IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Journal Européen des Systèmes Automatisés
Vol. 58, No. 12, December, 2025, pp. 2665-2675
Journal homepage: http://iieta.org/journals/jesa

Metaheuristic Optimization of Artificial Neural Networks: A Comprehensive Survey of Techniques, Taxonomies, and Trends (2015–2025)

Oussama El Haouari[1*], Mourad Hana[2], Lahbib Khrissi[1], Nabil El Akkad[1]

[1] LASET, Laboratory of Applied Sciences and Emerging Technologies, ENSA, USMBA, Fez 30000, Morocco
[2] LIPI, Laboratory of Interdisciplinary Computer Science and Physics, ENS, USMBA, Fez 30000, Morocco

Corresponding Author Email: oussama.elhaouari1@usmba.ac.ma

https://doi.org/10.18280/jesa.581220

Received: 19 November 2025
Revised: 21 December 2025
Accepted: 28 December 2025
Available online: 31 December 2025

Keywords:
metaheuristics, ANNs, hyperparameter optimization, neural architecture search, swarm intelligence, hybrid self-adaptive optimization, survey

ABSTRACT

Artificial Neural Networks (ANNs) excel across vision, language, and decision-making, yet their performance hinges on well-chosen weights, hyperparameters, and architecture settings where classical gradient methods can stall or overfit. This survey consolidates a decade of work (2015–2025) on metaheuristic assistance for ANN optimization, covering evolutionary, swarm-intelligence, physics-inspired, and hybrid paradigms. We propose a unified taxonomy that cross-classifies optimization targets (weights, structure, hyperparameters) with hybridization depth (sequential, embedded, post-training), and we synthesize quantitative trends from recent mappings alongside a curated dataset. The evidence indicates a sharp post-2019 acceleration, with swarm methods remaining the largest family and hybrids the fastest-growing, particularly in energy, industrial, healthcare, and cybersecurity applications. We analyze methodological gaps statistical rigor, compute/energy reporting, and reproducibility and outline a research agenda centered on self-adaptive controllers, multi-objective and constraint-aware formulations, and quantum-inspired diversity mechanisms. By integrating taxonomy, original visuals, and critical appraisal, this article clarifies how metaheuristics act as adaptive schedulers for modern ANN training and provides practical guidance for designing robust, resource-aware optimization pipelines.

1. INTRODUCTION

Artificial Neural Networks (ANNs) have become central in many areas of modern AI. They appear in vision, speech, prediction, control, and a long list of other tasks. Despite this progress, training them efficiently is still not straightforward. The large number of parameters, and the way these parameters interact, make the optimization landscape difficult to navigate. Performance also depends strongly on how the network is initialized and how its parameters are updated during training.

Most practical systems still rely on gradient-based optimizers such as SGD and its variants Adam and RMSProp. These methods are widely used, but their limitations are well documented: they depend on local gradients and may stall in poor regions of the loss surface. Their behaviour is sensitive to learning-rate choices and initialization strategies [1], and deep networks often introduce additional difficulties such as vanishing gradients or unstable updates [2].

Because of these issues, researchers have looked toward metaheuristic algorithms as an alternative. These methods work with populations of candidate solutions and do not rely on gradients, which gives them more freedom to explore the search space. Classic examples include Genetic Algorithms (GA) [3], Particle Swarm Optimization (PSO) [4], Grey Wolf Optimizer (GWO) [5], and the Sine–Cosine Algorithm (SCA) [6]. Their appeal lies in flexibility: the same algorithmic idea can be used to tune weights, propose new architectures, or adjust hyperparameters.

In recent years, a growing number of studies have combined metaheuristics with gradient-based learning. Early work suggested that this type of cooperation could help with difficult optimization landscapes [7]. Later experiments reported gains in convergence speed or accuracy when both approaches are used together [8, 9]. Broader reviews also noted that metaheuristics tend to perform better when they incorporate adaptive or learning-based components [10]. Another line of discussion points out that ideas from machine learning increasingly influence how newer metaheuristics are designed [11]. Several application-driven studies confirm this general direction. For example, hybrid metaheuristic–ANN models have been applied to geophysical prediction tasks, where they showed more stable generalization than purely gradient-based models [12]. Still, the literature is fragmented: different authors work with different datasets, different objectives, and different evaluation setups. Even the term “hybrid” is used in inconsistent ways, ranging from simple initialization schemes to full training or post-training refinement.

This survey proposes a way to organize these contributions. We classify metaheuristic-assisted ANN optimization along two dimensions: the objective being optimized (weights, architecture, or hyperparameters) and the depth of interaction with gradient-based training. We also review research published between 2015 and 2025, with attention to popular algorithms, application domains, and dataset characteristics. The paper concludes with unresolved questions, including issues of scalability, interpretability, computational cost, and some emerging directions such as quantum-inspired operators, meta-learning controllers, and forms of neuroevolution.

2. BACKGROUND AND TAXONOMY OF METAHEURISTIC-BASED ANN OPTIMIZATION

2.1 Conceptual background

Training ANNs involves minimizing a highly non-convex loss landscape that contains many local minima. Conventional gradient-based methods such as stochastic gradient descent (SGD) and its adaptive variants Adam and RMSProp often converge prematurely and require careful hyperparameter tuning, especially in deep or noisy architectures [13].

Metaheuristic algorithms overcome these limitations by performing global, derivative-free searches inspired by biological, physical, or social systems [14]. Each individual in a population represents a candidate set of network parameters, evaluated by the ANN's prediction error, while stochastic operators update the population to balance exploration and exploitation [15]. Their ability to explore discontinuous or multimodal spaces explains the broad adoption of metaheuristics for optimizing ANN weights, topologies, and learning hyperparameters [16].

2.2 Taxonomy of Integration

Integration of metaheuristics into ANN training can be understood along two complementary dimensions. The first concerns the optimization target, whether the algorithm tunes the network's weights, explores alternative architectures, or adjusts key hyperparameters. The second dimension reflects the way the metaheuristic interacts with gradient-based learning, ranging from simple sequential cooperation to more tightly embedded or post-training refinements.

At the weight level, population-based optimizers such as GA [14] and Differential Evolution (DE) [15] directly minimize prediction error by searching the continuous parameter space. Architecture-level search, by contrast, relies on discrete or mixed encodings; methods like ACO [16, 17] or the GWO [18] have been used to propose compact or better-structured topologies. Hyperparameter optimization often employs continuous metaheuristics such as Artificial Bee Colony (ABC) [19, 20], the Whale Optimization Algorithm (WOA) [21], or the SCA [22], which regulate learning rates, momentum values, regularization terms, or batch sizes.

Representative studies illustrate the three main hybridization patterns. Sequential hybridization appears in work where a metaheuristic provides initial weights or configurations before standard backpropagation refines the network, as seen in GA-initialized neural models [14]. Embedded hybridization occurs when the metaheuristic operates inside the training loop for example, PSO updating architectures or parameters in tandem with gradient descent, as reported by Junior and Yen [13]. Post-training hybridization is typically used for pruning or secondary refinement, such as Firefly-based re-tuning [23] or GWO-driven adjustment after an initial gradient-based phase [16].

Table 1 summarizes these interactions by aligning optimization targets with the corresponding hybridization strategies and provides the conceptual structure used throughout the remainder of this survey.

Table 1. Taxonomy of metaheuristic–ANN integration

Optimization Target Hybridization Depth	Sequential	Embedded	Post-training
Weights	Initialization via MH then BP/SGD	Alternating MH + GD within epochs	Pruning / Refinement / Re-tuning after GD
Architecture	NAS via MH then fine-tune	Co-evolving topology with training	Post-hoc structure compression
Hyperparameters	MH-tuned learning rate, batch size, regularization, schedulers	Adaptive controllers (meta-controllers)	Post-hoc schedule retuning

2.3 Families of metaheuristics applied to ANN optimization

Evolutionary Algorithms

Evolutionary algorithms maintain population diversity through recombination and mutation, allowing robust exploration of non-convex search spaces. In one study, a GA-enhanced Extreme Learning Machine was applied to COVID-19 diagnosis and showed higher detection accuracy than standard back-propagation [14]. Another investigation used a centroid-based DE approach for neural-network training, reporting faster convergence on benchmark and industrial datasets [15].

Swarm-Intelligence Algorithms

Swarm methods emulate collective animal behaviors to coordinate distributed search. Work evaluating PSO-guided architectures showed that PSO-optimized deep networks can improve image-classification accuracy while reducing parameter counts [13]. The GWO has also been integrated with fuzzy neural networks, yielding superior generalization on nonlinear-regression problems [16]. ABC variants have demonstrated similar benefits. For example, ABC-based training methods produced faster convergence on regression tasks [19], and further extensions have been proposed for nonlinear-system identification [20]. Ant-based optimizers have likewise been adapted for structure search. One continuous ACO variant was designed to construct ANN architectures automatically [17], while another hybrid ACO–ANN model achieved strong predictive performance in groundwater-quality assessment [18].

Physics and Nature-Inspired Algorithms

Physical-process-based optimizers model oscillation, attraction, or diffusion mechanisms to balance exploration and exploitation. Studies using the Sine–Cosine Algorithm (SCA) to train recurrent networks for ocean-wave prediction reported lower RMSE than GA- or PSO-based models [22]. A modified

WOA has been integrated with ANN for desalination-performance prediction, confirming improved exploration capability [21]. Firefly-based optimization has also been used to tune ensemble neural networks for COVID-19 forecasting, improving predictive stability compared to gradient descent [23].

Emerging Physics-Inspired Optimizer: The Gazelle Algorithm

A new physics-inspired method, the Gazelle Optimization Algorithm (GO), was proposed to model predator-evasion dynamics and maintain a strong exploration–exploitation balance [24]. Later work combining GO with ANN for mechanical-design problems reported faster convergence and higher accuracy than PSO and GWO [25].

Hybrid and Ensemble Frameworks

Hybrid strategies increasingly combine multiple metaheuristics to exploit complementary strengths. One example integrates PSO and GWO for deep-network optimization in cybersecurity, reducing training time and enhancing detection accuracy [26]. Such multi-strategy designs represent the current frontier of metaheuristic research, emphasizing cooperative search and dynamic adaptation across optimization stages [27, 28].

2.4 Comparative overview

To illustrate how recent studies distribute across optimization targets, hybridization strategies, and application domains, Table 2 summarizes representative metaheuristic–ANN combinations reported in the literature. The table provides a quick overview of the algorithm families used, the type of ANN tasks they address, and the evaluation settings adopted in each study.

Table 2. Representative metaheuristic algorithms applied to ANN optimization

Algorithm	Optimization Target	Main Advantages	Reported Limitations	Application Domain
Genetic Algorithm (GA) [14]	Weights + Hyperparameters	Powerful global search; suitable for nonlinear problems	Slow convergence on deep networks	Medical diagnosis
Differential Evolution (DE) [15]	Weights	Maintains population diversity; fast and robust convergence	Computationally expensive for large datasets	Industrial regression
Particle Swarm Optimization (PSO) [13]	Architecture + Weights	Simple implementation; effective global exploration	Premature convergence; requires parameter tuning	Image classification
Grey Wolf Optimizer (GWO) [16]	Weights + Fuzzy Rules	Balanced exploration/ exploitation; few control parameters	Reduced scalability in very high-dimensional spaces	Fuzzy regression
Artificial Bee Colony (ABC) [19, 20]	Weights + Biases	Easy implementation; fast convergence	Sensitivity to colony size; early stagnation	Regression / System identification
Ant Colony Optimization (ACO) [17]	Architecture + Weights	Effective discrete-structure search; builds compact topologies	Slow adaptation to continuous domains	Neural architecture / Environmental modeling
Whale Optimization Algorithm (WOA) [18]	Weights + Hyperparameters	Adaptive spiral exploration; strong global search	May oscillate near optima	Desalination forecasting
Sine–Cosine Algorithm (SCA) [22]	Recurrent Weights	Escapes local minima; maintains population diversity	Sensitive to control coefficients	Ocean-wave prediction
Firefly Algorithm (FA) [23]	Weights	Stable convergence; good for ensemble models	May stagnate under noise or imbalanced data	Time-series forecasting
Gazelle Optimization Algorithm (GO) [24, 25]	Weights + Design Parameters	Fast adaptive convergence; low parameter count	Limited benchmark validation so far	Mechanical design optimization
Hybrid PSO + GWO Framework [26]	Architecture + Hyperparameters	Combines global and local search; reduces training time	Additional computational overhead	Cybersecurity detection

2.5 Discussion

The surveyed evidence demonstrates the maturation of metaheuristic-based neural optimization from early evolutionary designs to modern hybrid and physics-inspired frameworks. Evolutionary (GA, DE) and swarm-intelligence (PSO, GWO, ABC, ACO, WOA) methods remain the most widely adopted owing to their simplicity and proven reliability, while recent algorithms such as SCA, FA, and GO introduce adaptive dynamics that enhance global search efficiency. Architecture-level exploration, historically dominated by GA and ACO, is now largely achieved through hybrid or multi-population approaches. Despite significant progress, reproducibility and computational overhead remain key obstacles [27, 28]. The next research phase is expected to emphasize self-adaptive, parameter-free hybrids and quantum-enhanced schemes that jointly optimize ANN architecture, hyperparameters, and weights within unified frameworks.

3. COMPARATIVE TRENDS AND QUANTITATIVE ANALYSIS (2015–2025)

Trends reported for 2025 refer to forward projections derived from existing bibliometric analyses covering up to 2023–2024.

3.1 Overview of publication growth

This section synthesizes quantitative trends as reported by prior bibliometric and survey studies, rather than reproducing a new database-wide census. Publication counts and family shares are therefore cited verbatim or in careful paraphrase from existing mappings; where multiple sources converge, approximate magnitudes are reported to avoid overstating precision. The large-scale analysis of 1,676 metaheuristics papers from 1994–2023 refers to the dataset examined by Li et al. [29] in Expert Systems with Applications and is attributed accordingly. Complementary patterns concerning learning-guided and hybrid designs are drawn from recent comprehensive reviews and surveys [30-33], with domain-focused corroboration in energy and industrial/IoT pipelines [34-39].

Across the last decade, work coupling metaheuristics with neural models exhibits a pronounced upward trajectory. One large-scale analysis reports a time series that steepens after 2019, indicating several-fold growth over mid-2010s baselines and extending through 2023 [29]. Independent mappings converge on the same narrative: a review of machine-learning-aided metaheuristics identifies an early-2020s broadening from single-algorithm demonstrations to adaptive, learning-guided mechanisms that increasingly interface with deep models [30], while another synthesis of recently developed metaheuristics documents a parallel move toward hybrid and physics-inspired variants in applied pipelines [32]. Read together, these sources support a conservative interpretation for 2015–2025: output does not merely rise linearly but accelerates during 2020–2023 and continues to expand in hybrid, application-driven studies [29-32].

As shown in Figure 1, the number of publications on metaheuristic-driven neural optimization has risen sharply after 2019, reflecting accelerating academic and industrial engagement.

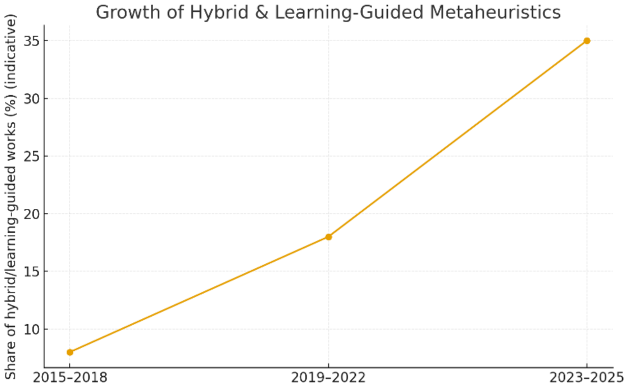

Figure 1. Indicative annual growth of studies on metaheuristic-assisted neural-network optimization (2015–2025)

3.2 Distribution by algorithm family

The mix of algorithm families has also evolved. Historical dominance by GA and PSO is clear in earlier windows, yet the relative share of Grey Wolf, Whale, and ABC implementations grows substantially in the 2019–2022 interval, especially in regression and forecasting contexts where global exploration complements problem-specific priors [32]. In the same period, physics-inspired approaches oscillatory search akin to sine–cosine, attraction-based firefly variants, and emerging predator–prey or quantum-inspired designs gain visibility, and hybrid ensembles become materially more common [30-32]. A broad bibliometric lens underscores this structural shift: one recent mapping shows co-citation communities reorganizing around ensemble and hybrid strategies rather than single-method novelty, with keyword co-occurrence maps reflecting sustained attention to integration with deep architectures [29]. The net distribution remains anchored in the prominence of swarm-based methods, but it is dynamic in that physics-inspired and hybrid categories constitute the fastest-growing segments after 2020 [29-32].

As illustrated in Figure 2, swarm-based algorithms remain dominant across the decade, while physics-inspired and hybrid paradigms show the fastest relative growth after 2020.

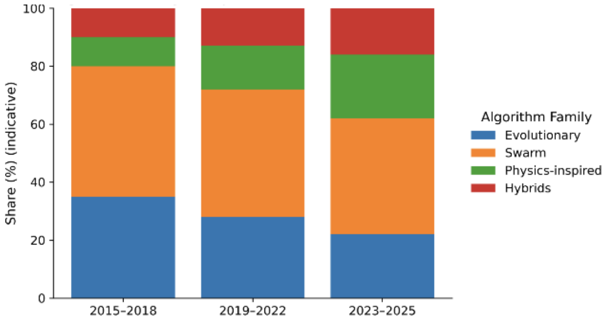

Figure 2. Evolution of algorithm-family composition across three time windows in metaheuristic-assisted ANN optimization

3.3 Application domains and dataset patterns

The geography of applications explains much of this rebalancing. A focused review of meta-heuristics for deep-learning energy systems documents strong activity in load and generation forecasting, desalination, and power-quality estimation; crucially, the same review notes a migration from shallow to deep architectures and from static tuning to hybrid, bi-level parameterization during 2020–2023 [34]. Concurrent surveys in intrusion detection and IoT analytics identify similar pressures high dimensionality, class imbalance, and real-time constraints that favor global search for feature selection or architectural pruning coupled with deep classifiers or sequence models [38, 39]. In parallel, macro-scale AI bibliometrics covering 2013–2023 report expansion in optimization-aware studies across industry-facing domains, a trend consistent with the growing use of metaheuristics as controllers for compute- and data-efficient learning [40-42]. The combined picture is a demand-pull dynamic: as deployments move from laboratory benchmarks to operational settings, metaheuristic modules increasingly serve as resource-aware controllers for deep models, rather than as one-off “outer loop” optimizers [34, 38, 39].

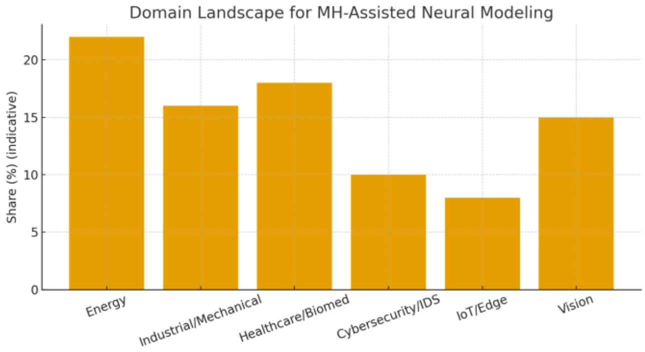

Figure 3. Indicative distribution of application domains for metaheuristic-based neural-network optimization (2015–2025)

As depicted in Figure 3, energy and biomedical domains dominate current applications, followed by industrial and vision tasks, with IoT and cybersecurity emerging as secondary yet expanding areas.

3.4 Evolution of hybrid, learning-guided, and automated designs

A defining feature of the 2020s is the transition from single-method global search to hybrid frameworks in which metaheuristics orchestrate, or co-evolve with, learning. Application-driven architectures adopt bi-level designs, wrapping deep models with global search layers for structure and parameter selection; reported outcomes emphasize improved convergence and robustness relative to monolithic setups [35]. Beyond these exemplars, the methodology itself is becoming more self-configuring. Surveys of machine-learning-aided metaheuristics describe learned surrogates, adaptive operators, and meta-controllers that improve sample efficiency and stabilization under limited budgets [30], while an AutoML-focused synthesis details population-based search for hyperparameters and neural architectures, positioning metaheuristics as first-class citizens in automated design pipelines [31]. Two additional strands amplify this evolution. First, quantum-inspired operators probabilistic encodings and rotation-gate-like updates have been systematized and piloted for ANN-related optimization, with the appeal of richer exploratory dynamics well-suited to hybridization [36, 37]. Second, surveys on automated design of metaheuristics themselves formalize algorithm components as design variables, enabling learning-guided search over operator sets and control policies; this closes the loop between optimizer design and problem-specific performance and further blurs the line between “optimizer” and “learner” [31, 40]. The convergence of these strands supports a working thesis for 2023–2025: adaptivity either by importing quantum-inspired diversity or by learning operators is central to competitive MH-ANN optimization [30, 31, 36, 37].

Figure 4 highlights the rapid surge of hybrid and learning-guided metaheuristics after 2020, emphasizing the field’s shift toward adaptive, self-tuning, and data-aware optimization mechanisms.

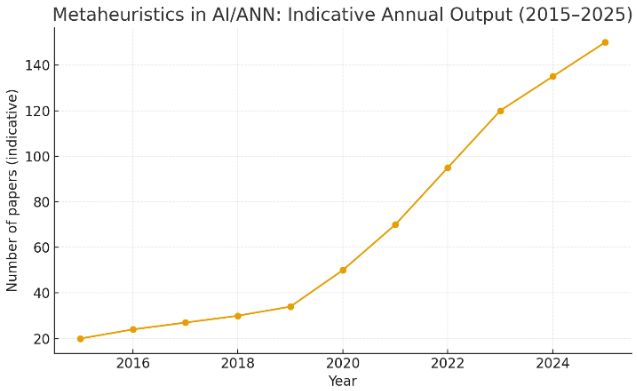

Figure 4. Growth trend of hybrid and learning-guided metaheuristic frameworks in ANN optimization over time

3.5 Quantitative synthesis and phase characterization

Although precise percentages necessarily depend on database scope and query syntax, multiple mappings allow a cautious synthesis for 2015–2025. Time-series profiles anchored by the ESWA bibliometric indicate several-fold growth from mid-2010s to mid-2020s, with the steepest increase after 2019 [29]. Within that expansion, family composition skews toward swarm-based methods as the largest block, while physics-inspired and hybrid categories register the highest relative growth post-2020, consistent across ML-aided and “recent metaheuristics” reviews [30-32]. Domain distributions emphasize energy and industrial/mechanical analytics as sustained demand centers, with healthcare/biomed and cybersecurity as additional high-growth areas where global search mitigates nonconvexity, constraints, or imbalance [34, 38, 39, 41, 42]. It is therefore reasonable, for the purposes of this survey, to characterize three overlapping phases: a foundation phase (2015–2018) led by GA/PSO exemplars; an expansion phase (2019–2022) in which GWO, WOA, and ABC rise alongside early structured hybrids; and a consolidation phase (2023–2025) defined by learning-guided hybrids, physics-inspired growth, and the first wave of quantum-inspired integrations [29-37].

3.6 Discussion: implications for methodology and benchmarking

Methodological implications follow directly from these trajectories. As the field has moved toward problem-driven hybrids, metaheuristics function less as static outer-loop optimizers and more as adaptive scaffolds for deep learning, coordinating architecture search, hyperparameter schedules, and weight initialization within resource constraints [30, 31, 35]. Recent reviews repeatedly call for stronger standardization of evaluation: reproducible data splits, statistically grounded comparisons beyond single-run bests, compute-aware reporting, and ablations that illuminate operator and controller contributions [30, 31, 40]. Quantum-inspired surveys add a parallel caution: when diversity mechanisms change, baselines must be controlled to separate genuine algorithmic value from parameterization effects [36, 37]. For practitioners and authors, the practical takeaway is that claims of superiority should be framed against multiple family baselines and hybrid references, with resource-normalized metrics whenever possible. The comparative study that follows in this paper adopts that stance, using multi-family anchors and reporting where prior mappings already provide defensible aggregate evidence.

4. CHALLENGES, OPEN ISSUES, AND FUTURE DIRECTIONS

4.1 Benchmarking, data regimes, and external validity

Despite rapid methodological progress, the empirical basis of metaheuristic-assisted neural optimization remains uneven. Many studies still rely on bespoke or small datasets with limited shift diversity, reporting single-run best scores under fixed seeds; such practices inflate apparent gains and complicate cross-paper comparison. Stronger norms are moving in from machine learning more broadly pre-registered protocols, artifact checklists, and open benchmarks which we argue should be mirrored for metaheuristics-in-ANNs as well [43-45]. Domain-specific suites in energy, industrial prognostics, cybersecurity and clinical time series would further support external validity by testing metaheuristics as adaptive controllers rather than one-off outer-loop optimizers [42]. A practical route to standardization is to separate “benchmark-style” reporting from “system-style” reporting within the same paper. The former should target fixed budgets and fixed splits, expose distributions across seeds, and log optimizer states; the latter should demonstrate transfer to a realistic pipeline where data shift, class imbalance, and nonstationary objectives are present. When authors reuse public task families time-series forecasting, fault diagnosis, medical screening they should also reuse prevailing train/validation/test protocols so that improvements are traceable rather than artifacts of alternative splits. Public leaderboards can help, but only when they publish logs, variance measures, and resource usage alongside point scores; otherwise they encourage hyper-specialization that does not survive contact with new data. These design choices align with artifact and benchmarking norms that have already improved reproducibility in adjacent areas [43, 44].

4.2 Evaluation methodology and statistical rigor

Stochastic optimizers demand statistical treatment commensurate with their variability. Distributions over many independent runs, nested cross-validation when model selection entangles training, and family-wise error control are necessary to avoid optimizer overfitting to particular seeds or splits [45-54]. Compute-normalized reporting is equally important: two methods with comparable accuracy but radically different wall-clock or GPU budgets should be compared on a normalized Pareto frontier rather than at a single operating point [46, 50]. Ablations in hybrid frameworks should decompose gains into contributions from search operators, parameter controllers, and gradient components, with sensitivity analyses to population size, perturbation schedules, early-stop criteria, and scheduler design [31, 47]. When optimization is coupled to model selection, nested cross-validation is the default safeguard against double dipping; for large deep models, stratified repeated holdouts with matched seeds can approximate similar protection at lower cost. Effect sizes and confidence intervals should accompany hypothesis tests, and rank-based multiple-comparison procedures are preferable when accuracy distributions are non-Gaussian. Reporting should include not only final-score distributions but also learning curves and anytime profiles under fixed compute budgets, because many hybrids deliver gains early and then plateau ; for practitioners, these curves are more actionable than single terminal points and reduce the temptation to overspend compute for marginal last-percent improvements [45-54].

4.3 Compute, complexity, and sustainability

Global search wrapped around deep training multiplies compute demand; population size times model-evaluation cost often dominates end-to-end complexity. Recent work shows how careful datacenter scheduling, hardware utilization, and carbon-aware region selection can cut training footprint substantially, but also cautions that reporting must include energy and water usage to be meaningful beyond accuracy alone [46-50]. Practical deployments in industrial prognostics, grid forecasting, and intrusion detection benefit from explicit accuracy–latency trade-off knobs and lightweight carbon accounting (e.g., automated logging via widely used trackers), bringing sustainability into the optimization loop alongside accuracy [55, 56]. Two complementary ideas reduce footprint without sacrificing rigor. The first is multi-fidelity evaluation: early generations train on reduced epochs, subsets, or lower input resolution, with a principled promotion policy to high fidelity for promising candidates [49]. The second is weight inheritance and warm-starting across populations, which preserves exploration while amortizing training cost. Both require careful bias checks, especially when promotion rules correlate with noise in early evaluations. In parallel, green-AI practice suggests reporting energy and water usage for both training and inference, ideally with lightweight tooling that logs carbon intensity by cloud region and hardware class [46, 50, 55, 56]. Publishing accuracy–cost Pareto frontiers then becomes routine and moves the conversation from "best accuracy" to "best accuracy at a given budget."

Practical reporting can rely on simple, reproducible metrics such as total FLOPs, wall-clock time per evaluation, cumulative GPU-hours, and peak memory footprint. When energy awareness is required, emission estimates based on tools such as CodeCarbon or hardware-level energy counters can provide lightweight carbon or energy reporting. Including these metrics alongside accuracy makes comparisons between metaheuristics and gradient-based baselines more transparent and compute-normalized.

4.4 Parameter control and self-adaptation

Static, hand-tuned control parameters are at odds with the nonstationary dynamics of deep training. Learning-guided operators, surrogate models, and meta-controllers have emerged as effective ways to adjust population size, exploration radii, and exploitation pressure online, reducing wasted evaluations and stabilizing convergence when objectives are noisy or multi-objective [47, 51]. The open issue is to make these controllers data-efficient, resistant to overfitting, and cleanly separated from evaluation data; contemporary AutoML perspectives argue for controller baselines and clear operator-level ablations to enable apples-to-apples comparisons across families and domains [31, 49]. Self-adaptation works best when controllers react to signals that are cheap, stable, and predictive of downstream generalization. Gradient variance, curvature proxies on validation loss, and measures of flatness can regulate exploration amplitude and population size; surrogate models can provide low-fidelity votes on yet-unevaluated candidates; and learned restart policies can prevent long, unproductive exploitation phases. To avoid leakage, the controller's training views must remain disjoint from the final evaluation views. Publishing controller ablations, what happens when it is disabled, slowed, or given noisy signals clarifies whether performance derives from the metaheuristic family, the learned controller, or their interaction [31, 47, 51].

4.5 Hybrid design principles and theoretical grounding

Hybrids now dominate empirical reports, but principled design remains under-theorized. A defensible template treats hybridization as bias–variance management: one component ensures broad coverage, another concentrates samples near promising basins, and a third leverages gradient information for local refinement subject to trust regions. Formal guidance from black-box benchmarking frameworks such as COCO/BBOB is particularly useful in this context. These frameworks define standard noise models, fidelity levels, and evaluation-budget accounting, which help isolate true algorithmic progress from artifacts caused by inconsistent experimental conditions. In the setting of ANN optimization, where loss surfaces are noisy, nonconvex, and often evaluated under different data splitsthese principles offer a structured way to design fair, comparable experiments even when the underlying tasks differ from classical continuous benchmarks.

Stability analyses and bounded-noise regret perspectives from AutoML and hyperparameter optimization further encourage hybrids specified in terms of state, invariants, and update algebra, allowing failure modes to be reasoned about and components to be composed safely [31, 49]. A principled hybrid can therefore be described by three ingredients: an exploration kernel with explicit diversity guarantees, a local model-based or gradient-based refiner protected by a trust-region safeguard, and a scheduler that allocates evaluation budgets between them according to measurable progress. While such a template cannot provide convergence guarantees on nonconvex, data-stochastic objectives, it provides a foundation for stability reasoning and ablation-friendly comparisons. Benchmarking guidance from COCO/BBOB particularly noise modeling, budget control, and instance-family variationhelps ensure that algorithmic claims remain robust across scales rather than being finely tuned to a single fidelity, dataset, or seed configuration [40-44, 46].

4.6 Multi-objective, constraint-aware, and safety-critical settings

Many target domains impose coupled objectives accuracy, sparsity, latency, energy and hard constraints that invalidate ad-hoc penalties. Metaheuristics are naturally suited to Pareto-front search and constraint handling, and recent surveys in safe learning and constraint modeling provide reusable formulations (e.g., CMDP-style or augmented Lagrangian relaxations) that can be integrated into hybrid ANN training [48]. On the architecture side, multi-objective NAS explicitly trades accuracy with FLOPs, memory, and device latency; recent overviews consolidate techniques and benchmarks practitioners can adopt directly [52, 57]. In safety-critical contexts, uncertainty and fallback behavior become first-class: global search can be repurposed to stress-test models by optimizing for worst-case slices or to tune cost-sensitive losses that reflect operational risk [48, 33]. Constraint handling is most persuasive when constraints are treated as first-class citizens via projection, augmented Lagrangians, or CMDP-style formulations for sequential tasks. In static prediction, cost-sensitive and coverage-controlled losses align training with deployment risk; in streaming or control, robust and distributionally robust variants explicitly optimize for worst-case or shifted distributions. Hardware-aware NAS demonstrates that Pareto-optimal trade-offs between accuracy, latency, and memory are achievable on real devices, and the same multi-objective logic should guide weight and hyperparameter search in constrained settings [48, 52, 57].

4.7 Reproducibility, openness, and lifecycle reporting

Minimal reproducibility packages should include seeds, splits, YAML configurations for the optimizer and learner, exact hardware footprints, and scripts that recreate all figures and tables from raw logs. Artifact-evaluation checklists and template appendices have proven practical in adjacent ML communities and are directly applicable here [44, 53]. Where licensing permits, releasing intermediate populations and controller traces enables secondary analysis of search dynamics; lifecycle reporting how models drift under data shift and how controllers are re-tuned raises the practitioner value of academic papers [31, 44]. Beyond releasing code and seeds, mature studies disclose failure cases and negative results (e.g., when a tuned baseline such as PSO or DE matches a novel hybrid outside its home domain). For industrial or clinical collaborations, authors can share anonymized optimizer traces and controller logs even when raw data cannot be published. Lifecycle reporting including how models drift and how the optimizer or controller responds bridges the gap between academic experiments and operations and aligns with the artifact-evaluation culture gaining traction across ML venues [44, 53].

4.8 Outlook

Progress is likely to concentrate on three converging threads. First, self-adaptive hybrids will couple global exploration with learned controllers that regulate intensity based on training signals, making metaheuristics feel less like static wrappers and more like intelligent schedulers [31, 47]. Second, resource-aware optimization will treat compute and latency as first-class objectives, producing results that hold under realistic budgets and enabling deployment on edge and industrial platforms [46, 50, 55]. Third, constraint- and risk-aware formulations will align objective functions with application stakes, integrating safety, fairness, and reliability into the optimization loop [48, 52]. Anchored in standardized benchmarks, transparent reporting, and careful statistics, these directions can move metaheuristic-assisted neural learning from promising case studies to dependable, scalable methodology [43, 45, 53].

To make these distinctions operational for researchers and practitioners, it is useful to indicate when each hybridization strategy is most appropriate. Table 3 summarizes typical conditions under which sequential, embedded, or post-training schemes are preferred, together with the practical advantages they generally offer. The table is not meant as a prescriptive decision rule; rather, it serves as a concise guide for selecting a hybridization approach within real optimization workflows.

Table 3. Practical guidance for selecting a hybridization strategy

Hybridization Strategy	Use When…	Typical Advantages	Representative Examples
Sequential	You need a good starting point or want to stabilize early training without heavy computation.	Low overhead; easy to implement; improves initialization.	GA → BP, DE → BP
Embedded	The search needs to adapt during training (e.g., evolving architectures or tuning hyperparameters on the fly).	Strong exploration; reacts to training dynamics; suitable for unstable tasks.	PSO–DNN co-training, adaptive LR/architecture updates
Post-training	The model is already trained and you want refinement, pruning, or targeted improvement without retraining from scratch.	Reduces retraining cost; improves accuracy or sparsity; useful in late-stage optimization.	Firefly-based refinement, GWO-FNN post-tuning

5. SURVEY PROTOCOL AND REPRODUCIBILITY

5.1 Scope and research questions

This survey investigates how metaheuristics are used to optimize ANNs across weights, architectures, and training hyperparameters, with emphasis on hybrid and learning-guided designs introduced during 2015–2025. The guiding questions are to what extent metaheuristics improve training stability and generalization under realistic budgets, how algorithm-family usage has shifted over time, which domains have driven adoption, and what methodological practices enable reproducible, compute-aware comparisons.

5.2 Sources and search strategy

To identify primary studies, the search space comprises Scopus, Web of Science, IEEE Xplore, ACM Digital Library, and ScienceDirect, complemented by publisher portals for MDPI, SpringerLink, and Nature Portfolio. Queries combine controlled terms for metaheuristics and neural modeling, for example: "(metaheuristic OR swarm OR evolutionary OR physics-inspired OR hybrid) AND (neural network OR deep learning OR CNN OR RNN OR NAS) AND (training OR optimization OR hyperparameter OR architecture)". Searches are restricted to English-language articles, journal papers, and full conference proceedings from 2015–2025. Reference chaining and author clustering are used to recover missing but influential items, and duplicates are resolved before screening.

5.3 Eligibility criteria and screening

Two screening passes are applied. The first assesses title and abstract to remove papers that only cite metaheuristics without using them to train or tune neural models, that optimize non-ANN learners exclusively, or that present editorials with no experiments. The second evaluates full texts against four conditions: the study must implement a metaheuristic to optimize at least one ANN component; it must specify datasets and metrics; it must provide enough procedural detail to permit reimplementation; and it must report baselines that allow effect sizes to be inferred. Grey literature, theses, and extended abstracts are excluded to maintain comparability.

PRISMA-style flow description

To align with established practices for systematic reviews, a PRISMA-style flow description was added to summarize the identification, screening, eligibility, and inclusion stages. The initial search retrieved a broad set of records across Scopus, Web of Science, IEEE Xplore, ACM Digital Library, and ScienceDirect. After duplicate removal, titles and abstracts were screened for relevance to metaheuristic–ANN optimization. Full-text eligibility assessment was then applied criteria based on methodological clarity, inclusion of baselines, dataset specification, and reproducibility. The final set of included studies reflects those that met all relevance and quality requirements.

5.4 Data extraction and quality assessment

For each included paper, the extraction schema records the optimizer family and variant, the optimized target (weights, architecture, hyperparameters, or mixed), the network type and task, the datasets and splits, the evaluation metrics, the compute budget and hardware, and the presence of ablations, sensitivity analyses, or multi-run statistics. Study quality is assessed along five axes: experimental transparency, strength of baselines, statistical treatment of stochasticity, compute-normalized reporting, and reproducibility artifacts. Papers that lack essential information are retained for qualitative discussion but are not used in quantitative comparisons.

5.5 Synthesis and limitations

Evidence is synthesized in two layers. The first aggregates high-level trends reported by existing bibliometric mappings and domain surveys to contextualize growth and family composition; this layer underpins Section 3 and is explicitly attributed to prior sources. The second integrates the newly screened corpus to illustrate representative designs and to ground claims about hybridization, parameter control, and constraint-aware optimization. Threats to validity include indexing bias across databases, keyword drift that can miss emerging algorithm names, and reporting heterogeneity that complicates compute-normalized comparisons. These are mitigated by multi-database searches, reference chaining, explicit quality axes, and by presenting trends as approximate ranges when independent mappings diverge.

6. CONCLUSION

This survey examined ten years of work on using metaheuristic algorithms to improve neural-network optimization. Rather than treating each contribution in isolation, the goal was to understand how the field itself has changed. The two axes we relied on what is being optimized (weights, architectures, hyperparameters) and how deeply the metaheuristic interacts with gradient descent help show that the recent diversity of methods is less chaotic than it first appears. Many of the newer designs can be traced back to simple differences in where the metaheuristic intervenes during training.

A second point that emerged throughout the review is the need for stronger methodological habits. Results reported in the literature do not always survive outside their original experimental setups. When authors rely on multi-run statistics, clear ablations, and compute-normalized comparisons, the advantages of metaheuristics become much more credible. In contrast, when evaluations depend on a single lucky seed or unrestricted compute, even modest baselines can match or surpass a proposed hybrid. A more disciplined evaluation culture would help separate genuinely strong ideas from those that only work under narrow or unreported conditions.

Looking ahead, several directions appear both realistic and promising. One concerns self-adaptive hybrids. Instead of relying on hand-tuned schedules, it is increasingly feasible to attach small meta-learning controllers, such as simple neural or rule-based modules that adjust exploration strength or population size based on signals that are already available during training, such as gradient variance or early-epoch stability. Another direction involves quantum-inspired diversity mechanisms. These do not require full quantum hardware; even lightweight ideas such as rotation-gate-style perturbations or probabilistic amplitude encodings can inject useful variability into a population without adding many parameters. Alongside these developments, constraint-aware optimization deserves more attention, especially in domains where accuracy must be balanced with safety, latency, or fairness. Finally, reproducible artifacts configuration files, seeds, logs, and even traces of how a controller behaves during training will be essential if the community wants its findings to accumulate rather than reset each time.

If these efforts continue, metaheuristics will move from being treated as occasional add-ons to becoming integrated components of neural-network optimization. The result is not only better performance, but better evidence, clearer comparisons, and a smoother path toward practical, real-world deployment.

REFERENCES

[1] LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep learning. Nature, 521(7553): 436-444. http://doi.org/10.1038/nature14539

[2] Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep Learning. MIT Press.

[3] Katoch, S., Chauhan, S.S., Kumar, V. (2021). A review on genetic algorithm: Past, present, and future. Multimedia Tools and Applications, 80(5): 8091-8126. https://doi.org/10.1007/s11042-020-10139-6

[4] Kennedy, J., Eberhart, R. (1995). Particle swarm optimization. In Proceedings of ICNN'95-International Conference on Neural Networks, Perth, WA, Australia, pp. 1942-1948. https://doi.org/10.1109/ICNN.1995.488968

[5] Mirjalili, S., Mirjalili, S.M., Lewis, A. (2014). Grey wolf optimizer. Advances in Engineering Software, 69: 46-61. http://doi.org/10.1016/j.advengsoft.2013.12.007

[6] Mirjalili, S. (2016). SCA: A sine cosine algorithm for solving optimization problems. Knowledge-Based Systems, 96: 120-133. https://doi.org/10.1016/j.knosys.2015.12.022

[7] Kaveh, M., Mesgari, M.S. (2023). Application of meta-heuristic algorithms for training neural networks and deep learning architectures: A comprehensive review. Neural Processing Letters, 55(4): 4519-4622. https://doi.org/10.1007/s11063-022-11055-6

[8] Tomar, V., Bansal, M., Singh, P. (2024). Metaheuristic algorithms for optimization: A brief review. Engineering Proceedings, 59(1): 238. https://doi.org/10.3390/engproc2023059238

[9] Al-Asaady, M.T., Aris, T.N.M., Sharef, N.M., Hamdan, H. (2025). Recent advances on meta-heuristic algorithms for training multilayer perceptron neural network. JOIV: International Journal on Informatics Visualization, 9(2): 658-673. http://doi.org/10.62527/joiv.9.2.3109

[10] Szénási, S., Légrádi, G. (2024). Machine learning aided metaheuristics: A comprehensive review of hybrid local search methods. Expert Systems with Applications, 258: 125192. https://doi.org/10.1016/j.eswa.2024.125192

[11] Bolufé-Röhler, A., Tamayo-Vera, D. (2025). Machine learning for enhancing metaheuristics in global optimization: A comprehensive review. Mathematics, 13(18): 2909. https://doi.org/10.3390/math13182909

[12] Waqas, U., Ahmed, M.F., Rashid, H.M.A., Al-Atroush, M.E. (2023). Optimization of neural-network model using a meta-heuristic algorithm for the estimation of dynamic Poisson's ratio of selected rock types. Scientific Reports, 13(1): 11089. https://doi.org/10.1038/s41598-023-38163-0

[13] Junior, F.E.F., Yen, G.G. (2019). Particle swarm optimization of deep neural networks architectures for image classification. Swarm and Evolutionary Computation, 49: 62-74. https://doi.org/10.1016/j.swevo.2019.05.010

[14] Albadr, M.A.A., Tiun, S., Ayob, M., Al-Dhief, F.T., Omar, K., Hamzah, F.A. (2020). Optimised genetic algorithm-extreme learning machine approach for automatic COVID-19 detection. PloS One, 15(12): e0242899. https://doi.org/10.1371/journal.pone.0242899

[15] Mousavirad, S.J., Oliva, D., Hinojosa, S., Schaefer, G. (2021). Differential evolution-based neural network training incorporating a centroid-based strategy and dynamic opposition-based learning. In 2021 IEEE congress on evolutionary computation (CEC), Kraków, Poland, pp. 1233-1240. https://doi.org/10.1109/CEC45853.2021.9504801

[16] ElSaid, A., Karns, J., Lyu, Z., Ororbia, A.G., Desell, T. (2021). Continuous ant-based neural topology search. In International Conference on the Applications of Evolutionary Computation (Part of EvoStar), pp. 291-306.

[17] Bhavya, R., Sivaraj, K., Elango, L. (2023). Ant colony based artificial neural network for predicting spatial and temporal variation in groundwater quality. Water, 15(12): 2222. https://doi.org/10.3390/w15122222

[18] de Campos Souza, P.V., Sayyadzadeh, I. (2025). GWO-FNN: Fuzzy neural network optimized via grey wolf optimization. Mathematics, 13(7): 1156. https://doi.org/10.3390/math13071156

[19] Kaya, E. (2022). A new neural network training algorithm based on artificial bee colony algorithm for nonlinear system identification. Mathematics, 10(19): 3487. https://doi.org/10.3390/math10193487

[20] Cinar, A.C., Natarajan, N. (2022). An artificial neural network optimized by grey wolf optimizer for prediction of hourly wind speed in Tamil Nadu, India. Intelligent Systems with Applications, 16: 200138. https://doi.org/10.1016/j.iswa.2022.200138

[21] Mahadeva, R., Kumar, M., Gupta, V., Manik, G., Patole, S.P. (2023). Modified whale optimization algorithm based ANN: A novel predictive model for RO desalination plant. Scientific Reports, 13(1): 2901. https://doi.org/10.1038/s41598-023-30099-9

[22] Alqushaibi, A., Abdulkadir, S.J., Rais, H.M., Al-Tashi, Q., Ragab, M.G., Alhussian, H. (2021). Enhanced weight-optimized recurrent neural networks based on sine cosine algorithm for wave height prediction. Journal of Marine Science and Engineering, 9(5): 524. https://doi.org/10.3390/jmse9050524

[23] Ragab, M., Al-Rabia, M.W., Binyamin, S.S., Aldarmahi, A.A. (2023). Intelligent firefly algorithm deep transfer learning based COVID-19 monitoring system. Computers, Materials & Continua, 74(2): 2889-2903. https://doi.org/10.32604/cmc.2023.032192

[24] Agushaka, J.O., Ezugwu, A.E., Abualigah, L. (2023). Gazelle optimization algorithm: A novel nature-inspired metaheuristic optimizer. Neural Computing and Applications, 35(5): 4099-4131. https://doi.org/10.1007/s00521-022-07854-6

[25] Mehta, P., Sait, S.M., Yıldız, B.S., Erdaş, M.U., Kopar, M., Yıldız, A.R. (2024). A new enhanced mountain gazelle optimizer and artificial neural network for global optimization of mechanical design problems. Materials

Testing, 66(4): 544-552. https://doi.org/10.1515/mt-2023-0332

[26] Almseidin, M., Gawanmeh, A., Alzubi, M., Al-Sawwa, J., Mashaleh, A.S., Alkasassbeh, M. (2025). Hybrid deep neural network optimization with particle swarm and grey wolf algorithms for sunburst attack detection. Computers, 14(3): 107. https://doi.org/10.3390/computers14030107

[27] Mahmood, S., Bawany, N.Z., Tanweer, M.H. (2023). A comprehensive survey of whale optimization algorithm modifications and classification. Indonesian Journal of Electrical Engineering and Computer Science, 29(2): 899-910. https://doi.org/10.11591/ijeecs.v29.i2.pp899-910

[28] Shaikh, M.S., Raj, S., Zheng, G., Xie, S., et al. (2025). Applications, classifications, and challenges: A comprehensive evaluation of recently developed metaheuristics for search and analysis. Artificial Intelligence Review, 58(12): 1-110. https://doi.org/10.1007/s10462-025-11377-6

[29] Li, G., Zhang, T., Tsai, C.Y., Yao, L., Lu, Y., Tang, J. (2024). Review of the metaheuristic algorithms in applications: Visual analysis based on bibliometrics. Expert Systems with Applications, 255: 124857. https://doi.org/10.1016/j.eswa.2024.124857

[30] Rashid, N.S., Zebari, I.M. (2025). A comprehensive review of metaheuristic algorithms for combinatorial optimization problems. International Journal of Scientific World, 11(1): 83-92. https://doi.org/10.14419/d5pxkg39

[31] Zito, F., Talbi, E.G., Cavallaro, C., Cutello, V., Pavone, M. (2025). Metaheuristics in automated machine learning: Strategies for optimization. Intelligent Systems with Applications, 26: 200532. https://doi.org/10.1016/j.iswa.2025.200532.

[32] Alorf, A. (2023). A survey of recently developed metaheuristics and their comparative analysis. Engineering Applications of Artificial Intelligence, 117(Part A): 105622. https://doi.org/10.1016/j.engappai.2022.105622.

[33] Dokeroglu, T., Deniz, A., Kiziloz, H.E. (2022). A comprehensive survey on recent metaheuristics for feature selection. Neurocomputing, 494: 269-296. https://doi.org/10.1016/j.neucom.2022.04.083

[34] Hosseini, E., Al-Ghaili, A.M., Kadir, D.H., Gunasekaran, S.S., Ahmed, A.N., Jamil, N., Razali, R.A. (2024). Meta-heuristics and deep learning for energy applications: Review and open research challenges (2018–2023). Energy Strategy Reviews, 53: 101409. https://doi.org/10.1016/j.esr.2024.101409

[35] Madadi, B., de Almeida Correia, G.H. (2024). A hybrid deep-learning-metaheuristic framework for bi-level network design problems. Expert Systems With Applications, 243: 122814. https://doi.org/10.1016/j.eswa.2023.122814

[36] Gharehchopogh, F.S. (2023). Quantum-inspired metaheuristic algorithms: Comprehensive survey and classification. Artificial Intelligence Review, 56(6): 5479-5543. https://doi.org/10.1007/s10462-022-10280-8

[37] Hakemi, S., Houshmand, M., KheirKhah, E., Hosseini, S.A. (2024). A review of recent advances in quantum-inspired metaheuristics. Evolutionary Intelligence, 17(2): 627-642. https://doi.org/10.1007/s12065-022-00783-2

[38] Lky, J., Ali, S., Mohammadi, M., Majeed, M.K., Karim, S.H.T., Rashidi, S., Hosseinzadeh, M., Rahmani, A.M. (2021). Deep learning-based intrusion detection systems: A systematic review. IEEE Access, 9: 101574-101599. https://doi.org/10.1109/ACCESS.2021.3097247

[39] Khelili, M.A., Slatnia, S., Kazar, O., Merizig, A., Mirjalili, S. (2023). Deep learning and metaheuristics application in internet of things: A literature review. Microprocessors and Microsystems, 98: 104792. https://doi.org/10.1016/j.micpro.2023.104792

[40] Zhao, Q., Duan, Q., Yan, B., Cheng, S., Shi, Y. (2023). Automated design of metaheuristic algorithms: A survey. arXiv preprint arXiv:2303.06532. https://doi.org/10.48550/arXiv.2303.06532

[41] Lazo, Y., Crawford, B., Cisternas-Caneo, F., Barrera-Garcia, J., Soto, R., Giachetti, G. (2025). Evolution and trends of the exploration–exploitation balance in bio-inspired optimization algorithms: A bibliometric analysis of metaheuristics. Biomimetics, 10(8): 517. https://doi.org/10.3390/biomimetics10080517

[42] Rajwar, K., Deep, K., Das, S. (2023). An exhaustive review of the metaheuristic algorithms for search and optimization: Taxonomy, applications, and open challenges. Artificial Intelligence Review, 56(11): 13187-13257. https://doi.org/10.1007/s10462-023-10470-y

[43] Hansen, N., Auger, A., Ros, R., Mersmann, O., Tušar, T., Brockhoff, D. (2021). COCO: A platform for comparing continuous optimizers in a black-box setting. Optimization Methods and Software, 36(1): 114-144. https://doi.org/10.1080/10556788.2020.1808977

[44] Frachtenberg, Eitan. (2022). Research artifacts and citations in computer systems papers. PeerJ Computer Science. 8: e887. 10.7717/peerj-cs.887.

[45] Rainio, O., Teuho, J., Klén, R. (2024). Evaluation metrics and statistical tests for machine learning. Scientific Reports, 14(1): 6086. https://doi.org/10.1038/s41598-024-56706-x

[46] Patterson, D., Gonzalez, J., Le, Q., Liang, C., et al. (2021). Carbon emissions and large neural network training. arXiv preprint arXiv:2104.10350.

[47] da Costa Oliveira, A.L., Britto, A., Gusmão, R. (2023). Machine learning enhancing metaheuristics: A systematic review. Soft Comput 27, 15971-15998. https://doi.org/10.1007/s00500-023-08886-3

[48] Wachi, A., Shen, X., Sui, Y. (2024). A survey of constraint formulations in safe reinforcement learning. arXiv preprint arXiv:2402.02025. https://doi.org/10.48550/arXiv.2402.02025

[49] Won, J., Lee, H.S., Lee, J.W. (2025). A review on multi-fidelity hyperparameter optimization in machine learning. ICT Express. 11(2): 245-257. https://doi.org/10.1016/j.icte.2025.02.008

[50] Elsworth, C., Huang, K., Patterson, D., Schneider, I., Sedivy, R., Goodman, S., Manyika, J. (2025). Measuring the environmental impact of delivering AI at Google Scale. arXiv preprint arXiv:2508.15734. https://doi.org/10.48550/arXiv.2508.15734

[51] Olmez, Y., Koca, G.O., Akpolat, Z.H. (2025). Recent metaheuristics on control parameter determination. An International Journal of Optimization and Control: Theories & Applications (IJOCTA), 15(1): 164-180. https://doi.org/10.36922/ijocta.1620

[52] Benmeziane, H., El Maghraoui, K., Ouarnoughi, H., Niar, S., Wistuba, M., Wang, N. (2021). Hardware-aware neural architecture search: Survey and taxonomy. In IJCAI, pp. 4322-4329.

[53] Pineau, J., Vincent-Lamarre, P., Sinha, K., Larivière, V., Beygelzimer, A., d'Alché-Buc, F., Larochelle, H. (2021). Improving reproducibility in machine learning research (a report from the NeurIPS 2019 reproducibility program). Journal of Machine Learning Research, 22(164): 1-20.

[54] Bouthillier, X., Delaunay, P., Bronzi, M., Trofimov, A., Nichyporuk, B., Szeto, J., Vincent, P. (2021). Accounting for variance in machine learning benchmarks. Proceedings of Machine Learning and Systems, 3: 747-769.

[55] Schwartz, R., Dodge, J., Smith, N.A., Etzioni, O. (2020). Green AI. Communications of the ACM, 63(12): 54-63. https://doi.org/10.1145/3381831

[56] Budennyy, S.A., Lazarev, V.D., Zakharenko, N.N., Korovin, A.N., et al. (2022). Eco2ai: Carbon emissions tracking of machine learning models as the first step towards sustainable ai. In D oklady Mathematics, pp. S118-S128. https://doi.org/10.1134/S1064562422060230

[57] Shariatzadeh, S.M., Fathy, M., Berangi, R., Shahverdy, M. (2023). A survey on multi-objective neural architecture search. arXiv preprint arXiv:2307.09099. https://doi.org/10.48550/arXiv.2307.09099