansari A, Ahmad IS, Bakar AA, Yaakub MR (2020) A hybrid metaheuristic method in training artificial neural network for bankruptcy prediction. IEEE Access 8:176640–176650. https://doi.org/10.1109/ACCESS.2020.3026529

Bacanin N, Petrović A, Zivkovic M, Bezdan T, Antonijevic M (2021) Feature selection in machine learning by hybrid sine cosine metaheuristics. International conference on advances in computing and data sciences. Springer, Cham, pp 604–616

Bandaru S, Deb K (2016) Metaheuristic techniques. Decision sciences. CRC Press, Boca Raton, pp 709–766

Bartz-Beielstein T, Branke J, Mehnen J, Mersmann O (2014) Evolutionary algorithms. Wiley Interdiscip Rev: Data Min Knowl Discovery 4(3):178–195. https://doi.org/10.1002/widm.1124

Bayliss C, Juan AA, Currie CSM, Panadero J (2020) A learnheuristic approach for the team orienteering problem with aerial drone motion constraints. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2020.106280

Beheshti Z, Shamsuddin SMH (2013) A review of population-based meta-heuristic algorithms. Int. J. Adv. Soft Comput. Appl 5(1):1–35. https://doi.org/10.1109/TEVC.2021.3130835

Behjat A, Zeng C, Rai R, Matei I, Doermann D, Chowdhury S (2020) A physics-aware learning architecture with input transfer networks for predictive modeling. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2020.106665

Bertolini M, Mezzogori D, Neroni M, Zammori F (2021) Machine learning for industrial applications: a comprehensive literature review. Expert Syst Appl 175:114820. https://doi.org/10.1016/j.eswa.2021.114820

Bertsimas D, Tsitsiklis J (1993) Simulated annealing. Stat Sci 8(1):10–15. https://doi.org/10.1214/ss/1177011077

Birattari M, Kacprzyk J (2009) Tuning metaheuristics: a machine learning perspective. Springer, Berlin

Bishop CM (2013) Model-based machine learning. Philos Trans R Soc A: Math, Phys Eng Sci 371(1984):20120222. https://doi.org/10.1098/rsta.2012.0222

Blum C, Roli A (2003) Metaheuristics in combinatorial optimization: overview and conceptual comparison. ACM Comput Surv (CSUR) 35(3):268–308. https://doi.org/10.1145/937503.937505

Bodnar C, Day B, Lió P (2020) Proximal distilled evolutionary reinforcement learning. Proc AAAI Conf Artif Intell 34:3283–3290. https://doi.org/10.1609/aaai.v34i04.5728

Boubekeur Y, Mussbacher G, McIntosh S (2020) Automatic assessment of students' software models using a simple heuristic and machine learning. In: Proceedings of the 23rd ACM/IEEE international conference on model driven engineering languages and systems: companion proceedings, pp. 84–93. Association for Computing Machinery, Inc, Virtual Event, Canada. https://doi.org/10.1145/3417990.3418741

Burlutskiy N, Petridis M, Fish A, Chernov A, Ali N (2016) An investigation on online versus batch learning in predicting user behaviour. In: Research and development in intelligent systems XXXIII: incorporating applications and innovations in intelligent systems XXIV 33. Springer, pp. 135–149. https://doi.org/10.1007/978-3-319-47175-4_9

Cahon S, Melab N, Talbi E-G (2004) ParadisEO: a framework for the reusable design of parallel and distributed metaheuristics. https://doi.org/10.5555/838237.838565

Calvet L, Armas JD, Masip D, Juan AA (2017) Learnheuristics: hybridizing metaheuristics with machine learning for optimization with dynamic inputs. De Gruyter Open Ltd. https://doi.org/10.1515/math-2017-0029

Cao MT, Nguyen NM, Chang KT, Tran XL, Hoang ND (2021) Automatic recognition of concrete spall using image processing and metaheuristic optimized logitboost classification tree. Adv Eng Softw. https://doi.org/10.1016/j.advengsoft.2021.103031

Cerezo M, Verdon G, Huang H-Y, Cincio L, Coles PJ (2022) Challenges and opportunities in quantum machine learning. Nat Comput Sci 2(9):567–576. https://doi.org/10.1038/s43588-022-00311-3

Chen Z, Song D (2023) Modeling landslide susceptibility based on convolutional neural network coupling with metaheuristic optimization algorithms. Int J Digit Earth 16:3384–3416. https://doi.org/10.1080/17538947.2023.2249863

Chen P, Wang Q (2024) Learning for multiple purposes: A q-learning enhanced hybrid metaheuristic for parallel drone scheduling traveling salesman problem. Comput Ind Eng 187:109851. https://doi.org/10.1016/j.cie.2023.109851

Chen Z, Causmaecker PD, Dou Y (2023) A combined mixed integer programming and deep neural network-assisted heuristics algorithm for the nurse rostering problem. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2022.109919

Chen D, Li X, Li S (2023) A novel convolutional neural network model based on beetle antennae search optimization algorithm for computerized tomography diagnosis. IEEE Trans Neural Netw Learn Syst 34:1418–1429. https://doi.org/10.1109/TNNLS.2021.3105384

Chen M, Gao L, Chen Q, Liu Z (2020) Dynamic partial removal: A neural network heuristic for large neighborhood search. ArXiv **abs/2005.09330**https://doi.org/10.48555/abs-2005.09330

Chiroma H, Gital AY, Rana N, Abdulhamid SM, Muhammad AN, Umar AY, Abubakar AI (2020) Nature inspired meta-heuristic algorithms for deep learning: recent progress and novel perspective. In: Advances in computer vision: proceedings of the 2019 computer vision conference (CVC). Springer, 1 1, pp. 59–70. https://doi.org/10.1038/s41598-024-56670-6

Costa Oliveira AL, Britto A, Gusmão R (2023) Machine learning enhancing metaheuristics: a systematic review. Soft Comput 27(21):15971–15998. https://doi.org/10.1007/s00500-023-08886-

Dai L, Zhang H (2020) Propagation-model-free base station deployment for mobile networks: integrating machine learning and heuristic methods. IEEE Access 8:83375–83386. https://doi.org/10.1109/ACCESS.2020.2990631

Dakic P, Zivkovic M, Jovanovic L, Bacanin N, Antonijevic M, Kaljevic J, Simic V (2024) Intrusion detection using metaheuristic optimization within iot/iiot systems and software of autonomous vehicles. Sci Rep 14(1):22884. https://doi.org/10.1016/j.eswa.2024.125192

Dasi H, Ying Z, Ashab MFB (2024) Proposing hybrid prediction approaches with the integration of machine learning models and metaheuristic algorithms to forecast the cooling and heating load of buildings. Energy 291:130297. https://doi.org/10.1016/j.energy.2024.130297

Devikanniga D, Vetrivel K, Badrinath N (2019) Review of meta-heuristic optimization based artificial neural networks and its applications. J Phys: Conf Ser 1362:012074. https://doi.org/10.1007/s11063-022-11055-6

Ding Z, Huang Y, Yuan H, Dong H (2020) Introduction to reinforcement learning. Deep reinforcement learning: fundamentals, research and applications, 47–123 https://doi.org/10.1007/978-981-15-4095-0_2

Dogan H, Temel FA, Yolcu OC, Turan NG (2023) Modelling and optimization of sewage sludge composting using biomass ash via deep neural network and genetic algorithm. Biores Technol 370:128541. https://doi.org/10.1016/j.biortech.2022.01.874

Dokeroglu T, Kucukyilmaz T (2025) Multi-objective Harris hawk metaheuristic algorithms for the diagnosis of Parkinson's disease. Expert Syst Appl 270:126503. https://doi.org/10.1016/j.eswa.2025.125192

Dokeroglu T, Sevinc E, Kucukyilmaz T, Cosar A (2019) A survey on new generation metaheuristic algorithms. Comput Ind Eng 137:106040. https://doi.org/10.1016/j.cie.2019.106040

Dokeroglu T, Deniz A, Kiziloz HE (2022) A comprehensive survey on recent metaheuristics for feature selection. Neurocomputing 494:269–296. https://doi.org/10.1016/j.neucom.2022.04.047

Donatti A, Corrêa SL, Martins JSB, Abelem AJG, Both CB, Oliveira Silva F, Suruagy JA, Pasquini R, Moreira R, Cardoso KV, Carvalho TC (2024) Survey on machine learning-enabled network slicing: covering the entire life cycle. IEEE Trans Netw Serv Manag 21(1):994–1011. https://doi.org/10.1109/TNSM.2023.3287651

Dorigo M, Maniezzo V, Colorni A (1991) Ant system: an autocatalytic optimizing process. https://api.semanticscholar.org/CorpusID:18900660

Dorigo M, Stützle T (2003) The ant colony optimization metaheuristic: Algorithms, applications, and advances. Handbook of metaheuristics, 250–285 https://doi.org/10.1007/0-306-48056-5_9

Durgut R, Aydin ME, Atli I (2021) Adaptive operator selection with reinforcement learning. Inf Sci 581:773–790. https://doi.org/10.1016/j.ins.2021.10.025

Eid MM, El-Kenawy ESM, Khodadadi N, Mirjalili S, Khodadadi E, Abotaleb M, Alharbi AH, Abdelhamid AA, Ibrahim A, Amer GM, Kadi A, Khafaga DS (2022) Meta-heuristic optimization of lstm-based deep network for boosting the prediction of monkeypox cases. Mathematics. https://doi.org/10.3390/math10203845

Elaziz MA, Dahou A, Abualigah L, Yu L, Alshinwan M, Khasawneh AM, Lu S (2021) Advanced metaheuristic optimization techniques in applications of deep neural networks: a review. Neural Comput Appl. https://doi.org/10.1007/s00521-021-05960-5

El-Hasnony IM, Barakat SI, Mostafa RR (2020) Optimized Anfis model using hybrid metaheuristic algorithms for Parkinson's disease prediction in iot environment. IEEE Access 8:119252–119270. https://doi.org/10.1109/ACCESS.2020.3005614

El-Kenawy ESM, Mirjalili S, Ibrahim A, Alrahmawy M, El-Said M, Zaki RM, Eid MM (2021) Advanced meta-heuristics, convolutional neural networks, and feature selectors for efficient covid-19 x-ray chest image classification. IEEE Access 9:36019–36037. https://doi.org/10.1109/ACCESS.2021.3061058

Elmasry W, Akbulut A (2020) Zaim AH Evolving deep learning architectures for network intrusion detection using a double pso metaheuristic. Comput Netw. https://doi.org/10.1016/j.comnet.2019.107042

Eshtay M, Faris H, Obeid N (2019) Metaheuristic-based extreme learning machines: a review of design formulations and applications. Int J Mach Learn Cybern 10:1543–1561. https://doi.org/10.1007/s13042-018-0833-6

Fadhillah MF, Lee S, Lee CW, Park YC (2021) Application of support vector regression and metaheuristic optimization algorithms for groundwater potential mapping in gangneung-si, south korea. Remote Sens. https://doi.org/10.3390/rs13061196

Fallahi A, Bani EA, Niaki STA (2022) A constrained multi-item eoq inventory model for reusable items: reinforcement learning-based differential evolution and particle swarm optimization. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2022.118018

Farahmand-Tabar S, Ashtari P (2024) Intelligent cross-entropy optimizer: a novel machine learning-based meta-heuristic for global optimization. Swarm Evol Comput 91:101739. https://doi.org/10.1016/j.swevo.2024.101518

Fattahi H, Hasanipanah M (2021) Prediction of blast-induced ground vibration in a mine using relevance vector regression optimized by metaheuristic algorithms. Natl Resour Res 30:1849–1863. https://doi.org/10.1007/s11053-020-09764-7

Fu K, Cai X, Yuan B, Yang Y, Yao X (2022) An efficient surrogate assisted particle swarm optimization for antenna synthesis. IEEE Trans Antennas Propag 70:4977–4984. https://doi.org/10.1109/TAP.2022.3153080

García J, Maureira C (2021) A knn quantum cuckoo search algorithm applied to the multidimensional knapsack problem. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2020.107077

Gaspar A, Oliva D, Cuevas E, Zaldívar D, Pérez M, Pajares G (2021) Hyperparameter optimization in a convolutional neural network using metaheuristic algorithms. Springer, Cham, pp 37–59

Ghahramani Z (2003) Unsupervised learning. Summer school on machine learning. Springer, Cham, pp 72–112

Gibson S, Issac B, Zhang L, Jacob SM (2020) Detecting spam email with machine learning optimized with bio-inspired metaheuristic algorithms. IEEE Access 8:187914–187932. https://doi.org/10.1109/ACCESS.2020.3030751

Gill SS, Xu M, Ottaviani C, Patros P, Bahsoon R, Shaghaghi A, Golec M, Stankovski V, Wu H, Abraham A, Singh M, Mehta H, Ghosh SK, Baker T, Parlikad AK, Lutfiyya H, Kanhere SS, Sakellariou R, Dustdar S, Rana O, Brandic I, Uhlig S (2022) AI for next generation computing: emerging trends and future directions. Elsevier, Amsterdam

Glover F (1990) Tabu search: a tutorial. Interfaces 20(4):74–94. https://doi.org/10.1287/inte.20.4.74

Gui J, Sun Z, Wen Y, Tao D, Ye J (2023) A review on generative adversarial networks: algorithms, theory, and applications. IEEE Trans Knowl Data Eng 35:3313–3332. https://doi.org/10.1109/TKDE.2021.3130191

Hakim WL, Rezaie F, Nur AS, Panahi M, Khosravi K, Lee CW, Lee S (2022) Convolutional neural network (cnn) with metaheuristic optimization algorithms for landslide susceptibility mapping in icheon, South Korea. J Environ Manag. https://doi.org/10.1016/j.jenvman.2021.114367

Halim AH, Ismail I, Das S (2021) Performance assessment of the metaheuristic optimization algorithms: an exhaustive review. Artif Intell Rev 54:2323–2409. https://doi.org/10.1007/s10462-020-09906-6

Han Y, Peng H, Mei C, Cao L, Deng C, Wang H, Wu Z (2023) Multi-strategy multi-objective differential evolutionary algorithm with reinforcement learning. Knowl-Based Syst 277:110801. https://doi.org/10.1016/j.knosys.2023.110801

Han Y, Wang L, Chen R, Fu W, Li T, Liu G, Zhou H (2024) Topside ionospheric tec modeling using multiple leo satellites based on genetic algorithm-optimized machine learning models. GPS Solut 28(1):19. https://doi.org/10.1007/s10291-023-01565-8

Han X, Xue Y, Wang Z, Zhang Y, Muravev A, Gabbouj M (2024) Sadenas: a self-adaptive differential evolution algorithm for neural architecture search. Swarm Evol Comput 91:101736. https://doi.org/10.1016/j.swevo.2024.101736

Harrison KR, Ombuki-Berman BM, Engelbrecht AP (2019) A parameter-free particle swarm optimization algorithm using performance classifiers. Inf Sci 503:381–400. https://doi.org/10.1016/j.ins.2019.07.016

Hastie T, Tibshirani R, Friedman J, Hastie T, Tibshirani R, Friedman J (2009) Overview of supervised learning. The elements of statistical learning: data mining, inference, and prediction. Springer, Cham, pp 9–41

He B, Armaghani DJ, Lai SH (2023) Assessment of tunnel blasting-induced overbreak: a novel metaheuristic-based random forest approach. Tunnell Underground Space Technol. https://doi.org/10.1016/j.tust.2022.104979

He Z, Wilson SB, Monjezi M, Tran TT (2024) Estimating Brazilian tensile strength of granite rocks using metaheuristic algorithms-based self-organizing neural networks. Rock Mech Rock Eng 57(7):4653–4668. https://doi.org/10.1007/s00603-024-03801-0

Hoang ND (2019) Image processing based automatic recognition of asphalt pavement patch using a metaheuristic optimized machine learning approach. Adv Eng Inform 40:110–120. https://doi.org/10.1016/j.aei.2019.04.004

Holland JH (1973) Genetic algorithms and the optimal allocation of trials. SIAM J Comput 2(2):88–105. https://doi.org/10.1137/0202009

Houssein EH (2019) Machine learning and meta-heuristic algorithms for renewable energy: a systematic review. Advanced control and optimization paradigms for wind energy systems. Springer, Cham, pp 165–187

Hu Z, Gong W, Pedrycz W, Li Y (2023) Deep reinforcement learning assisted co-evolutionary differential evolution for constrained optimization. Swarm Evol Comput 83:101387. https://doi.org/10.1016/j.swevo.2023.101194

Huang Q (2020) Model-based or model-free, a review of approaches in reinforcement learning. In: 2020 International conference on computing and data science (CDS). IEEE, pp. 219–221. https://doi.org/10.1109/CDS49703.2020.00049

Huang S, Yang K, Qi S, Wang R (2024) When large language model meets optimization. Swarm Evol Comput 90:101663. https://doi.org/10.1016/j.swevo.2024.101663

Hussain K, Mohd Salleh MN, Cheng S, Shi Y (2019) Metaheuristic research: a comprehensive survey. Artif Intell Rev 52:2191–2233. https://doi.org/10.1007/s10462-019-09728-4

Jaddi NS, Abdullah S (2020) Global search in single-solution-based metaheuristics. Data Technol Appl 54(3):275–296. https://doi.org/10.1108/DTA-07-2019-0115

Jiadong Q, Ohl JP, Tran T-T (2024) Predicting clay compressibility for foundation design with high reliability and safety: a geotechnical engineering perspective using artificial neural network and five metaheuristic algorithms. Reliabil Eng Syst Saf 243:109827. https://doi.org/10.1016/j.advengsoft.2023.103668

Jiang F, He J, Tian T (2019) A clustering-based ensemble approach with improved pigeon-inspired optimization and extreme learning machine for air quality prediction. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2019.105827

Jordan MI, Mitchell TM (2015) Machine learning: trends, perspectives, and prospects. Science 349(6245):255–260. https://doi.org/10.1126/science.aaa8415

José-García A, Jacques J, Sobanski V, Dhaenens C (2023) Metaheuristic biclustering algorithms: from state-of-the-art to future opportunities. ACM Comput Surv 56(3):1–38. https://doi.org/10.1145/3617590

Jovanovic D, Antonijevic M, Stankovic M, Zivkovic M, Tanaskovic M, Bacanin N (2022) Tuning machine learning models using a group search firefly algorithm for credit card fraud detection. Mathematics. https://doi.org/10.3390/math10132272

Kaelbling LP, Littman ML, Moore AW (1996) Reinforcement learning: a survey. J Artif Intell Res 4:237–285. https://doi.org/10.1613/jair.301

Kakarash ZA, Mardukhia F, Moradi P (2023) Multi-label feature selection using density-based graph clustering and ant colony optimization. J Comput Design Eng 10:122–138. https://doi.org/10.1093/jcde/qwac120

Kaleybar HJ, Davoodi M, Brenna M, Zaninelli D (2023) Applications of genetic algorithm and its variants in rail vehicle systems: a bibliometric analysis and comprehensive review. IEEE Access. https://doi.org/10.1109/ACCESS.2023.3292790

Karaboga D, Gorkemli B, Ozturk C, Karaboga N (2014) A comprehensive survey: artificial bee colony (abc) algorithm and applications. Artif Intell Rev 42:21–57. https://doi.org/10.1007/s10462-012-9328-0

Kardani N, Bardhan A, Gupta S, Samui P, Nazem M, Zhang Y, Zhou A (2022) Predicting permeability of tight carbonates using a hybrid machine learning approach of modified equilibrium optimizer and extreme learning machine. Acta Geotech 17:1239–1255. https://doi.org/10.1007/s11440-021-01257-y

Karimi-Mamaghan M, Mohammadi M, Meyer P, Karimi-Mamaghan AM, Talbi EG (2022) Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: A state-of-the-art. Elsevier, Amsterdam

Kaveh A, Khavaninzadeh N (2023) Efficient training of two anns using four meta-heuristic algorithms for predicting the frp strength. Structures 52:256–272. https://doi.org/10.1016/j.istruc.2023.03.178

Kaveh A, Eskandari A, Movasat M (2023) Buckling resistance prediction of high-strength steel columns using metaheuristic-trained artificial neural networks. Structures. https://doi.org/10.1016/j.istruc.2023.07.043

Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95 - international conference on neural networks, 4, pp. 1942–19484. https://doi.org/10.1109/ICNN.1995.488968

Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks. IEEE, vol. 4, pp. 1942–1948. https://doi.org/10.1109/ICNN.1995.488968

Khan MS, Jabeen F, Ghouzali S, Rehman Z, Naz S, Abdul W (2021) Metaheuristic algorithms in optimizing deep neural network model for software effort estimation. IEEE Access 9:60309–60327. https://doi.org/10.1109/ACCESS.2021.3072380

Khan M, Ullah Z, Mašek O, Naqvi SR, Khan MNA (2022) Artificial neural networks for the prediction of biochar yield: a comparative study of metaheuristic algorithms. Bioresour Technol. https://doi.org/10.1016/j.biortech.2022.127215

Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220(4598):671–680. https://doi.org/10.1126/science.220.4598.671

Kumar J, Singh AK (2021) Performance evaluation of metaheuristics algorithms for workload prediction in cloud environment. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2021.107895

Kuo R, Chiu T-H (2024) Hybrid of jellyfish and particle swarm optimization algorithm-based support vector machine for stock market trend prediction. Appl Soft Comput 154:111394. https://doi.org/10.1016/j.eswa.2024.125192

Ladosz P, Weng L, Kim M, Oh H (2022) Exploration in deep reinforcement learning: a survey. Inf Fusion 85:1–22. https://doi.org/10.1016/j.inffus.2022.03.003

Le TM, Vo TM, Pham TN, Dao SVT (2021) A novel wrapper-based feature selection for early diabetes prediction enhanced with a metaheuristic. IEEE Access 9:7869–7884. https://doi.org/10.1109/ACCESS.2020.3047942

Li Z (2023) A local opposition-learning golden-sine grey wolf optimization algorithm for feature selection in data classification. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2023.110319

Li Y, Liao C, Wang L, Xiao Y, Cao Y, Guo S (2023) A reinforcement learning-artificial bee colony algorithm for flexible job-shop scheduling problem with lot streaming. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2023.110658

Li C, Zhou J, Du K, Dias D (2023) Stability prediction of hard rock pillar using support vector machine optimized by three metaheuristic algorithms. Int J Min Sci Technol. https://doi.org/10.1016/j.ijmst.2023.06.001

Li S, Li W, Tang J, Wang F (2023) A new evolving operator selector by using fitness landscape in differential evolution algorithm. Inf Sci 624:709–731. https://doi.org/10.1016/j.ins.2022.11.071

Li W, Özcan E, Drake JH, Maashi M (2023) A generality analysis of multiobjective hyper-heuristics. Inf Sci 627:34–51. https://doi.org/10.1016/j.ins.2023.01.047

Li B, Shen L, Zhao Y, Yu W, Lin H, Chen C, Li Y, Zeng Q (2023) Quantification of interfacial interaction related with adhesive membrane fouling by genetic algorithm back propagation (gabp) neural network. J Colloid Interface Sci 640:110–120. https://doi.org/10.1016/j.jcis.2023.02.030

Li B, Wei Z, Wu J, Yu S, Zhang T, Zhu C, Zheng D, Guo W, Zhao C, Zhang J (2023) Machine learning-enabled globally guaranteed evolutionary computation. Nat Mach Intell 5(4):457–467. https://doi.org/10.1038/s42256-023-00655-z

Li W, Liang P, Sun B, Sun Y, Huang Y (2023) Reinforcement learning-based particle swarm optimization with neighborhood differential mutation strategy. Swarm Evol Comput. https://doi.org/10.1016/j.swevo.2023.101274

Li P, Hao J, Tang H, Fu X, Zhen Y, Tang K (2024) Bridging evolutionary algorithms and reinforcement learning: a comprehensive survey on hybrid algorithms. IEEE Trans Evol Comput. https://doi.org/10.1109/TEVC.2024.3443913

Lin Z, Gao K, Wu N, Suganthan PN (2024) Problem-specific knowledge based multi-objective meta-heuristics combined q-learning for scheduling urban traffic lights with carbon emissions. IEEE Trans Intell Transp Syst. https://doi.org/10.1109/TITS.2024.3397077

Lin Q, Liu Z, Yang Y, Wong K-C, Lu Y, Li J (2024) Multi-objective evolutionary neural architecture search for network intrusion detection. Swarm Evol Comput 91:101702. https://doi.org/10.1016/j.neucom.2023.120240

Liu Z, Yang D, Wang Y, Lu M, Li R (2023) Egnn: Graph structure learning based on evolutionary computation helps more in graph neural networks. Appl Soft Comput 135:110040. https://doi.org/10.1016/j.asoc.2023.110040

Liu H, Zong Z, Li Y, Jin D (2023) Neurocrossover: an intelligent genetic locus selection scheme for genetic algorithm using reinforcement learning. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2023.110680

Liu S, Chen C, Qu X, Tang K, Ong Y-S Large language models as evolutionary optimizers. In: 2024 IEEE congress on evolutionary computation (CEC). IEEE, pp. 1–8 (2024). https://doi.org/10.1145/3638530.3648432

Liu Z, Chen B, Zhou H, Koushik G, Hebert M, Zhao D (2020) Mapper: multi-agent path planning with evolutionary reinforcement learning in mixed dynamic environments. In: 2020 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, pp. 11748–11754. https://doi.org/10.1109/IROS45743.2020.9340876

Lucas F, Billot R, Sevaux M, Sörensen K (2020) Reducing space search in combinatorial optimization using machine learning tools. In: Kotsireas IS, Pardalos PM (eds) Learning and intelligent optimization. Springer, Cham, pp 143–150

Luo H, Paal SG (2021) Metaheuristic least squares support vector machine-based lateral strength modelling of reinforced concrete columns subjected to earthquake loads. Structures 33:748–758. https://doi.org/10.1016/j.istruc.2021.04.048

Luong NH, Phan QM, Vo A, Pham TN, Bui DT (2024) Lightweight multi-objective evolutionary neural architecture search with low-cost proxy metrics. Inf Sci 655:119856. https://doi.org/10.1016/j.ins.2023.119856

Ma R, Karimzadeh M, Ghabussi A, Zandi Y, Baharom S, Selmi A, Maureira-Carsalade N (2022) Assessment of composite beam performance using gwo-elm metaheuristic algorithm. Eng Comput 38:2083–2099. https://doi.org/10.1007/s00366-021-01363-1

Ma Z, He X, Yan P, Zhang F, Teng Q (2023) A fast and flexible algorithm for microstructure reconstruction combining simulated annealing and deep learning. Comput Geotech 164:105755. https://doi.org/10.1016/j.cageo.2023.05.028

Mahesh B (2020) Machine learning algorithms-a review. Int J Sci Res (IJSR) 9(1):381–386. https://doi.org/10.21275/ijsr-v9i1.2018.1234

Mahjoubi S, Barhemat R, Guo P, Meng W, Bao Y (2021) Prediction and multi-objective optimization of mechanical, economical, and environmental properties for strain-hardening cementitious composites (shcc) based on automated machine learning and metaheuristic algorithms. J Clean Prod. https://doi.org/10.1016/j.jclepro.2021.129665

Majumdar S, Khadka S, Miret S, McAleer S, Tumer K (2020) Evolutionary reinforcement learning for sample-efficient multiagent coordination. Int Conf Mach Learn. https://doi.org/10.5555/3524938.3525555

Mamoudan MM, Ostadi A, Pourkhodabakhsh N, Fathollahi-Fard AM, Soleimani F (2023) Hybrid neural network-based metaheuristics for prediction of financial markets: a case study on global gold market. J Comput Design Eng 10:1110–1125. https://doi.org/10.1093/jcde/qwad039

Mazaheri V, Khodadadi H (2020) Heart arrhythmia diagnosis based on the combination of morphological, frequency and nonlinear features of ecg signals and metaheuristic feature selection algorithm. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2020.113697

McDevitt LJS, Harrison KR, Ombuki-Berman BM (2023) A framework for meta-heuristic parameter performance prediction using fitness landscape analysis and machine learning. IEEE, Chicago, IL, USA, pp. 1–8. https://doi.org/10.1109/CEC53210.2023.10254195

McGill E, Er V, Penney T, Egan M, White M, Meier P, Whitehead M, Lock K, Cuevas RA, Smith R, Savona N, Rutter H, Marks D, Vocht F, Cummins S, Popay J, Petticrew M (2021) Evaluation of public health interventions from a complex systems perspective: a research methods review. Soc Sci Med. https://doi.org/10.1016/j.socscimed.2021.113697

Memeti S, Pllana S, Binotto A, Kołodziej J, Brandic I (2018) A review of machine learning and meta-heuristic methods for scheduling parallel computing systems. In: Proceedings of the international conference on learning and optimization algorithms: theory and applications, pp. 1–6. https://doi.org/10.1145/3230905.3230906

Moayedi H, Gör M, Khari M, Foong LK, Bahiraei M, Bui DT (2020) Hybridizing four wise neural-metaheuristic paradigms in predicting soil shear strength. Measurement. https://doi.org/10.1016/j.measurement.2020.107576

Moerland TM, Broekens J, Plaat A, Jonker CM et al (2023) Model-based reinforcement learning: a survey. Mach Learn 16(1):1–118. https://doi.org/10.1561/2200000086

Molokomme DN, Onumanyi AJ, Abu-Mahfouz AM (2024) Hybrid metaheuristic schemes with different configurations and feedback mechanisms for optimal clustering applications. Clust Comput 27(7):8865–8887. https://doi.org/10.1007/s10586-024-04416-4

Momenitabar M, Ebrahimi ZD, Ghasemi P (2022) Designing a sustainable bioethanol supply chain network: a combination of machine learning and meta-heuristic algorithms. Ind Crops Prod. https://doi.org/10.1016/j.indcrop.2022.115848

Monteiro R, Sau K (2023) Landscape-sketch-step: an ai/ml-based metaheuristic for surrogate optimization problems https://doi.org/10.48555/abs-2309.07936

Moosavi SKR, Zafar MH, Saadat A, Abaid Z, Ni W, Jamalipour A, Sanfilippo F (2024) Transductive transfer learning-assisted hybrid deep learning model for accurate state of charge estimation of li-ion batteries in electric vehicles. IEEE Trans Intell Transp Syst. https://doi.org/10.1109/TITS.2024.3403518

Morales-Hernández A, Nieuwenhuyse IV, Gonzalez SR (2023) A survey on multi-objective hyperparameter optimization algorithms for machine learning. Artif Intell Rev 56:8043–8093. https://doi.org/10.1007/s10462-022-10359-2

Msika S, Quintero A, Khomh F (2019) Sigma: strengthening ids with gan and metaheuristics attacks. arXiv preprint https://doi.org/10.1109/ICIMP49002.2021.00011

Navazi F, Yuan Y, Archer N (2023) An examination of the hybrid meta-heuristic machine learning algorithms for early diagnosis of type ii diabetes using big data feature selection. Healthc Anal. https://doi.org/10.1016/j.health.2023.100227

Nayak DR, Padhy N, Mallick PK, Bagal DK, Kumar S (2022) Brain tumour classification using noble deep learning approach with parametric optimization through metaheuristics approaches. Computers. https://doi.org/10.3390/computers11010010

Neelakandan S, Prakash M, Geetha BT, Nanda AK, Metwally AM, Santhamoorthy M, Gupta MS (2022) Metaheuristics with deep transfer learning enabled detection and classification model for industrial waste management. Chemosphere. https://doi.org/10.1016/j.chemosphere.2022.136046

Newell A, Shaw JC, Simon HA (1957) Empirical explorations of the logic theory machine: a case study in heuristic. In: Papers Presented at the February 26-28, 1957, western joint computer conference: techniques for reliability. IRE-AIEE-ACM ’57 (Western), pp. 218–230. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/1455567.1455605

Ng DTK, Leung JKL, Chu SKW, Qiao MS (2021) Conceptualizing ai literacy: an exploratory review. Comput Educ: Artif Intell. https://doi.org/10.1016/j.caeai.2021.100041

Ng CSW, Ghahfarokhi AJ, Amar MN, Torsæter O (2021) Smart proxy modeling of a fractured reservoir model for production optimization: implementation of metaheuristic algorithm and probabilistic application. Nat Resour Res 30:2431–2462. https://doi.org/10.1007/s11053-021-09844-2

Ng CSW, Ghahfarokhi AJ, Amar MN (2022) Well production forecast in volve field: application of rigorous machine learning techniques and metaheuristic algorithm. J Petrol Sci Eng. https://doi.org/10.1016/j.petrol.2021.109468

Ng CSW, Nait Amar M, Jahanbani Ghahfarokhi A, Imsland LS (2023) A survey on the application of machine learning and metaheuristic algorithms for intelligent proxy modeling in reservoir simulation. Comput Chem Eng 170:108107. https://doi.org/10.1016/j.compchemeng.2022.108107

Ngo NT, Truong TTH, Truong NS, Pham AD, Huynh NT, Pham TM, Pham VHS (2022) Proposing a hybrid metaheuristic optimization algorithm and machine learning model for energy use forecast in non-residential buildings. Sci Rep. https://doi.org/10.1038/s41598-022-04923-7

Ni X, Hu W, Fan Q, Cui Y, Qi C (2024) A q-learning based multi-strategy integrated artificial bee colony algorithm with application in unmanned vehicle path planning. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2023.121303

O Costa PR, Rhuggenaath J, Zhang Y, Akcay, A (2020) Learning 2-opt heuristics for the traveling salesman problem via deep reinforcement learning. In: Asian conference on machine learning. PMLR, pp. 465–480. https://doi.org/10.1007/s42979-021-00779-2

Ojha VK, Abraham A, Snášel V (2017) Metaheuristic design of feedforward neural networks: a review of two decades of research. Eng Appl Artif Intell 60:97–116. https://doi.org/10.1016/j.engappai.2017.01.013

Oliva D, Houssein EH, Hinojosa S (2021) Metaheuristics in machine learning: theory and applications. Springer, Berlin

Panda N, Majhi SK (2021) Effectiveness of swarm-based metaheuristic algorithm in data classification using pi-sigma higher order neural network. In: Panigrahi CR, Pati B, Mohapatra P, Buyya R, Li K-C (eds) Progress in advanced computing and intelligent engineering. Springer, Singapore, pp 77–88

Pandey A, Pandey VK (2025) Wavelet based classification using meta-heuristic algorithm with deep transfer learning technique. SN Comput Sci 6(3):208. https://doi.org/10.1007/s42979-025-03683-1

Pandey RK, Aggarwal S, Nath G, Kumar A, Vaferi B (2022) Metaheuristic algorithm integrated neural networks for well-test analyses of petroleum reservoirs. Sci Rep. https://doi.org/10.1038/s41598-022-21075-w

Paryani S, Neshat A, Pourghasemi HR, Ntona MM, Kazakis N (2022) A novel hybrid of support vector regression and metaheuristic algorithms for groundwater spring potential mapping. Sci Total Environ. https://doi.org/10.1016/j.scitotenv.2021.151055
Passos LA, Papa JP (2020) A metaheuristic-driven approach to fine-tune deep Boltzmann machines. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2019.105717
Pattanashetty PT, Pankaj, P, Kinger R, Sayed R, Arya A, Sharma R (2024) Enhancing fine-tuning of pre-trained language models for sentiment analysis using metaheuristic algorithms. In: 2024 11th International conference on soft computing & machine intelligence (ISCMI). IEEE, pp. 234–241. https://doi.org/10.1109/ISCMI63661.2024.10851676
Pérez-Sala L, Curado M, Tortosa L, Vicent JF (2023) Deep learning model of convolutional neural networks powered by a genetic algorithm for prevention of traffic accidents severity. Chaos, Solitons Fract 169:113245. https://doi.org/10.1016/j.chaos.2023.113245
Petrović A, Bacanin N, Zivkovic M, Marjanovic M, Antonijevic M, Strumberger I (2022) The adaboost approach tuned by firefly metaheuristics for fraud detection. In: 2022 IEEE world conference on applied intelligence and computing (AIC), pp. 834–839. https://doi.org/10.1109/AIC55036.2022.9848902
Pham TN, Tran LV, Dao SVT (2020) Early disease classification of mango leaves using feed-forward neural network and hybrid metaheuristic feature selection. IEEE Access 8:189960–189973. https://doi.org/10.1109/ACCESS.2020.3031914
Połap D, Woźniak M (2021) Meta-heuristic as manager in federated learning approaches for image processing purposes. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2021.107872
Poyatos J, Molina D, Martinez AD, Del Ser J, Herrera F (2023) Evoprunedeeptl: an evolutionary pruning model for transfer learning based deep neural networks. Neural Netw 158:59–82. https://doi.org/10.1016/j.neunet.2022.04.005
Premkumar M, Sinha G, Ramasamy MD, Sahu S, Subramanyam CB, Sowmya R, Abualigah L, Derebew B (2024) Augmented weighted k-means grey wolf optimizer: an enhanced metaheuristic algorithm for data clustering problems. Sci Rep 14(1):5434. https://doi.org/10.1038/s41598-024-55619-z
Price KV (2013) Differential evolution. Handbook of optimization: from classical to modern approach. Springer, Berlin, pp 187–214
Pujari KN, Miriyala SS, Mittal P, Mitra K (2023) Better wind forecasting using evolutionary neural architecture search driven green deep learning. Expert Syst Appl 214:119063. https://doi.org/10.1016/j.eswa.2022.119063
Rajesh C, Kumar S (2022) An evolutionary block based network for medical image denoising using differential evolution. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2022.108776
Razavi-Termeh SV, Sadeghi-Niaraki A, Seo M, Choi S-M (2023) Application of genetic algorithm in optimization parallel ensemble-based machine learning algorithms to flood susceptibility mapping using radar satellite imagery. Sci Total Environ 873:162285. https://doi.org/10.1016/j.envsoft.2023.105194
Revin I, Potemkin VA, Balabanov NR, Nikitin NO (2023) Automated machine learning approach for time series classification pipelines using evolutionary optimization. Knowl-Based Syst 268:110483. https://doi.org/10.1016/j.knosys.2023.110483
Rohaninejad M, Tavakkoli-Moghaddam R, Vahedi-Nouri B, Hanzálek Z, Shirazian S (2022) A hybrid learning-based meta-heuristic algorithm for scheduling of an additive manufacturing system consisting of parallel slm machines. Int J Prod Res 60:6205–6225. https://doi.org/10.1080/00207543.2021.1987550
Rojas E, Quirós-Corella F, Jones T, Meneses E (2021) Large-scale distributed deep learning: a study of mechanisms and trade-offs with pytorch. In: Latin American high performance computing conference. Springer, pp. 177–192. https://doi.org/10.1007/978-3-031-04209-6_13
Ruddick J, Camargo LR, Putratama MA, Messagie M, Coosemans T (2025) Treec: a method to generate interpretable energy management systems using a metaheuristic algorithm. Knowl-Based Syst 309:112756. https://doi.org/10.1016/j.knosys.2024.109789
Sadeg S, Hamdad L, Remache AR, Karech MN, Benatchba K, Habbas Z (2019) Qbso-fs: a reinforcement learning based bee swarm optimization metaheuristic for feature selection. In: Rojas I, Joya G, Catala A (eds) Advances in computational intelligence. Springer, Cham, pp 785–796
Sadrossadat E, Basarir H, Karrech A, Elchalakani M (2022) Multi-objective mixture design and optimisation of steel fiber reinforced uhpc using machine learning algorithms and metaheuristics. Eng Comput 38:2569–2582. https://doi.org/10.1007/s00366-021-01403-w
Saha A, Rajak S, Saha J, Chowdhury C (2022) A survey of machine learning and meta-heuristics approaches for sensor-based human activity recognition systems. J Ambient Intell Humaniz Comput. https://doi.org/10.1007/s12652-022-03870-5
SaiSindhuTheja R, Shyam GK (2021) An efficient metaheuristic algorithm based feature selection and recurrent neural network for dos attack detection in cloud computing environment. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2020.106997

Sakhnini J, Karimipour H, Dehghantanha A (2019) Smart grid cyber attacks detection using supervised learning and heuristic feature selection. In: 2019 IEEE 7th international conference on smart energy grid engineering (SEGE), pp. 108–112. https://doi.org/10.1109/SEGE.2019.8859946

Sangjinda K, Kumar DR, Keawsawasvong S, Wipulanusat W, Jamsawang P (2024) Novel neural network-based metaheuristic models for the stability prediction of rectangular trapdoors in anisotropic and non-homogeneous clay. Adv Eng Softw 193:103668. https://doi.org/10.1016/j.advengsoft.2024.103668

Santos A, Madureira A, Varela M (2018) The influence of problem specific neighborhood structures in metaheuristics performance. J Math 2018(1):8072621. https://doi.org/10.1155/2018/8072621

Saraswat M, Tyagi N (2023) Optimum feature selection using meta-heuristic algorithms. In: International conference on communication and intelligent systems. Springer, pp. 447–455. https://doi.org/10.1007/978-981-97-2082-8_31

Scherer ME, Hill RR, Lunday BJ, Cox BA, White ED (2024) Applying instance space analysis for metaheuristic selection to the 0–1 multidemand multidimensional knapsack problem. Comput Oper Res 170:106747. https://doi.org/10.1016/j.cor.2024.106821

Seyyedabbasi A (2023) A reinforcement learning-based metaheuristic algorithm for solving global optimization problems. Adv Eng Softw. https://doi.org/10.1016/j.advengsoft.2023.103411

Shao Z, Shao W, Chen J, Pi D (2024) Mql-mm: A meta-q-learning-based multi-objective metaheuristic for energy-efficient distributed fuzzy hybrid blocking flow-shop scheduling problem. IEEE Trans Evol Comput. https://doi.org/10.1109/TEVC.2024.3399314

Shariati M, Davoodnabi SM, Toghroli A, Kong Z, Shariati A (2021) Hybridization of metaheuristic algorithms with adaptive neuro-fuzzy inference system to predict load-slip behavior of angle shear connectors at elevated temperatures. Compos Struct. https://doi.org/10.1016/j.compstruct.2021.114524

Sharma A, Rani R (2019) C-hmoshssa: Gene selection for cancer classification using multi-objective meta-heuristic and machine learning methods. Comput Methods Programs Biomed 178:219–235. https://doi.org/10.1016/j.cmpb.2019.06.029

Shukla AK, Singh P, Vardhan M (2019) A new hybrid wrapper tlbo and sa with svm approach for gene expression data. Inf Sci 503:238–254. https://doi.org/10.1016/j.ins.2019.06.063

Si T, Bagchi J, Miranda PBC (2022) Artificial neural network training using metaheuristics for medical data classification: An experimental study. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2021.116423

Sindhu Meena K, Suriya S (2020) A survey on supervised and unsupervised learning techniques. In: Kumar LA, Jayashree LS, Manimegalai R (eds) Proceedings of international conference on artificial intelligence, smart grid and smart city applications. Springer, Cham, pp 627–644

Song Y, Fang X, Liu B, Li C, Li Y, Yang SX (2020) A novel foraging algorithm for swarm robotics based on virtual pheromones and neural network. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2020.106156

Song Z, Tang C, Song S, Tang Y, Li J, Ji J (2023) A complex network-based firefly algorithm for numerical optimization and time series forecasting. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2023.110158

Stein N, Bäck T (2024) Llamea: A large language model evolutionary algorithm for automatically generating metaheuristics. IEEE Trans Evol Comput. https://doi.org/10.1109/TEVC.2024.3490659

Sun P, Liu H, Zhang Y, Meng Q, Tu L, Zhao J (2021) An improved atom search optimization with dynamic opposite learning and heterogeneous comprehensive learning. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2021.107140

Sung K-K, Poggio T (1998) Example-based learning for view-based human face detection. IEEE Trans Pattern Anal Mach Intell 20(1):39–51. https://doi.org/10.1109/34.655648

Szénási S, Légrádi G (2024) Machine learning aided metaheuristics: a comprehensive review of hybrid local search methods. Expert Syst Appl 258:125192. https://doi.org/10.1016/j.eswa.2024.125192

Talbi E-G (2021) Machine learning into metaheuristics: a survey and taxonomy. ACM Comput Surv (CSUR) 54(6):1–32. https://doi.org/10.1145/3459664

Tao H, Habib M, Aljarah I, Faris H, Afan HA, Yaseen ZM (2021) An intelligent evolutionary extreme gradient boosting algorithm development for modeling scour depths under submerged weir. Inf Sci 570:172–184. https://doi.org/10.1016/j.ins.2021.04.063

Tapia D, Crawford B, Soto R, Palma W, Lemus-Romani J, Cisternas-Caneo F, Castillo M, Becerra-Rozas M, Paredes F, Misra S (2021) Embedding q-learning in the selection of metaheuristic operators: the enhanced binary grey wolf optimizer case. Institute of Electrical and Electronics Engineers Inc

Tavasoli N, Rezaee K, Momenzadeh M, Sehhati M (2021) An ensemble soft weighted gene selection-based approach and cancer classification using modified metaheuristic learning. J Comput Design Eng 8:1172–1189. https://doi.org/10.1093/jcde/qwab039

Tayal A, Solanki A, Singh SP (2020) Integrated frame work for identifying sustainable manufacturing layouts based on big data, machine learning, meta-heuristic and data envelopment analysis. Sustain Cities Soc. https://doi.org/10.1016/j.scs.2020.102383

Tinós R (2020) Artificial neural network based crossover for evolutionary algorithms. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2020.106512

Tiwari S, Chanak P, Singh SK (2023) A review of the machine learning algorithms for covid-19 case analysis. IEEE Trans Artif Intell 4(1):44–59. https://doi.org/10.1109/TAI.2022.3142241
Toscano R, Lyonnet P (2010) A new heuristic approach for non-convex optimization problems. Inf Sci 180(10):1955–1966. https://doi.org/10.1016/j.ins.2009.12.028
Tran-Ngoc H, Khatir S, Ho-Khac H, Roeck GD, Bui-Tien T, Wahab MA (2021) Efficient artificial neural networks based on a hybrid metaheuristic optimization algorithm for damage detection in laminated composite structures. Compos Struct. https://doi.org/10.1016/j.compstruct.2020.113339
Truong V-H, Pham H-A, Tangaramvong S (2025) An efficient method for nonlinear inelastic truss optimization based on improved k-nearest neighbor comparison and rao algorithm. Structures 71:108158. https://doi.org/10.1016/j.eswa.2024.125192
Tunga H, Kar S, Giri D (2024) Multi-objective brainstorm optimization algorithm based on Bayesian inference learning automaton for solving cvrp with time windows. Appl Soft Comput 162:111755. https://doi.org/10.1016/j.asoc.2024.111697
Van Engelen JE, Hoos HH (2020) A survey on semi-supervised learning. Mach Learn 109(2):373–440. https://doi.org/10.1007/s10994-019-05855-6
Van Thieu N, Mirjalili S (2023) Mealpy: an open-source library for latest meta-heuristic algorithms in python. J Syst Architect 139:102871. https://doi.org/10.1016/j.sysarc.2023.102871
Veeranjaneyulu K, Lakshmi M, Janakiraman S (2024) Swarm intelligent metaheuristic optimization algorithms-based artificial neural network models for breast cancer diagnosis: emerging trends, challenges and future research directions. Arch Comput Methods Eng. https://doi.org/10.1016/j.eswa.2024.125192
Wang Y, Pan S, Li C, Yin M (2020) A local search algorithm with reinforcement learning based repair procedure for minimum weight independent dominating set. Inf Sci 512:533–548. https://doi.org/10.1016/j.ins.2019.09.059
Wang Y, Albrecht CM, Braham NAA, Mou L, Zhu XX (2022) Self-supervised learning in remote sensing: a review. IEEE Geosci Remote Sens Mag 10:213–247. https://doi.org/10.1109/MGRS.2022.3198244
Wang H, Ji C, Shi C, Yang J, Wang S, Ge Y, Chang K, Meng H, Wang X (2023) Multi-objective optimization of a hydrogen-fueled wankel rotary engine based on machine learning and genetic algorithm. Energy 263:125961. https://doi.org/10.1016/S036054422202847X
Wang J, Tang H, Lei D (2023) A q-learning artificial bee colony for distributed assembly flow shop scheduling with factory eligibility, transportation capacity and setup time. Eng Appl Artif Intell. https://doi.org/10.1016/j.engappai.2023.106230
Wang J, Chen G, Chen Y, Ye Z, Lin M, Su R, Hu N (2024) Intelligent mixture optimization for stabilized soil containing solid waste based on machine learning and evolutionary algorithms. Constr Build Mater 445:137794. https://doi.org/10.1016/j.conbuildmat.2024.137794
Wang Y, Zhang Q, Wang G-G, Cheng H (2024) The application of evolutionary computation in generative adversarial networks (gans): a systematic literature survey. Artif Intell Rev 57(7):182. https://doi.org/10.1007/s10462-024-10818-y
Wang Y, Ji M, Fu Y, Wang H, Gao K (2024) A q-learning-based meta-heuristic algorithm for open shop scheduling considering group and transportation activities. In: 2024 IEEE 20th international conference on automation science and engineering (CASE). IEEE, pp. 2918–2923. https://doi.org/10.1016/j.swevo.2024.101518
Wang Z, Schafer BC (2020) Machine leaming to set meta-heuristic specific parameters for high-level synthesis design space exploration. In: 2020 57th ACM/IEEE design automation conference (DAC), pp. 1–6. https://doi.org/10.1109/DAC18072.2020.9218674
Winch RF (1947) Heuristic and empirical typologies: a job for factor analysis. Am Sociol Rev 12:68. https://doi.org/10.2307/2086492
Wu L, Huang G, Fan J, Ma X, Zhou H, Zeng W (2020) Hybrid extreme learning machine with meta-heuristic algorithms for monthly pan evaporation prediction. Comput Electron Agric. https://doi.org/10.1016/j.compag.2019.105115
Wu D, Foong LK, Lyu Z (2022) Two neural-metaheuristic techniques based on vortex search and backtracking search algorithms for predicting the heating load of residential buildings. Eng Comput 38:647–660. https://doi.org/10.1007/s00366-020-01074-z
Wu J, Zhao G, Wang M, Xu Y, Wang N (2024) Concrete carbonation depth prediction model based on a gradient-boosting decision tree and different metaheuristic algorithms. Case Stud Constr Mater 21:03864. https://doi.org/10.1016/j.eswa.2024.125192
Xie C, Nguyen H, Bui XN, Choi Y, Zhou J, Nguyen-Trang T (2021) Predicting rock size distribution in mine blasting using various novel soft computing models based on meta-heuristics and machine learning algorithms. Geosci Front. https://doi.org/10.1016/j.gsf.2020.11.005
Xiong G, Xie X, Yuan Z, Fu X (2023) Differential evolution-based optimized hierarchical extreme learning machines for fault section diagnosis of large-scale power systems. Expert Syst Appl 233:120937. https://doi.org/10.1016/j.eswa.2023.120937

Xu W, Tu J, Xu N, Liu Z (2024) Predicting daily heating energy consumption in residential buildings through integration of random forest model and meta-heuristic algorithms. Energy 301:131726. https://doi.org/10.1016/j.energy.2024.125192

Xue Y, Chen C, Słowik A (2023) Neural architecture search based on a multi-objective evolutionary algorithm with probability stack. IEEE Trans Evol Comput 27(4):778–786. https://doi.org/10.1109/TEVC.2023.3252612

Xue Z, Yao S, Ma H, Zhang C, Zhang K, Chen Z (2023) Thermo-economic optimization of an enhanced geothermal system (egs) based on machine learning and differential evolution algorithms. Fuel 340:127569. https://doi.org/10.1016/j.fuel.2023.127569

Yang CL, Sutrisno H (2020) A clustering-based symbiotic organisms search algorithm for high-dimensional optimization problems. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2020.106722

Yang B, Wu L, Xiong J, Zhang Y, Chen L (2023) Location and path planning for urban emergency rescue by a hybrid clustering and ant colony algorithm approach. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2023.110783

Yi W, Qu R, Jiao L, Niu B (2023) Automated design of metaheuristics using reinforcement learning within a novel general search framework. IEEE Trans Evol Comput 27:1072–1084. https://doi.org/10.1109/TEVC.2022.3197298

Yin S, Jin M, Lu H, Gong G, Mao W, Chen G, Li W (2023) Reinforcement-learning-based parameter adaptation method for particle swarm optimization. Complex Intell Syst 9:5585–5609. https://doi.org/10.1007/s40747-023-01012-8

Yonar A, Yonar H (2023) Modeling air pollution by integrating anfis and metaheuristic algorithms. Model Earth Syst Environ 9:1621–1631. https://doi.org/10.1007/s40808-022-01573-6

Yu X, Zhou J (2023) A robust method based on reinforcement learning and differential evolution for the optimal photovoltaic parameter extraction. Appl Soft Comput 148:110916. https://doi.org/10.1016/j.compeleceng.2023.108966

Yuan C, Moayedi H (2020) Evaluation and comparison of the advanced metaheuristic and conventional machine learning methods for the prediction of landslide occurrence. Eng Comput 36:1801–1811. https://doi.org/10.1007/s00366-019-00798-x

Yüksel D, Kandiller L, Taşgetiren MF (2024) Q-learning guided algorithms for bi-criteria minimization of total flow time and makespan in no-wait permutation flowshops. Swarm Evol Comput 89:101617. https://doi.org/10.1016/j.swevo.2024.101617

Yun KK, Yoon SW, Won D (2023) Interpretable stock price forecasting model using genetic algorithm-machine learning regressions and best feature subset selection. Expert Syst Appl 213:118803. https://doi.org/10.1016/j.eswa.2022.118803

Zavala GR, Nebro AJ, Luna F, Coello Coello CA (2014) A survey of multi-objective metaheuristics applied to structural optimization. Struct Multidiscip Optim 49:537–558. https://doi.org/10.1007/s00158-013-0996-4

Zerouali B, Bailek N, Tariq A, Kuriqi A, Guermoui M, Alharbi AH, Khafaga DS, El-Kenawy E-SM (2024) Enhancing deep learning-based slope stability classification using a novel metaheuristic optimization algorithm for feature selection. Sci Rep 14(1):21812. https://doi.org/10.1038/s41598-024-72588-5

Zhang J, Huang Y, Wang Y, Ma G (2020) Multi-objective optimization of concrete mixture proportions using machine learning and metaheuristic algorithms. Constr Build Mater. https://doi.org/10.1016/j.conbuildmat.2020.119208

Zhang J, Huang Y, Ma G, Sun J, Nener B (2020) A metaheuristic-optimized multi-output model for predicting multiple properties of pervious concrete. Constr Build Mater. https://doi.org/10.1016/j.conbuildmat.2020.118803

Zhang P, Wu HN, Chen RP, Chan THT (2020) Hybrid meta-heuristic and machine learning algorithms for tunneling-induced settlement prediction: a comparative study. Tunnell Underground Space Technol. https://doi.org/10.1016/j.tust.2020.103383

Zhang J, Huang Y, Ma G, Nener B (2021) Mixture optimization for environmental, economical and mechanical objectives in silica fume concrete: A novel frame-work based on machine learning and a new meta-heuristic algorithm. Resour Conserv Recyc. https://doi.org/10.1016/j.resconrec.2021.105395

Zhang P, Yin ZY, Jin YF, Chan THT, Gao FP (2021) Intelligent modelling of clay compressibility using hybrid meta-heuristic and machine learning algorithms. Geosci Front 12:441–452. https://doi.org/10.1016/j.gsf.2020.02.014

Zhang H, Nguyen H, Bui XN, Pradhan B, Mai NL, Vu DA (2021) Proposing two novel hybrid intelligence models for forecasting copper price based on extreme learning machine and meta-heuristic algorithms. Resour Policy. https://doi.org/10.1016/j.resourpol.2021.102195

Zhang X, Peng H, Zhang J, Wang Y (2023) A cost-sensitive attention temporal convolutional network based on adaptive top-k differential evolution for imbalanced time-series classification. Expert Syst Appl 213:119073. https://doi.org/10.1016/j.eswa.2022.119073

Zhang X, Zhang Q, Li Y, Zhang H (2023) Modeling and optimization of photo-fermentation biohydrogen production from co-substrates basing on response surface methodology and artificial neural network integrated genetic algorithm. Biores Technol 374:128789. https://doi.org/10.1016/j.biortech.2023.04.013

Zhang T, Chen L, Wang J (2023) Multi-objective optimization of elliptical tube fin heat exchangers based on neural networks and genetic algorithm. Energy 269:126729. https://doi.org/10.1016/j.energy.2023.1238

Zhang Y, Zhou K, Tang J (2024) Harnessing collaborative learning automata to guide multi-objective optimization based inverse analysis for structural damage identification. Appl Soft Comput 160:111697. https://doi.org/10.1016/j.asoc.2024.111697

Zhao Y, Hu H, Song C, Wang Z (2022) Predicting compressive strength of manufactured-sand concrete using conventional and metaheuristic-tuned artificial neural network. Measurement. https://doi.org/10.1016/j.measurement.2022.110993

Zhao F, Liu Y, Zhu N, Xu T (2023) Jonrinaldi: a selection hyper-heuristic algorithm with q-learning mechanism. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2023.110815

Zhao F, Wang Q, Wang L (2023) An inverse reinforcement learning framework with the q-learning mechanism for the metaheuristic algorithm. Knowl-Based Syst. https://doi.org/10.1016/j.knosys.2023.110368

Zhao F, Wang Z, Wang L, Xu T, Zhu N (2023) Jonrinaldi: a multi-agent reinforcement learning driven artificial bee colony algorithm with the central controller. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2023.119672

Zheng K, Zhang Q, Peng L, Zeng S (2023) Adaptive memetic differential evolution-back propagation-fuzzy neural network algorithm for robot control. Inf Sci 637:118940. https://doi.org/10.1016/j.fss.2023.02.002

Zhou B, Zhao Z (2023) An adaptive artificial bee colony algorithm enhanced by deep q-learning for milk-run vehicle scheduling problem based on supply hub. Knowl-Based Syst. https://doi.org/10.1016/j.knosys.2023.110367

Zhou G, Moayedi H, Foong LK (2021) Teaching-learning-based metaheuristic scheme for modifying neural computing in appraising energy performance of building. Eng Comput 37:3037–3048. https://doi.org/10.1007/s00366-020-00981-5

Zhou J, Qiu Y, Armaghani DJ, Zhang W, Li C, Zhu S, Tarinejad R (2021) Predicting tbm penetration rate in hard rock condition: a comparative study among six xgb-based metaheuristic techniques. Geosci Front. https://doi.org/10.1016/j.gsf.2020.09.020

Zhou J, Qiu Y, Zhu S, Armaghani DJ, Li C, Nguyen H, Yagiz S (2021) Optimization of support vector machine through the use of metaheuristic algorithms in forecasting tbm advance rate. Eng Appl Artif Intell. https://doi.org/10.1016/j.engappai.2020.104015

Zhu F, Wang X, Huang C, Yang Z, Chen X, Alhammadi A, Zhang Z, Yuen C, Debbah M (2024) Robust beamforming for ris-aided communications: gradient-based manifold meta learning. IEEE Trans Wirel Commun. https://doi.org/10.48550/arXiv.2402.10626

Zhu Q, Wu X, Lin Q, Chen W-N (2024) Two-stage evolutionary reinforcement learning for enhancing exploration and exploitation. Proc AAAI Conf Artif Intell 38:20892–20900. https://doi.org/10.1609/aaai.v38i18.30079

Zhuang F, Qi Z, Duan K, Xi D, Zhu Y, Zhu H, Xiong H, He Q (2021) A comprehensive survey on transfer learning. Proc IEEE 109(1):43–76. https://doi.org/10.1109/JPROC.2020.3004555

Authors and Affiliations

Ruining Zhang[1] · Jian Wang[2] · Chanjuan Liu[1] · Kaile Su[3] · Hisao Ishibuchi[4] · Yaochu Jin[5]

✉ Chanjuan Liu
chanjuanliu@dlut.edu.cn

Ruining Zhang
dgjxzrn@163.com

Jian Wang

wangjiannl@upc.edu.cn

Kaile Su
k.su@griffith.edu.au

Hisao Ishibuchi
hisao@sustech.edu.cn

Yaochu Jin
jinyaochu@westlake.edu.cn

[1] School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China

[2] College of Science, China University of Petroleum (East China), Qingdao 266580, China

[3] School of Information and Communication Technology, Griffith University, Brisbane 4111, Australia

[4] Guangdong Provincial Key Laboratory of Brain-Inspired Intelligent Computation, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China

[5] School of Engineering, Westlake University, Hangzhou 310024, China