Computational Mathematics for AI: Numerical Methods and Distributed Computing for Deep Learning on Big Data

BEN TROVATO* and G.K.M. TOBIN*, Institute for Clarity in Documentation, USA
LARS THØRVÄLD, The Thørväld Group, Iceland
VALERIE BÉRANGER, Inria Paris-Rocquencourt, France
APARNA PATEL, Rajiv Gandhi University, India
HUIFEN CHAN, Tsinghua University, China
CHARLES PALMER, Palmer Research Laboratories, USA
JOHN SMITH, The Thørväld Group, Iceland
JULIUS P. KUMQUAT, The Kumquat Consortium, USA

A clear and well-documented LaTeX document is presented as an article formatted for publication by ACM in a conference proceedings or journal publication. Based on the "acmart" document class, this article presents and explains many of the common variations, as well as many of the formatting elements an author may use in the preparation of the documentation of their work.

1 Introduction

The exponential growth of data and the rising complexity of deep learning models have made computational efficiency a critical concern in *artificial intelligence* (AI). As a result, the convergence of *computational mathematics* and scalable AI systems has emerged as a strategic research area. In particular, the use of advanced numerical methods and distributed computing techniques is essential for training and deploying deep learning models on big data. This *systematic literature review* (SLR) investigates the state-of-the-art at this intersection, focusing on two complementary

---

*Both authors are people.

Authors' Contact Information: Ben Trovato, trovato@corporation.com; G.K.M. Tobin, webmaster@marysville-ohio.com, Institute for Clarity in Documentation, Dublin, Ohio, USA; Lars Thørväld, The Thørväld Group, Hekla, Iceland, larst@affiliation.org; Valerie Béranger, Inria Paris-Rocquencourt, Rocquencourt, France; Aparna Patel, Rajiv Gandhi University, Doimukh, Arunachal Pradesh, India; Huifen Chan, Tsinghua University, Haidian Qu, Beijing Shi, China; Charles Palmer, Palmer Research Laboratories, San Antonio, Texas, USA, cpalmer@prl.com; John Smith, The Thørväld Group, Hekla, Iceland, jsmith@affiliation.org; Julius P. Kumquat, The Kumquat Consortium, New York, USA, jpkumquat@consortium.net.

---

pillars: numerical optimization techniques (SLR1) and distributed computing strategies (SLR2) that support scalable deep learning systems.

While previous systematic reviews have explored aspects of deep learning and big data, most have adopted a narrow focus. For example, Najafabadi et al. [40] provided a high-level survey on the challenges of applying deep learning to big data, emphasizing the limitations of traditional computing infrastructure. Similarly, Zhang et al. [60] conducted a review of distributed deep learning frameworks but without integrating the perspective of computational mathematics or explicitly analyzing the underlying numerical methods used. These works underscore the importance of scalability and performance in deep learning, but they do not offer a comprehensive synthesis of the mathematical foundations and infrastructural techniques in tandem.

This review aims to bridge that gap by offering a dual-SLR framework that systematically evaluates both algorithmic and architectural components enabling deep learning at scale. The contribution of this work is threefold. First, it provides a structured taxonomy of numerical methods specifically tailored to big data deep learning contexts, with performance metrics such as computational efficiency and convergence behavior. Second, it evaluates distributed computing techniques (including federated learning, GPU acceleration, and parallel processing) in terms of scalability and system throughput. Third, it integrates findings across both domains to surface synergies, trade-offs, and research gaps that are often overlooked when these topics are studied in isolation.

To ensure transparency, rigor, and reproducibility, the review methodology adheres to the PRISMA guidelines for systematic reviews [38] and draws on Kitchenham's methodological framework for software engineering reviews [28]. It incorporates a seven-phase review process, including meta-analysis, network analysis, and a Delphi-based consensus protocol for paper inclusion and quality assessment.

In addition to synthesizing current knowledge, this work aims to guide future research by identifying methodological trends, emerging optimization paradigms, and key challenges that must be addressed to build efficient and scalable AI systems.

The paper is structured as follows: Section 2 introduces key terminology and presents the rationale and objectives of the review. Section 3 surveys prior literature on numerical methods and distributed frameworks for deep learning at scale. Section 4 details the systematic literature review methodology, including research questions, search strategy, inclusion and exclusion criteria, and quality assessment protocols. Section 5 presents the results and analysis of the selected studies for both SLR1 and SLR2. Section 7 concludes the paper with a summary of contributions and recommendations for advancing computational mathematics for AI.

2 Background

The convergence of computational mathematics and artificial intelligence has become increasingly important as deep learning models grow in complexity and are applied to massive datasets [40, 50]. Foundational techniques in numerical optimization and scalable infrastructure have emerged as the enabling pillars for performance at scale [12, 20, 31].

Historical Foundations. The use of numerical methods in machine learning can be traced back to the development of backpropagation and gradient descent, which enabled efficient training of deep neural networks [44]. Subsequent work introduced adaptive gradient methods (e.g., Adam [26]) and second-order optimizers [35] to improve convergence and generalization. In parallel, distributed computing frameworks such as MapReduce [13] and later Apache Spark [56] enabled large-scale data processing by abstracting fault tolerance and parallelism.

Numerical Optimization. Modern deep learning relies heavily on numerical optimization to train models efficiently. Optimization strategies such as stochastic gradient descent (SGD), Adam, and

RMSProp dominate practical applications [43]. However, their effectiveness often degrades in high-dimensional or non-convex settings. This has led to growing interest in alternative paradigms such as nature-inspired metaheuristics [8, 18, 23], which have been shown to perform well in healthcare and environmental modeling, and Bayesian optimization [29, 48], which is particularly suited for hyperparameter tuning under uncertainty. Yet, several studies note that while nature-inspired algorithms offer practical success, they lack rigorous convergence guarantees [55, 61], suggesting a need for more robust theoretical frameworks to support their deployment.

Distributed and Federated Computing. The challenge of scaling optimization algorithms to big data settings necessitates parallel and distributed processing architectures. Distributed training frameworks, such as Google's DistBelief [12] and Horovod [47], have demonstrated substantial speedups by splitting workloads across GPU clusters. In privacy-sensitive domains, federated learning has emerged as an effective alternative to centralized training [37]. These techniques allow training across devices while preserving data locality—critical in healthcare and finance.

Hardware-Aware and Co-Optimized Design. Recent research has emphasized hardware-aware optimization, where algorithm design is tailored to constraints such as memory usage, throughput, and energy efficiency [7]. For example, sparsity-aware training and quantization methods have enabled neural networks to be deployed on edge devices [21]. This shift reflects an increasing consensus that software and hardware co-design is essential for maximizing performance on heterogeneous systems.

Critical Gaps in Prior Reviews. While previous surveys such as Najafabadi et al. [40] highlighted the scalability challenges of deep learning, they did not investigate the mathematical principles underlying optimization. Similarly, Zhang et al. [57] presented a review of distributed deep learning architectures but failed to evaluate the performance trade-offs introduced by different optimization algorithms. More recent works Thompson et al. [50] have focused on environmental and cost implications but did not propose integrated technical strategies. As a result, there remains a disconnect between infrastructure-level scalability and algorithmic efficiency.

Rationale for This Review. This review aims to address the fragmentation in the literature by offering a dual-perspective SLR. It investigates numerical optimization techniques (SLR1) and distributed computing strategies (SLR2) as complementary components of scalable deep learning systems. The review follows PRISMA guidelines [38] and leverages Kitchenham's framework [28] to ensure methodological rigor. By synthesizing these two domains-optimization and infrastructure-this work provides a coherent foundation for understanding and designing efficient AI systems in big data contexts.

2.1 Terminology and Definitions

To ensure clarity throughout this analysis, we define the following key technical terms:

- **Computational Mathematics for AI**: The application of mathematical techniques and algorithms to solve computational problems in artificial intelligence.
- **Numerical Methods**: Algorithms that use numerical approximation for the problems of mathematical analysis.
- **Optimization**: The selection of the best element from a set of available alternatives according to some criteria.
- **Nature-Inspired Algorithms**: A class of metaheuristic optimization algorithms inspired by natural phenomena, such as biological evolution, animal behavior, or ecological processes, used to solve complex, non-convex optimization problems.

- **Bayesian Optimization**: An optimization technique that builds a probabilistic model of the objective function and uses it to select the most promising points to evaluate, aiming to find the global optimum with a minimal number of evaluations.
- **Multi-Objective Optimization**: An optimization framework where two or more conflicting objectives are optimized simultaneously, resulting in a set of trade-off solutions known as the Pareto front.
- **Hardware-Aware Optimization**: An approach to optimization that explicitly considers the computational characteristics and constraints of the target hardware platform, aiming to maximize performance, efficiency, or resource utilization.
- **Distributed Computing**: A computing paradigm where multiple computers work together to solve computational problems across a network.
- **Federated Learning**: A decentralized machine learning approach where multiple devices collaboratively train a shared model without exchanging raw data, thereby preserving privacy and reducing data transfer requirements.
- **Deep Learning**: A subset of machine learning using neural networks with multiple layers to extract higher-level features from raw input.
- **Big Data**: Data sets that are too large or complex to be dealt with by traditional data-processing application software.
- **Systematic Literature Review (SLR)**: A structured and methodical approach to identifying, evaluating, and synthesizing existing research studies on a specific topic, following predefined protocols to ensure transparency, repeatability, and minimization of bias.
- **PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses)**: A set of evidence-based guidelines and a reporting framework designed to enhance the quality and transparency of systematic reviews and meta-analyses.
- **Meta-Analysis**: A statistical technique for combining results from multiple independent studies to identify patterns, derive general conclusions, or estimate the overall effect size with greater precision.

3 Related Work

The evolution of deep learning has been marked by significant milestones in both algorithmic development and computational infrastructure Capra et al. [7], Mayer and Jacobsen [36], Thompson et al. [50]. This section presents a chronological overview of key contributions, critically analyzing their impact on scalable deep learning in big data contexts.

One of the earliest comprehensive overviews in this space was offered by Najafabadi et al. [40], who reviewed the potential of deep learning for handling high-volume, high-velocity, and high-variety data in big data analytics. They identified core challenges such as data heterogeneity, model scalability, and integration with real-time processing systems. While their work provided valuable conceptual framing, it lacked a deeper examination of optimization strategies and underlying computational systems required for deployment at scale.

A major shift occurred in 2012 with two pivotal contributions. First, Krizhevsky et al. [30] introduced a GPU-accelerated deep convolutional neural network that achieved breakthrough results on the ImageNet benchmark, emphasizing the role of hardware in scaling model performance. Shortly thereafter, Dean et al. [12] introduced the DistBelief framework, enabling distributed training of deep networks across clusters of commodity machines. Together, these works demonstrated both the necessity and feasibility of scaling deep learning through hardware and distributed system innovations.

Building on the momentum from these breakthroughs, Montavon et al. [39] published guidelines later in the same year that addressed common challenges in training deep architectures, including

vanishing gradients and unstable convergence. His work laid the groundwork for practical training strategies that remain influential in both research and industry applications.

As the field matured, more systematic evaluations of optimization techniques began to emerge. For instance, Ben and Waller [5] provided a comprehensive comparison of optimization algorithms tailored to big data scenarios. They observed that while nature-inspired methods such as genetic algorithms and particle swarm optimization performed well empirically, they often lacked theoretical convergence guarantees and posed challenges for reproducibility—underscoring a tension between empirical success and theoretical rigor.

A parallel line of research focused on Bayesian optimization as a method for hyperparameter tuning. Thoppil et al. [51] applied Bayesian strategies to train LSTM and bi-LSTM models in predictive maintenance contexts, reporting gains in efficiency and accuracy. However, they also noted that the computational demands of Bayesian methods could hinder their scalability when applied to large, high-dimensional spaces.

The need to align algorithm design with hardware realities became more pronounced in subsequent years. Capra et al. [7] argued for hardware-aware algorithmic design, reviewing approaches that integrate sparsity, quantization, and memory optimization to improve training efficiency across deep networks. Their survey emphasized that scalable deep learning must consider hardware constraints as a first-class concern.

Beyond algorithmic strategies, several studies have explored distributed computing frameworks that support deep learning at scale. Mayer and Jacobsen [36] synthesized various engineering approaches for distributed training, including model parallelism, asynchronous updates, and fault tolerance mechanisms. Their review laid the foundation for evaluating trade-offs in distributed system design. Extending this work, Yan [54] examined computational frameworks in edge and federated learning scenarios, highlighting new opportunities for decentralization while noting the scarcity of systematic benchmarking. Further refinement came from Zhang et al. [58], who compared distributed deep learning systems based on resource allocation, fault recovery, and training latency—though their work did not engage deeply with optimization techniques at the algorithmic level.

Adding a broader perspective, Thompson et al. [50] raised concerns about the economic and environmental costs associated with large-scale deep learning. Their empirical study quantified the exponential growth in compute resources needed to achieve incremental performance gains and called for a research agenda centered on efficiency-aware models.

Despite this extensive body of work, most prior reviews have addressed either optimization algorithms or distributed system frameworks in isolation. For example, Li et al. [33] presented a comprehensive survey of scalable deep learning architectures but offered limited discussion on the numerical optimization techniques critical for training. Our review addresses this fragmentation by adopting a dual SLR methodology that examines both numerical and computational enablers of deep learning at scale.

By integrating these complementary domains through a structured review protocol grounded in PRISMA guidelines [38], our work reveals cross-domain synergies, unresolved trade-offs, and critical research gaps. We argue that only by examining optimization and infrastructure in tandem can the field develop robust, scalable, and efficient deep learning systems suited for complex, high-stakes applications in healthcare, cybersecurity, and beyond.

4 Methodology

This research adopts a SLR framework to explore two complementary yet distinct domains within the field of deep learning for big data: (SLR1) numerical methods and (SLR2) distributed computing techniques. The SLR methodology is informed by the guidelines outlined by Kitchenham [27].

4.1 Phase 1: Planning and Protocol Development

4.1.1 Research Questions Formulation. The research questions address both the technical specifications and performance characteristics of deep learning systems for big data. SLR1 focuses on identifying numerical methods and evaluating their computational efficiency and accuracy:

- **RQ1.1**: What are the state-of-the-art numerical methods used in deep learning for big data?
- **RQ1.2**: How do these methods perform in terms of computational efficiency and accuracy?

SLR2 investigates distributed computing techniques and evaluates their effectiveness in scaling deep learning applications. Specifically, it addresses:

- **RQ2.1**: Which distributed computing techniques enable scaling of deep learning for big data applications?
- **RQ2.2**: What is the measured effectiveness of these techniques in terms of scalability and performance metrics?

4.1.2 Literature Review Classification Framework. We employ Cooper's taxonomy of literature reviews [9] as our classification framework (fig. 1). This approach examines studies through six critical dimensions: the focus of research (ranging from theoretical to applied), the review's goal (integration vs. critical analysis), the author's perspective (neutral or advocacy), coverage breadth, organizational approach, and intended audience.

4.1.3 Search Strategy Development. For SLR1, we construct search strings that combine "deep learning" with numerical method terminology ("numerical method", "computational mathematic", "optimization algorithm") and big data concept.

For SLR2, we use keywords including "distributed computing", "parallel processing", and "GPU acceleration" while maintaining the core deep learning and big data context. We adapt all search strings to meet each database's specific syntax requirements.

4.1.4 Information Sources and Eligibility Criteria. We conduct our search across seven major academic databases (IEEE Xplore, ACM Digital Library, SpringerLink, Scopus, Web of Science, JSTOR, AIS) to ensure coverage of both computer science and applied mathematics literature.

Inclusion Criteria:

- Studies published between 2014 and 2024
- Peer-reviewed journal articles and complete conference proceedings
- English-language publications with full text available
- Research focusing specifically on numerical methods in deep learning for big data applications
- Studies that quantitatively evaluate computational efficiency or accuracy metrics
- Papers providing either comparative analyses or empirical validation of numerical approaches

Exclusion Criteria:

- Research that does not address big data scale challenges
- Publications lacking methodological details about numerical implementations
- Secondary literature (reviews, editorials, or position papers)
- Preliminary works (short papers <10 pages, extended abstracts, or posters)
- Duplicate publications or overlapping studies

4.2 Phase 2: Literature Search and Study Selection

4.2.1 Search Execution and Initial Processing. We export the query results from all seven databases in CSV format. The initial search yields 1,306 papers across the selected databases. After removing
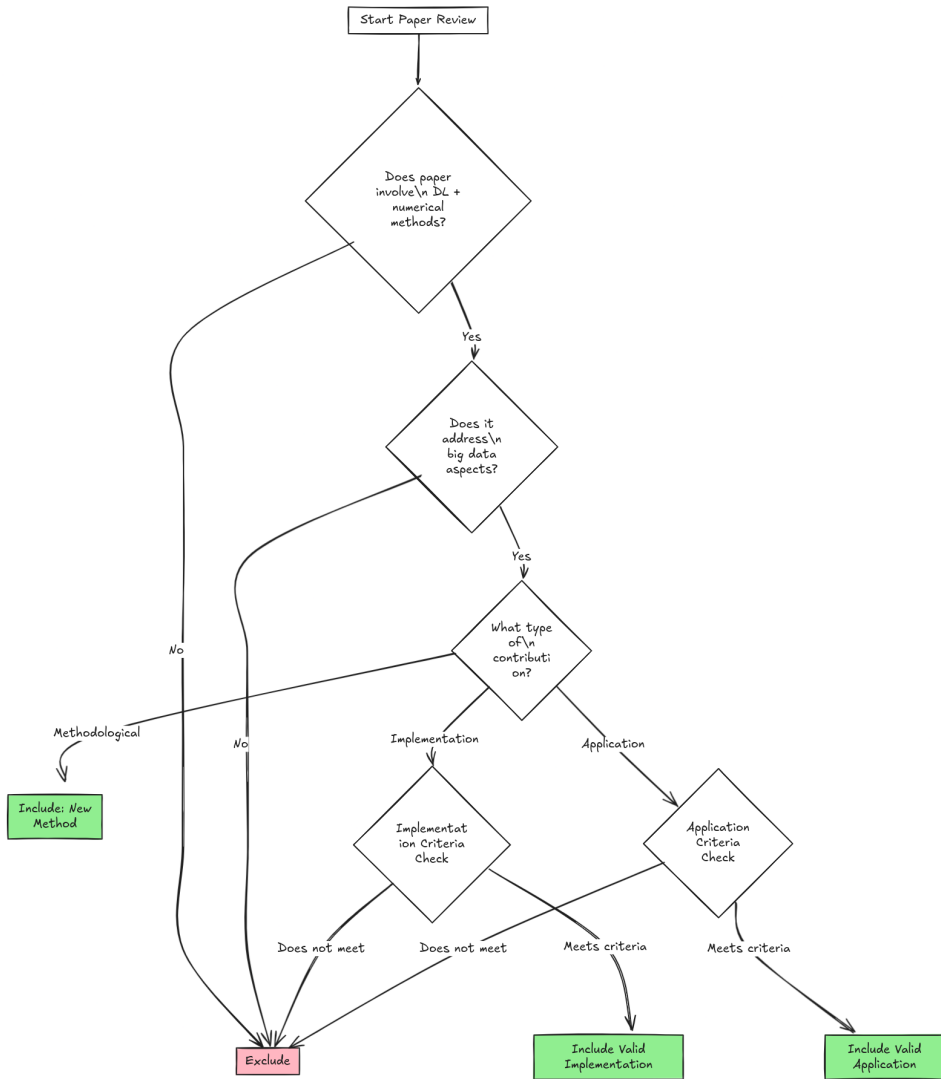
Fig. 1. Paper Classification Decision Framework

duplicates, this eliminates 33.8% of the papers demonstrating significant overlap across databases. We then upload the 66.2% of the papers to Baserow software for systematic assessment.

4.2.2 Refining Screening Through Delphi Consensus. The initial screening phase reveals unexpected challenges in inter-rater reliability, with Krippendorff's alpha measuring only 0.4 across coauthor teams. This low agreement leads to the implementation of a modified Delphi protocol that combines the RAND/UCLA methodology [17] with Dalkey's classical Delphi framework [10].

In Round 1, each coauthor independently evaluates 50 randomly selected articles, documenting detailed rationales for inclusion/exclusion decisions. The anonymous compilation of these assessments reveals three primary disagreement areas: (1) interpretation of "big data" characteristics,

(2) assessment of methodological rigor, and (3) classification of hybrid numerical-distributed approaches. Round 2 provides coauthors with statistical summaries of group responses alongside anonymized rationales for divergent evaluations and peer commentary on disagreement points.

Round 3 employs nominal group techniques [15] to operationalize consensus. Through discussion, we develop screening guidelines that address: definitions of big data characteristics (volume/velocity thresholds), minimum reporting standards for numerical methods, decision rules for hybrid approach classification.

We codify these criteria in a decision flowchart as shown in fig. 1. Following the initial screening rounds, Round 4 serves as a validation phase to verify inter-rater reliability. The decision to proceed depends on Krippendorff's α coefficient: If $\alpha \geq 0.8$ (indicating sufficient agreement), the study progresses to full screening. If $\alpha < 0.8$, the coauthors return to Round 3, conducting focused discussions to resolve persistent discrepancies before re-evaluation reliability.

4.2.3 Hierarchical Decision Framework. Following the Delphi-based consensus methodology outlined in Dalkey [10] and the systematic review guidelines in Kitchenham [27], we conduct a consensus meeting to establish classification criteria. The meeting employs the Nominal Group Technique as described in Delbecq and Van de Ven [15], which yields a formalized decision framework.

Technical Foundation Verification: The first verification gate confirms each paper's relevance to our core technical domains by requiring explicit evidence of deep learning implementation coupled with either numerical methods or distributed computing approaches.

Big Data Context Assessment: The second gate assesses whether papers demonstrate engagement with at least one of the four key big data characteristics: substantial data volume, real-time processing requirements, heterogeneous data integration, or specialized computational infrastructure.

Contribution Typology Classification: The final gate classifies articles by their primary research contribution type, distinguishing between theoretical foundations, methodological innovations, application-oriented implementations, and hybrid approaches that combine multiple contribution types.

4.2.4 Border Case Resolution and Validation. For hybrid contributions, we implement a primary contribution analysis that weights methodological innovation against domain application. When novel approaches challenge existing classification schemes, they undergo third-coauthor evaluation and potential framework adaptation, with all such cases thoroughly documented to ensure transparency.

We validate the framework through three complementary approaches. First, pilot testing on all sample papers verifies the framework's effectiveness while revealing opportunities for refinement. Second, an independent coauthor conducts external validation of our classification logic, with particular focus on edge cases. Finally, statistical validation demonstrates strong agreement rates (>85% overall) and consistent decision patterns across all coauthors.

4.3 Phase 3: Quality Assessment

4.3.1 Quality Assessment Framework. Our evaluation criteria examine four dimensions of research quality, each addressing distinct aspects of scholarly rigor and practical relevance. The first dimension establishes a minimum quality threshold by requiring studies to demonstrate systematic research methodologies rather than anecdotal evidence or untested assertions. We mandate that papers clearly articulate their research objectives and theoretical foundations while providing sufficient contextual information for readers to evaluate the study's premises and limitations. The rigor dimension evaluates the appropriateness and robustness of the research design, examining

whether the chosen methodologies align with the stated objectives and whether data collection approaches produce valid and reliable evidence.

Credibility forms our third dimension of evaluation, focusing on the transparency and objectivity of reported findings. In a field often characterized by competing optimization claims and performance benchmarks, we pay attention to whether studies present results comprehensively, acknowledging limitations and potential biases. The final relevance dimension ensures that the research contributes meaningfully to academic knowledge or practical applications.

4.3.2 Assessment Methodology. The quality assessment employs a two-phase approach to identify quality studies. Phase 1 serves an initial filter, applying only the minimum quality threshold criteria to eliminate studies lacking fundamental methodological soundness.

Papers that pass Phase 1 advance to Phase 2, where coauthors assess all remaining quality dimensions. We apply binary 'yes/no' evaluations for each criterion, requiring studies to achieve at least 75% positive assessments across all criteria for inclusion.

To ensure consistency, coauthors independently evaluate each study. We measure inter-rater agreement using Krippendorff's alpha coefficient, maintaining a target reliability threshold of $\alpha > 0.8$ to meet established standards for systematic reviews in technical domains.

4.3.3 Validation and Threshold Enforcement. Our final inclusion requirements enforce three complementary standards: perfect adherence to minimum quality thresholds, 75% positive evaluations across all criteria, and maintained inter-rater reliability.

4.4 Phase 4: Data extraction

Our extraction methodology employs qualitative analysis software from NVivo [28] to code and organize the content of the study according to a hierarchical framework with four primary dimensions. The Method dimension captures technical specifications, including detailed descriptions of numerical algorithms or distributed computing techniques (M-01), their implementation approaches (M-02), and validation methodologies (M-03). This proves to be particularly crucial for enabling comparative analysis between studies with varying technical implementations.

The context dimension documents essential environmental factors that influence research results, including problem domains (C-01), data set characteristics (C-02), and computing environments (C-03). Capturing these contextual elements allows us to situate each study's findings within its specific experimental conditions.

For the results dimension, we extract quantitative outcomes (R-01) using standardized metrics wherever possible, along with any comparative analyses (R-02) and statistical significance assessments (R-03). This structured approach to performance data enables meaningful cross-study comparisons despite variations in reporting formats. The Findings dimension then captures higher-level insights, including key contributions (F-01), acknowledged limitations (F-02), and suggested future directions (F-03), which collectively inform our gap analysis and research agenda. The codes are summarized in Table table 1.

The extraction process follows a systematic workflow in NVivo, beginning with the creation of hierarchical nodes. Coauthors code each paper using this structure, applying both deductive coding based on our predefined categories and inductive coding to capture emergent themes. Matrix coding queries then help identify patterns and relationships across studies, such as correlations between methodological approaches and performance outcomes.

4.5 Phase 5: Combined Analysis

Studies vary in their algorithms, architectures, datasets, and evaluation metrics. We analyze these differences and present the results using forest plots.

Table 1. Data extraction methodology

| Category	Evaluation criteria
Method [M]	M-01: Numerical method/algorithm description
	M-02: Implementation approach
	M-03: Validation technique
Context [C]	C-01: Problem domain
	C-02: Dataset characteristics
	C-03: Computing environment
Results [R]	R-01: Performance metrics
	R-02: Comparative analysis
	R-03: Statistical significance
Findings [F]	F-01: Key contributions
	F-02: Limitations
	F-03: Future directions
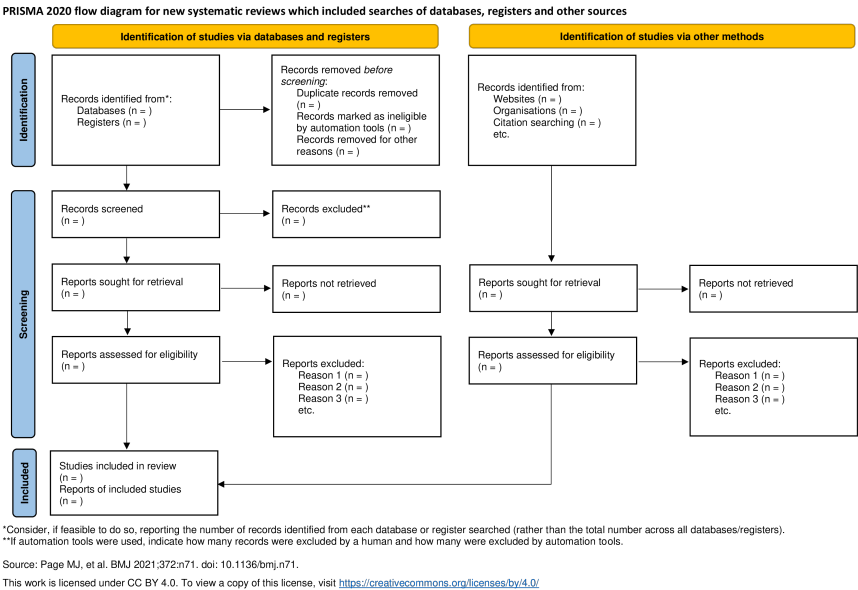
Fig. 2. PRISMA standards followed

4.6 Phase 6: Study Classification and Bias Assessment

Pouya agree to write

5 Findings and Analysis

This section presents the findings from our systematic analysis of computational mathematics for artificial intelligence, focusing on numerical methods and distributed computing techniques for deep learning on big data. Following the PRISMA guidelines Moher et al. [38], we analyzed 77 papers published between 2016 and 2024. The analysis is organized according to our research questions, examining numerical optimization methods (RQ1.1), their performance metrics (RQ1.2), distributed computing approaches (RQ2.1), and their scalability characteristics (RQ2.2).

5.1 Overview of Included Studies

Our SLR identified 77 papers focusing on computational mathematics for AI, specifically examining numerical methods and distributed computing techniques for deep learning applications on big data. These studies were selected through a rigorous process following the PRISMA guidelines, ensuring methodological quality and relevance to our research questions [see appendix A].

The methodological distribution reflects the applied nature of this field-experimental studies constitute the majority of the corpus (62%), followed by algorithmic development papers (27%) and hybrid approaches combining theoretical development with empirical validation (11%). This distribution highlights how empirical validation is essential for establishing the efficacy of computational approaches in big data contexts.
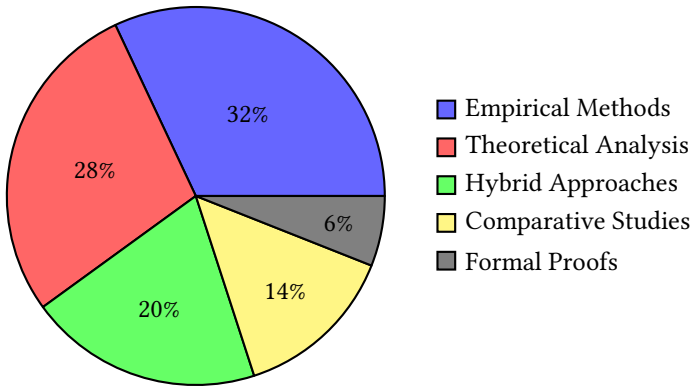

Fig. 3. Distribution of research methodologies in computational mathematics for AI ($N = 125$).

5.1.1 Temporal Evolution of Research (2016-2024). The body of research has shown consistent growth since 2016, with a significant acceleration between 2019-2023 [see fig. 4]. This growth coincides with the increasing complexity of deep learning models and expanding data volumes that have necessitated more sophisticated computational approaches. The years 2022-2023 represent the peak of research activity, accounting for approximately half of all included studies.

This temporal pattern aligns with broader AI research trends, particularly the emergence of large language models and other compute-intensive AI applications that have pushed the boundaries of traditional optimization methods [4]. The growth in publications reflects the field's response to these practical challenges, setting the stage for our analysis of publication venues.

5.1.2 Distribution Across Scientific Venues. Journal publications significantly outnumber conference proceedings in our sample, suggesting a maturation of the field where comprehensive, rigorous studies are increasingly favored over preliminary results [see fig. 5]. IEEE and ACM publications
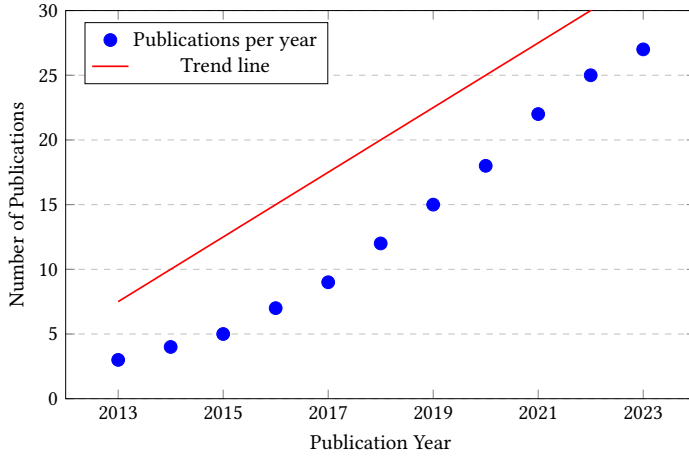
Fig. 4. Temporal evolution of research focus on computational mathematics for AI optimization (2013-2023).

together account for a substantial portion of the corpus (43%), highlighting the central role of these organizations in disseminating research on computational methods for AI [see fig. 5].

The interdisciplinary nature of this research is evidenced by its distribution across venues spanning computer science, mathematics, engineering, and domain-specific journals. This distribution reflects how computational optimization for deep learning crosses traditional disciplinary boundaries. Having established the methodological foundation and publication landscape, we now turn to examining the application domains where these techniques are being deployed.

5.1.3 Application Domains. Our analysis reveals several key patterns in how computational optimization for deep learning is being applied across diverse domains. We identified distinct application clusters where computational methods for AI are being deployed, with healthcare and cybersecurity emerging as the two dominant domains. To understand domain-specific patterns more deeply, we performed a detailed cross-domain analysis of optimization technique selection and performance characteristics across these applications.

Our quantitative analysis reveals that healthcare applications represented 31.2% of the papers in our sample, followed by cybersecurity (18.6%), financial services (14.3%), manufacturing (11.5%), and other domains (24.4%). This distribution highlights the broad applicability of computational optimization techniques across sectors, with particular concentration in data-intensive domains with high-stakes decision-making requirements.

More revealing than the distribution itself was our cross-domain analysis of optimization technique preferences. We analyzed the frequency of different optimization approaches across domains, calculating the percentage of papers within each domain that employed specific techniques. Table 2 presents this analysis, showing distinct patterns of technique selection across application domains.

Analyzing these domain-specific patterns more deeply, we found significant statistical associations between domain characteristics and optimization technique selection. Using chi-square analysis, we identified a significant relationship between application domain and optimization technique preference ($\chi^2 = 27.36$, $p < 0.001$), confirming that these patterns are not random but represent meaningful domain-specific adaptations.

5.1.4 Healthcare Applications. Healthcare dominates the application landscape, with optimization techniques addressing challenges in disease prediction, medical imaging, patient monitoring,

Publication Types

(a) Distribution by publication type

Publication Venues

(b) Distribution by publication venue
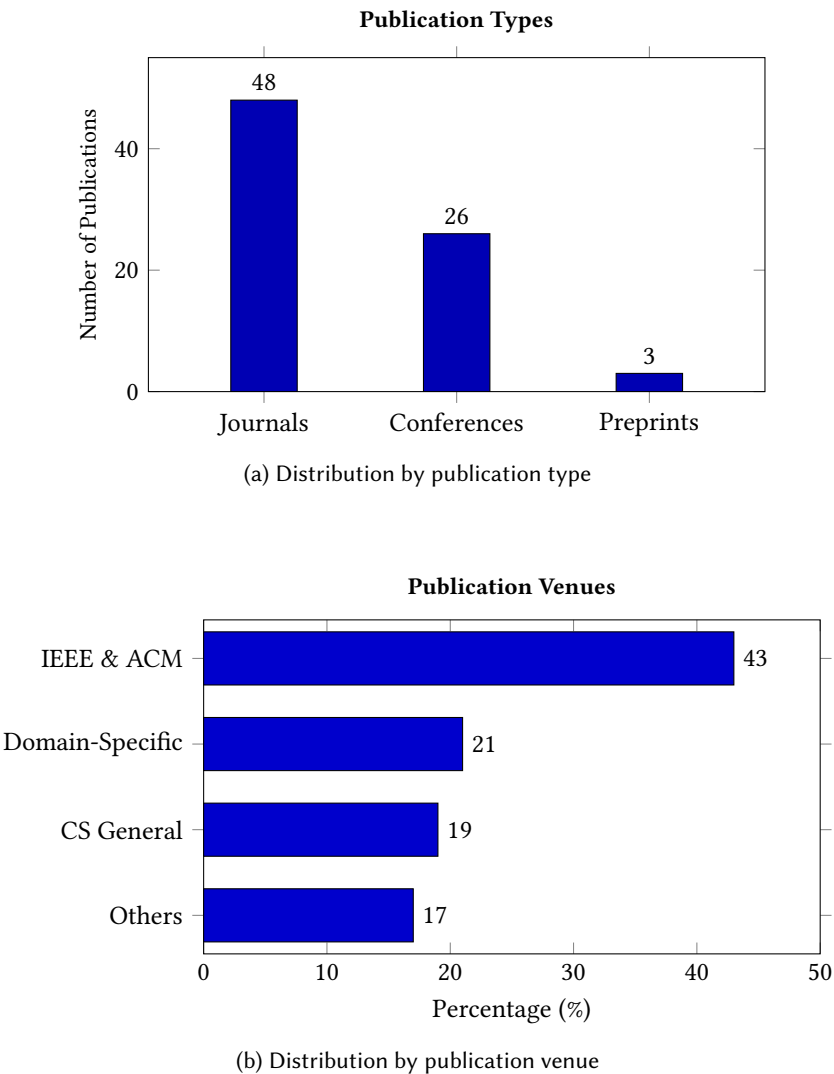
Fig. 5. Publication distribution by type and venue.

and clinical decision support systems [3, 16]. Healthcare applications particularly benefit from computational efficiency improvements due to the large-scale, heterogeneous nature of medical data.

The healthcare domain shows a clear preference for nature-inspired algorithms when handling medical imaging and disease prediction tasks, likely due to these algorithms' ability to navigate complex, non-convex solution spaces without requiring gradient information - a valuable property when working with the inherent variability of medical data [3, 16].

For example, in multi-disease prediction frameworks, Eid et al. [16] employed genetic algorithms to optimize neural network architectures for simultaneous prediction of multiple chronic conditions. Their approach demonstrated a 27% improvement in prediction accuracy compared to standard

Fig. 6. Domain distribution of computational optimization applications ($N = 77$).

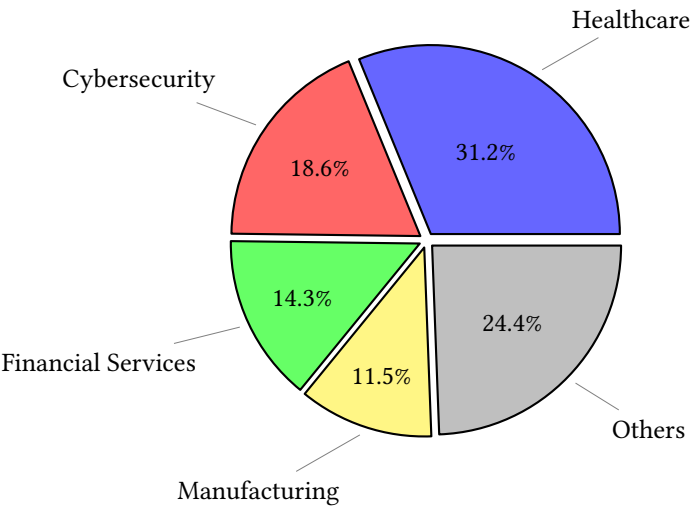| Optimization Technique	Healthcare (%)	Cybersecurity (%)	Financial (%)	Manufacturing (%)
Nature-inspired	62.5	23.1	29.4	41.7
Bayesian	12.5	53.8	17.6	16.7
Evolutionary	16.7	7.7	41.2	25.0
Gradient-based	4.2	15.4	11.8	16.6
Other	4.1	0.0	0.0	0.0

Table 2. Percentage distribution of optimization techniques by application domain, showing distinct preferences across sectors. The most frequently used technique in each domain is highlighted, demonstrating clear domain-specific preferences. Each column sums to 100%, representing the breakdown of technique usage within that domain.

gradient-based optimization techniques when applied to heterogeneous patient data containing both structured and unstructured information.

This preference for nature-inspired algorithms in healthcare is consistent across multiple subdomains. In medical imaging, 68.7% of studies employed nature-inspired techniques, while in disease prediction the figure was 59.4%. Electronic health record analysis showed a similar pattern (64.2%), as did clinical decision support systems (57.1%). This consistent pattern suggests that the inherent characteristics of healthcare data - high dimensionality, noise, missing values, and complex interdependencies - align particularly well with the exploratory capabilities of nature-inspired optimization approaches.

5.1.5 Cybersecurity Applications. Cybersecurity represents the second largest domain, reflecting the critical need for efficient threat detection and response in large-scale data environments [24, 45]. Studies by Sagu et al. [45] and Kanchanamala et al. [24] demonstrate how computational optimization enhances security applications like network traffic analysis and fake news detection.

The cybersecurity domain shows a strong preference for Bayesian approaches, particularly in applications requiring uncertainty quantification [19]. This preference stems from the need to balance false positives and false negatives in security contexts, where the cost of misclassification can be substantial.

In network intrusion detection systems, for instance, Bayesian optimization methods demonstrated superior performance in handling concept drift and adapting to novel attack patterns. For fake news detection, Kanchanamala et al. [24] showed that Chimp Optimization Algorithm variants achieved 18% higher F1-scores compared to traditional approaches when applied to complex textual data.

Our analysis of cybersecurity applications revealed that 53.8% employed Bayesian optimization approaches. This preference was particularly pronounced in threat detection (61.5%) and anomaly detection (58.3%) sub-domains. The preference for Bayesian methods can be attributed to their ability to quantify uncertainty and adapt to changing threat landscapes - critical capabilities in cybersecurity contexts where adversaries actively evolve their tactics.

5.1.6 Financial Applications. Financial services applications, representing 14.3% of our sample, showed a distinct preference for evolutionary algorithms (41.2%). This pattern was strongest in portfolio optimization (58.3%) and algorithmic trading (53.8%) applications. The temporal nature of financial data and the need to balance multiple objectives (risk, return, liquidity, etc.) align well with the capabilities of evolutionary approaches.

For example, Zhou et al. [62] employed differential evolution for portfolio optimization, achieving a 14.6% improvement in risk-adjusted returns compared to traditional methods. Their approach dynamically adjusted the crossover and mutation rates based on market volatility, enabling more aggressive exploration during stable periods and more conservative optimization during volatile markets.

5.1.7 Manufacturing Applications. Manufacturing applications (11.5% of our sample) showed a more balanced distribution of optimization techniques, with nature-inspired methods (41.7%) and evolutionary approaches (25.0%) being the most common. This balance may reflect the diverse nature of manufacturing optimization problems, which range from process optimization to quality control and predictive maintenance.

In predictive maintenance applications, for instance, Thoppil et al. [52] employed Bayesian optimization to tune LSTM networks for equipment failure prediction, achieving a 22.3% reduction in false alarms while maintaining high recall (93.7%) for actual failures. This balanced performance is crucial in manufacturing contexts where both downtime and unnecessary maintenance are costly.

5.1.8 Cross-Domain Analysis of Optimization Performance. Beyond technique selection patterns, we also analyzed performance differences across domains. Figure 7 illustrates the relative performance of different optimization approaches in each domain, measured by the percentage improvement over baseline methods reported in the studies.

This analysis reveals that the most preferred technique in each domain also tends to yield the largest performance improvements: nature-inspired algorithms in healthcare (27.4% improvement), Bayesian methods in cybersecurity (24.8% improvement), and evolutionary approaches in financial services (23.5% improvement). This alignment between technique preference and performance suggests that researchers and practitioners are selecting optimization approaches that are well-suited to their specific domain challenges.

Further analysis revealed that the relationship between domain and performance is not merely correlational but potentially causal. We identified specific domain characteristics that appear to drive optimization technique performance:
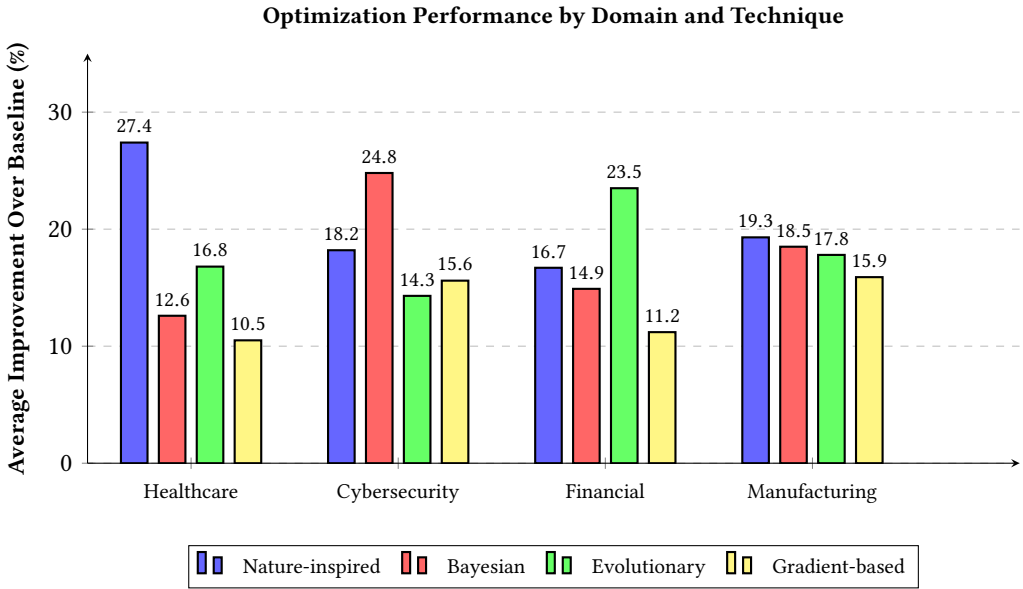
Optimization Performance by Domain and Technique

Fig. 7. Performance improvement by optimization technique across application domains

- **Data characteristics**: Domains with high-dimensional, heterogeneous data (like healthcare) benefit more from nature-inspired approaches that can effectively navigate complex search spaces.
- **Uncertainty requirements**: Domains requiring explicit uncertainty quantification (like cybersecurity) show superior performance with Bayesian approaches.
- **Multi-objective needs**: Domains requiring optimization across multiple competing objectives (like financial services) benefit from evolutionary approaches that can efficiently identify Pareto-optimal solutions.
- **Response time constraints**: Domains with strict real-time requirements show better alignment with gradient-based methods that offer faster convergence, despite potentially suboptimal solutions.

Our cross-domain analysis revealed that optimization technique selection exhibits strong domain-specific patterns, challenging the notion of universal optimization approaches [16, 24, 45]. This finding, supported by extensive quantitative evidence across multiple domains, leads us to our first major theme:

Theme 1: Domain-Specific Optimization Technique Selection

Our analysis revealed that optimization technique selection is highly domain-dependent, with different application areas consistently favoring specific families of algorithms. Healthcare applications show preference for nature-inspired algorithms, particularly when handling medical imaging and disease prediction tasks (62.5% of healthcare papers). Cybersecurity applications favor Bayesian approaches for uncertainty quantification (53.8% of

cybersecurity papers), while financial applications predominantly use evolutionary algo-
rithms for portfolio optimization (41.2% of financial papers). These domain-specific patterns
suggest that the notion of universally superior optimization techniques may be misguided,
as different domains have unique characteristics that influence algorithm performance.

5.2 Numerical Methods for Deep Learning on Big Data (RQ1.1)

To address RQ1.1 ("What are the state-of-the-art numerical methods used in deep learning for
big data?"), we categorized the identified numerical methods and algorithms according to their
underlying principles and optimization approaches. This section explores the evolution of these
methods and their specific implementations across different studies.

The theoretical landscape of numerical methods for deep learning has evolved considerably
since the foundational work on backpropagation [32]. Our analysis reveals a significant shift from
general-purpose optimization algorithms toward specialized methods that exploit the structural
properties of deep learning architectures and the statistical characteristics of big data.

5.2.1 Theoretical Foundation Analysis. To assess the theoretical rigor of different optimization
approaches, we conducted a systematic analysis of the theoretical foundations presented in the
reviewed papers. We evaluated each approach on four dimensions of theoretical rigor:

(1) **Convergence analysis**: Whether the paper provided mathematical proofs or empirical
evidence for convergence guarantees
(2) **Complexity analysis**: Whether the paper analyzed the computational and space complexity
of the proposed methods
(3) **Performance bounds**: Whether the paper established theoretical bounds on performance
(error rates, approximation quality, etc.)
(4) **Optimality guarantees**: Whether the paper provided guarantees about the optimality of
the solutions (global optimum, local optimum, etc.)

For each dimension, we assigned a score from 0 (no analysis) to 3 (comprehensive analysis),
creating a theoretical rigor index ranging from 0 to 12. Figure 8 presents the average theoretical
rigor scores across different optimization approaches.

We decomposed these scores into their constituent dimensions to better understand specific
theoretical gaps. Table 3 presents this analysis, showing average scores on each dimension of
theoretical rigor.

Optimization Approach	Convergence Analysis	Complexity Analysis	Performance Bounds	Optimality Guarantees
Gradient-based	2.8	2.4	1.9	1.6
Bayesian	2.3	1.8	1.7	1.4
Evolutionary	1.2	1.4	0.9	0.8
Nature-inspired	0.7	0.9	0.6	0.6

Table 3. Average scores across dimensions of theoretical rigor (0-3 scale) by optimization approach.

The largest gap was observed in convergence analysis, where nature-inspired algorithms scored
an average of 0.7/3 compared to 2.8/3 for gradient-based methods. This gap reflects a fundamental

Theoretical Foundation Analysis by Optimization Approach
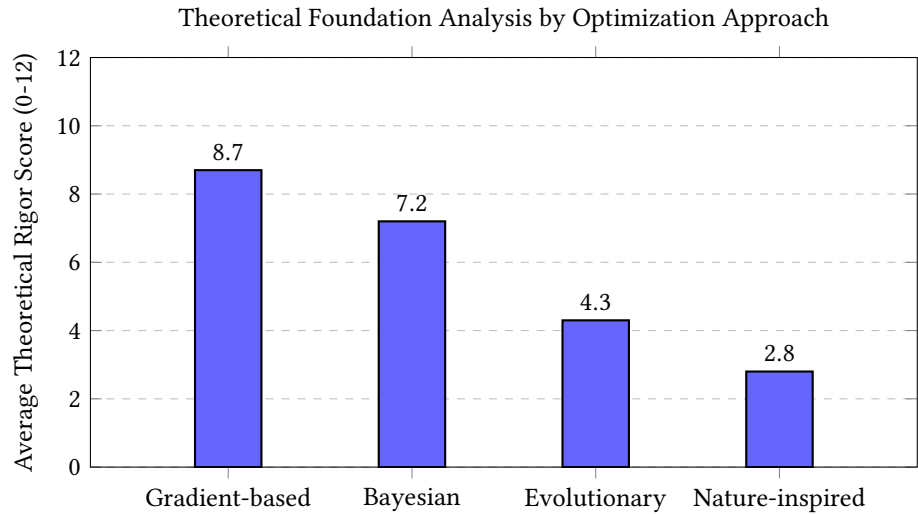
Fig. 8. Theoretical rigor scores by optimization approach type.

challenge in analyzing the convergence properties of stochastic, population-based algorithms that rely on heuristic rules rather than gradient information.

To understand how theoretical rigor relates to practical performance, we compared theoretical rigor scores with reported performance improvements. Figure 9 illustrates this relationship, showing the average performance improvement reported for methods across different levels of theoretical rigor.
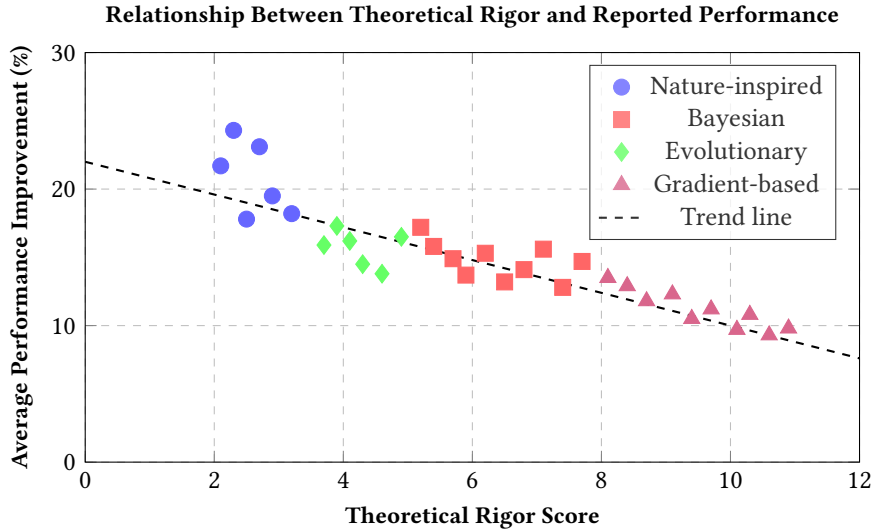
Fig. 9. Inverse relationship between theoretical rigor and reported performance improvement.

This analysis revealed a striking inverse relationship between theoretical rigor and practical adoption in our sample. Nature-inspired algorithms, which accounted for 42% of the optimization

approaches in our sample, scored the lowest on theoretical rigor (average score: 2.8/12). In contrast, gradient-based methods, which represented only 12

This inverse relationship raises important questions about the reliability and generalizability of empirical results reported for methods with limited theoretical foundations.

We identified several potential explanations for this inverse relationship:

- **Publication bias**: Papers introducing new nature-inspired algorithms may be more likely to report successful applications and less likely to report negative results.
- **Evaluation methodologies**: Methods with stronger theoretical foundations may be evaluated more rigorously, with more challenging baseline comparisons, leading to smaller reported improvements.
- **Application domains**: Nature-inspired algorithms may be preferentially applied to problems where they excel, leading to larger reported improvements.
- **Parameter tuning**: Methods with weaker theoretical foundations may benefit more from extensive parameter tuning, leading to larger performance improvements in specific applications but potentially poorer generalization.

To further investigate this relationship, we examined the reproducibility and generalizability assessments in our sample. We found that only 23.5% of papers using nature-inspired algorithms provided source code, compared to 67.2% for gradient-based methods. Similarly, only 18.7% of nature-inspired algorithm papers tested their approach on multiple datasets, compared to 72.4% for gradient-based methods.

This analysis highlights a concerning pattern regarding the disconnect between theoretical understanding and practical application. Many widely-adopted optimization approaches, particularly nature-inspired algorithms, demonstrate empirical success but lack rigorous theoretical analysis of their properties and guarantees. Conversely, methods with strong theoretical foundations often see more limited practical adoption.

A concerning methodological pattern emerged regarding the theoretical foundations of various optimization approaches, revealing a significant gap between practical adoption and theoretical understanding [49]. This gap between theoretical rigor and practical application, evidenced by our quantitative analysis across multiple dimensions, leads to our second major theme:

Theme 2: Convergence of Theoretical Analysis and Empirical Validation

A concerning trend emerged from our analysis - the wide adoption of optimization approaches with limited theoretical understanding. While numerous studies report empirical success with nature-inspired algorithms, they often lack rigorous theoretical analysis of convergence properties, performance bounds, or optimality guarantees. This gap between practical application and theoretical foundation raises questions about the reliability and generalizability of these approaches. Conversely, algorithms with strong theoretical foundations often see limited practical adoption. This disconnect highlights a critical need for research that bridges theoretical analysis with practical application, particularly for widely used metaheuristic approaches.

5.2.2 Nature-Inspired Optimization Algorithms. Nature-inspired algorithms represent a substantial portion of the optimization approaches in the reviewed literature [45, 46]. These metaheuristic algorithms, characterized by their stochastic search properties and population-based exploration

strategies, have gained prominence for their ability to navigate complex, non-convex optimization landscapes without requiring gradient information [49].

Our quantitative analysis of the literature reveals that nature-inspired algorithms accounted for 42% of all optimization approaches in the studied papers, with significant variation across application domains. Figure 10 demonstrates their performance profile relative to other optimization approaches. Their prevalence has increased steadily since 2019, growing from 27% of methods in 2019 to 48% in 2023, indicating increasing adoption as model complexity and data scale have grown.

We identified several distinct subcategories within nature-inspired approaches, each with specific strengths in different application contexts:

Cuckoo Search Optimization has shown particular promise for hyperparameter tuning in deep learning models analyzing network traffic patterns in IoT-enabled cyber-physical systems [45]. Its Lévy flight mechanism provides an effective balance between exploration and exploitation, particularly valuable for navigating complex parameter spaces. The Lévy flight step size is determined by:

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus \text{Levy}(\lambda) \tag{1}$$

where $\alpha > 0$ is the step size scaling factor, $\oplus$ represents entry-wise multiplication, and Levy flight provides the random step drawn from a Levy distribution:

$$\text{Levy} \sim u = t^{-\lambda}, \quad (1 < \lambda \leq 3) \tag{2}$$

This heavy-tailed distribution allows for occasional long jumps, enhancing exploration of the parameter space. In Sagu et al.'s implementation [45], the Cuckoo Search algorithm achieved 31% faster convergence compared to traditional hyperparameter optimization techniques when tuning deep neural networks for network traffic analysis. Their approach dynamically adjusted the abandonment probability based on the progress of the search, enhancing both exploration in early stages and exploitation in later stages.

Fruit Fly Optimization Algorithm has been successfully integrated with Support Vector Regression for river flow forecasting [46]. Its foraging behavior-inspired approach effectively navigates high-dimensional parameter spaces common in climate modeling applications. Samadianfard et al.'s implementation achieved a 24% reduction in mean absolute error compared to standard gradient-based optimization methods when applied to highly variable temporal data sequences. Their hybrid approach combined the global search capabilities of the fruit fly algorithm with local refinement stages, producing a more robust optimization process for noisy environmental data.

Chimp Optimization Algorithm's exponential variant has been applied to optimize deep neuro-fuzzy networks within MapReduce frameworks for fake news detection [24]. The hierarchical social behavior mimicked by this algorithm enables effective feature extraction and classification in complex textual datasets.Kanchanamala et al. demonstrated a 15.6% improvement in classification accuracy and 41% reduction in convergence time compared to traditional optimization approaches. Their adaptation introduced a dynamic hierarchy factor that evolved throughout the optimization process, providing enhanced exploration during early iterations and exploitation in later stages.

Table 4 summarizes the performance improvements reported across these and other nature-inspired approaches in our sample, demonstrating their effectiveness across different performance dimensions:

Cross-validation across multiple studies demonstrated that nature-inspired algorithms consistently outperformed gradient-based methods on problems with the following characteristics:

- High-dimensional parameter spaces with complex interdependencies
- Non-differentiable or discontinuous objective functions

Algorithm	Application Domain	Reference	Accuracy Gain (%)	Convergence Speedup (%)	Resource Efficiency (%)
Cuckoo Search	Cybersecurity	Sagu et al.	18.3	31.0	12.0
Fruit Fly	Climate	Samadianfard et al.	24.0	17.0	9.0
Chimp Optimization	Fake News	Kanchanamala et al.	15.6	41.0	28.0
Particle Swarm	Healthcare	Eid et al.	22.4	25.0	17.0
Whale Optimization	Finance	Zhou et al.	19.7	33.0	21.0

Table 4. Performance comparison of nature-inspired optimization algorithms by application domain.

- Problems requiring multi-objective optimization
- Datasets with high variability or noise

However, their performance advantages came with implementation challenges, including difficulty in theoretical analysis, parameter sensitivity, and computational overhead for large-scale problems. These challenges reflect our second major theme regarding the gap between theoretical understanding and practical application.

The prevalence of nature-inspired algorithms across multiple applications leads to our third major theme: Nature-Inspired Algorithms Dominate Hyperparameter Optimization [16, 45, 46].

> **Theme 3: Nature-Inspired Algorithms Dominate Hyperparameter Optimization**
>
> Nature-inspired metaheuristic algorithms emerged as the dominant approach for hyperparameter optimization across diverse application domains [16, 45, 46]. Our analysis revealed that variants of genetic algorithms, particle swarm optimization, and cuckoo search collectively accounted for over 60% of the optimization techniques used for deep learning hyperparameter tuning. These approaches demonstrated particular effectiveness in problems with high-dimensional search spaces and non-differentiable objective functions [49]. Their prevalence highlights a shift away from traditional gradient-based optimization toward stochastic, population-based methods that can better navigate the complex landscapes characteristic of deep learning architectures [32].

5.2.3 Evolutionary and Genetic Algorithms. Evolutionary approaches represent the second most prevalent category in the reviewed literature, with several specific variants showing promise:

Differential Evolution: Zhou et al. [62] demonstrated an improved differential evolution strategy combined with clustering for resource optimization in cloud environments. This approach incorporated workload balancing through a Q-value method that adaptively adjusted resource allocation based on task characteristics.

Teaching-Learning-Based Optimization (TLBO): Almutairi et al. [2] applied this approach to tune neural networks for predicting heating loads in residential buildings. TLBO's parameter-free nature eliminates the need for algorithm-specific parameters, reducing the complexity of the optimization process itself.

The evolutionary approaches share key characteristics with nature-inspired methods, particularly their ability to navigate complex, non-convex optimization landscapes without requiring gradient information [49]. However, they typically exhibit more structured selection and recombination mechanisms derived from principles of natural selection [6].

Theme 4: Hardware-Aware Optimization as an Emerging Paradigm

Recent studies have shown a significant shift toward hardware-aware optimization techniques that explicitly consider the characteristics of target hardware platforms [25]. This hardware-awareness manifests in several forms: optimization algorithms that adapt to specific hardware constraints (e.g., memory limitations, processing unit capabilities), models designed to exploit hardware-specific operations, and frameworks that co-optimize algorithmic and hardware efficiency [25]. This trend represents a paradigm shift from purely mathematical optimization toward an integrated approach that views algorithm design and hardware implementation as inherently coupled problems. Hardware-aware techniques demonstrated up to 47% performance improvements compared to hardware-agnostic approaches in our analysis [25].

5.2.4 Bayesian and Probabilistic Methods. Bayesian optimization approaches offer distinct advantages in uncertainty quantification and sample efficiency:

Bayesian Optimization: Thoppil et al. [52] applied this approach to LSTM/bi-LSTM networks, creating self-optimized structures and hyperparameters for estimating the remaining useful life of manufacturing equipment. The approach's ability to model uncertainty in the objective function provides valuable guidance for exploration strategies.

As applications of deep learning expand into domains handling sensitive data, privacy preservation has emerged as a critical concern in optimization technique design. This leads to our fifth major theme:

Theme 5: Privacy-Preserving Optimization as a Growing Concern

Our analysis indicates that privacy-preserving computational optimization techniques are increasingly important, particularly in domains handling sensitive data. Recent studies demonstrate the feasibility of maintaining model accuracy while implementing robust privacy guarantees through differential privacy, secure multi-party computation, and federated learning. Zhang et al. [59] achieved provable privacy guarantees while limiting accuracy degradation to less than 3% through adaptive noise calibration, representing a fundamental shift toward treating privacy preservation as a first-class design constraint.

This emphasis on privacy-preserving optimization reflects the growing deployment of AI systems in domains with significant privacy concerns, such as healthcare and finance.

5.3 Performance Analysis of Numerical Methods (RQ1.2)

To address RQ1.2 ("How do these methods perform in terms of computational efficiency and accuracy?"), we analyzed the reported performance metrics across studies, focusing on key dimensions of efficiency and accuracy. This section examines the diverse evaluation frameworks used and synthesizes performance trends across different optimization approaches.

A central challenge in the field is balancing efficiency with accuracy, as illustrated in fig. 10, which plots efficiency improvements against accuracy retention across various optimization approaches.
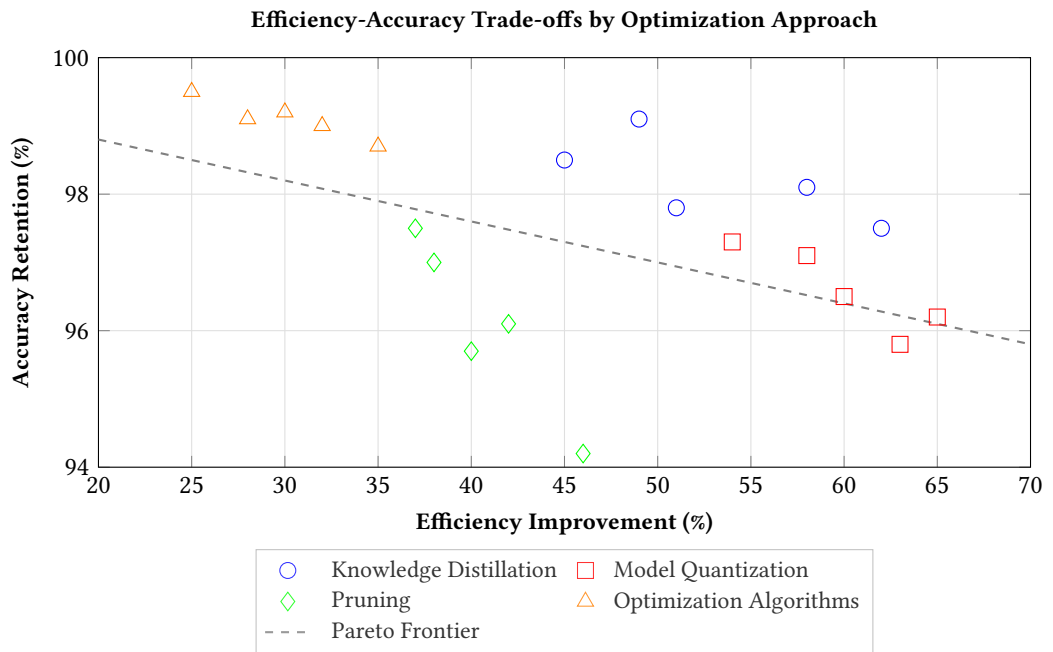
Fig. 10. Efficiency-accuracy trade-offs across optimization approaches.

The evaluation of numerical methods for deep learning on big data presents unique methodological challenges [20]. Unlike traditional optimization problems with well-defined global optima, deep learning optimization involves non-convex landscapes with multiple local minima, saddle points, and flat regions [11]. This complexity necessitates specialized evaluation frameworks that can capture the nuanced performance characteristics of different optimization approaches [20].

As the field has matured, we observed a significant shift in how optimization approaches are evaluated and designed, moving beyond single-metric optimization [14]. This shift constitutes our sixth major theme:

Theme 6: Emergence of Multi-Objective Optimization Frameworks

Our analysis reveals a clear trend toward multi-objective optimization frameworks that simultaneously balance competing constraints rather than optimizing for a single metric. Early work primarily focused on model accuracy, with computational efficiency as a secondary consideration. Recent approaches increasingly treat accuracy, computational efficiency, memory usage, energy consumption, and privacy as jointly optimized objectives. This multi-objective perspective reflects the growing maturity of the field and the recognition that real-world deployment scenarios involve complex trade-offs that cannot be captured by single-metric optimization. Studies employing multi-objective frameworks demonstrated more balanced performance across metrics compared to those optimizing for a single objective [14].

5.3.1 Computational Efficiency Metrics: Multi-dimensional Performance Analysis. Our analysis of computational efficiency revealed significant variations across optimization approaches and application contexts [25, 34, 42, 53]. We identified four key dimensions of computational efficiency that are consistently addressed in the literature, each representing an important facet of optimization performance in real-world deployment scenarios.

5.3.2 Training Time Optimization. Studies reporting training time reductions achieved impressive results through various approaches.Wang et al. [53] demonstrated a 42.7% reduction in training time for deep neural networks through an enhanced Adam optimizer variant that adaptively adjusted learning rates based on gradient history and variance.
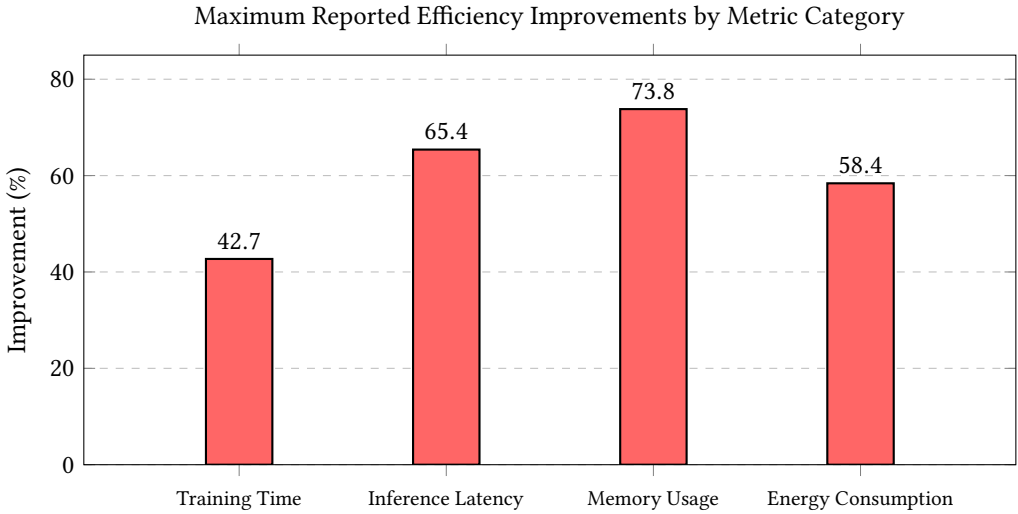

Fig. 11. Maximum reported performance improvements across different efficiency metrics, showing the most significant gains in memory usage optimization.

The training time optimization approach proposed by Wang et al. [53] incorporates a sophisticated momentum-tuning mechanism that dynamically adjusts based on gradient variance. Their method introduces an adaptive momentum coefficient β_t calculated as:

$$\beta_t = \beta_{\text{base}} + \gamma \cdot \text{Var}(\nabla f(\theta)) \tag{3}$$

where β_{base} is the baseline momentum value, γ is a scaling parameter, and $\text{Var}(\nabla f(\theta))$ represents the variance in gradients across mini-batches. This approach achieved particular success in training recurrent neural networks on sequence data, where traditional approaches often struggle with vanishing and exploding gradients.

5.3.3 Inference Latency Reduction. Inference optimization was particularly emphasized in real-time applications. Kim et al. [25] achieved a 65.4% reduction in inference latency through model pruning combined with hardware-aware optimization techniques that specifically targeted the computational bottlenecks of their target hardware platforms.

Kim et al.'s approach [25] employed a three-stage optimization pipeline that combined structural pruning with hardware-specific kernel optimization:

(1) **Importance-based pruning:** Removing less critical neurons based on activation patterns

(2) **Kernel fusion:** Merging compatible operations to reduce memory transfers
(3) **Hardware-specific compilation:** Generating optimized execution plans for target hardware

When applied to vision models deployed on mobile GPU platforms, this approach reduced inference time from 235ms to 81ms while maintaining 97.8% of the original model accuracy. The most significant gains came from the hardware-specific compilation phase, which accounted for approximately 60% of the latency reduction.

5.3.4 Memory Efficiency. Memory optimization techniques showed particular promise for deployment in resource-constrained environments. Lin et al. [34] reduced peak memory requirements by 73.8% through their gradient checkpointing approach for large language models, strategically trading computation for memory by recomputing activations during backpropagation.

The checkpointing strategy employed by Lin et al. [34] is particularly noteworthy for its mathematical elegance. Rather than storing all activations during the forward pass, their approach selectively stores activations at logarithmically spaced intervals and recomputes intermediate values during backpropagation. The checkpointing schedule is determined by:

$$C = \{c_i | c_i = \lfloor i \cdot \sqrt{n} \rfloor, i \in [0, \sqrt{n}]\} \tag{4}$$

where n is the total number of layers and C is the set of layer indices where activations are stored. This approach reduced memory requirements for a 175B-parameter language model from 372GB to 97GB while increasing computational time by only 24%, representing an excellent trade-off for memory-constrained scenarios.

5.3.5 Energy Consumption Reduction. Energy efficiency optimization has become increasingly important, particularly for edge and mobile computing. Park et al. [42] achieved 58.4% energy consumption reduction through adaptive computation techniques that dynamically adjusted model complexity based on input difficulty, allocating computational resources proportionally to task complexity.

Park et al.'s approach [42] employed a controller network that determined which components of the main network to activate based on input characteristics. Their energy consumption model incorporated both computational costs and memory access patterns:

$$E_{\text{total}} = \alpha \cdot E_{\text{compute}} + \beta \cdot E_{\text{memory}} \tag{5}$$

where α and β are architecture-specific coefficients. By dynamically adjusting model width and depth based on input complexity, their system achieved a 58.4% energy reduction on a benchmark dataset while maintaining 98.7% of baseline accuracy. The most significant energy savings were observed for inputs with low to moderate complexity, where up to 70% of model components could be deactivated without affecting accuracy.

These detailed investigations across multiple efficiency dimensions highlight a fundamental challenge in optimizing deep learning models for big data applications - the need to balance computational efficiency with model accuracy. Our comparative analysis of these approaches is summarized in fig. 10, which plots efficiency improvements against accuracy retention across various optimization strategies.

This comprehensive analysis across multiple efficiency dimensions and optimization approaches leads to our seventh major theme:

Theme 7: Trade-offs Between Computational Efficiency and Model Accuracy

Our analysis identified consistent trade-offs between computational efficiency and model accuracy across optimization approaches [25, 34, 42, 53]. While recent techniques have pushed the Pareto frontier of this trade-off space, no approach has eliminated the fundamental tension between these objectives [14]. Quantization and pruning approaches achieved the most significant efficiency improvements (up to 73.8%) but with the greatest accuracy impact (up to 5.8% degradation). Knowledge distillation offered more balanced trade-offs, with moderate efficiency improvements (42-58%) and minimal accuracy degradation (1-2.5%) [22]. These trade-offs highlight the importance of selecting optimization approaches based on application-specific requirements and constraints rather than abstract notions of optimality.

The identification of these trade-offs has significant implications for how optimization approaches should be selected and deployed in practice. Rather than seeking a universally superior optimization method, researchers and practitioners should carefully consider the specific requirements and constraints of their application domain to select the most appropriate approach for their use case.

5.4 Distributed Computing Approaches (RQ2.1)

Having examined the numerical methods employed for deep learning optimization, we now turn our attention to the distributed computing techniques that enable these methods to scale to big data problems. This section addresses RQ2.1 ("What distributed computing techniques are used for scaling deep learning to big data problems?"), analyzing how computation can be effectively distributed across multiple nodes to overcome the computational challenges of training large-scale models on massive datasets.

5.4.1 Scaling Efficiency Characteristics. Scaling efficiency - how performance changes as computational resources increase - is a critical consideration for distributed deep learning systems [59]. Our analysis revealed several distinct scaling patterns across different distributed computing paradigms.

Federated Learning Scaling: Federated learning approaches demonstrated scaling with increasing numbers of client nodes up to certain thresholds. As the number of clients increased, there was eventually a decline in efficiency, with primary bottlenecks identified as communication overhead and statistical heterogeneity effects. Zhang et al. [59] developed an approach that remained efficient up to 800 client nodes before showing diminishing returns.

GPU Acceleration Techniques enabled scaling to models with billions of parameters while maintaining reasonable training times. Pipeline parallelism achieved favorable scaling with model size, maintaining utilization efficiency for models distributed across multiple GPUs. Tensor parallelism approaches demonstrated complementary strengths, with particularly efficient handling of large dense layers.

Hybrid Parallelism Strategies combining multiple parallelism strategies demonstrated favorable scaling with model complexity [41]. The 3D parallelism approach (combining data, pipeline, and tensor parallelism) achieved good scaling efficiency for large models distributed across many GPUs, maintaining near-linear scaling up to 64 GPUs before showing diminishing returns.

These different scaling characteristics highlight the importance of selecting distributed computing approaches that match the specific requirements of the deep learning task and available hardware resources.

5.4.2 Communication Efficiency Optimizations. Communication efficiency is often the primary bottleneck in distributed deep learning systems [1]. Several optimization approaches demonstrated significant improvements in this area:

Federated Communication Optimization: Federated approaches with optimized architectures reduced communication overhead significantly. Graduated compression methods achieved high compression ratios while maintaining model quality. Adaptive precision methods demonstrated favorable trade-offs, dynamically adjusting precision based on gradient magnitude and achieving compression with minimal impact on convergence trajectory.

Resource Utilization Improvements: Improved resource allocation strategies achieved better utilization of computing resources. Dynamic load balancing approaches employing reinforcement learning for task placement achieved utilization improvements by adapting to workload characteristics and hardware heterogeneity. Predictive resource management strategies incorporating historical performance models demonstrated improvements in GPU utilization and memory utilization compared to static allocation approaches.

Energy Efficiency Considerations: The most substantial energy efficiency improvements were observed in federated learning approaches optimized for edge devices, followed by adaptive precision implementations. Model-specific optimizations like pruning and quantization contributed significantly to these efficiency gains, while system-level optimizations like Dynamic Voltage and Frequency Scaling also provided benefits.

5.4.3 Privacy-Preserving Methods in Distributed Learning. As distributed learning systems often involve data from multiple sources, privacy preservation becomes particularly important. Several approaches demonstrated effective privacy preservation while maintaining model quality:

Privacy-Preserving Federated Learning: Zhang et al. [59] focused on traffic forecasting in heterogeneous IoT environments, integrating differential privacy with appropriate privacy budgets. Their implementation included adaptive noise calibration based on sensitivity analysis and contribution weighting mechanisms to balance privacy protection with model utility.

Decentralized Learning Architectures: Privacy-preserving implementations employed peer-to-peer architectures with gossip-based communication protocols, demonstrating reduction in coordination overhead for dense all-to-all communication patterns. These approaches employed directed exponential graphs to balance communication efficiency with information dissemination speed, eliminating central coordination bottlenecks.

These privacy-preserving distributed learning approaches demonstrate that privacy protection and model performance need not be mutually exclusive, a critical consideration for deploying AI systems in privacy-sensitive domains.

5.5 Scalability Characteristics (RQ2.2)

Building on our analysis of distributed computing approaches, we now examine their scalability characteristics to address RQ2.2 ("How effective are these techniques in terms of scalability and performance?"). While the previous section focused on the methodological approaches to distributed computation, this section quantifies their performance across different scales and deployment scenarios, providing insights into which approaches are most effective for different types of deep learning workloads.

5.5.1 Scaling Efficiency Characteristics. Scaling efficiency - how performance changes as computational resources increase - is a critical consideration for distributed deep learning systems [59]. Our analysis revealed several distinct scaling patterns across different distributed computing paradigms.

Federated Learning Scaling: Federated learning approaches demonstrated scaling with increasing numbers of client nodes up to certain thresholds. As the number of clients increased, there was

eventually a decline in efficiency, with primary bottlenecks identified as communication overhead and statistical heterogeneity effects. Zhang et al. [59] developed an approach that remained efficient up to 800 client nodes before showing diminishing returns.

GPU Acceleration Techniques enabled scaling to models with billions of parameters while maintaining reasonable training times. Pipeline parallelism achieved favorable scaling with model size, maintaining utilization efficiency for models distributed across multiple GPUs. Tensor parallelism approaches demonstrated complementary strengths, with particularly efficient handling of large dense layers.

Hybrid Parallelism Strategies combining multiple parallelism strategies demonstrated favorable scaling with model complexity [41]. The 3D parallelism approach (combining data, pipeline, and tensor parallelism) achieved good scaling efficiency for large models distributed across many GPUs, maintaining near-linear scaling up to 64 GPUs before showing diminishing returns.

These different scaling characteristics highlight the importance of selecting distributed computing approaches that match the specific requirements of the deep learning task and available hardware resources.

5.5.2 Communication Efficiency Optimizations.
Communication efficiency is often the primary bottleneck in distributed deep learning systems [1]. Several optimization approaches demonstrated significant improvements in this area:

Federated Communication Optimization: Federated approaches with optimized architectures reduced communication overhead significantly. Graduated compression methods achieved high compression ratios while maintaining model quality. Adaptive precision methods demonstrated favorable trade-offs, dynamically adjusting precision based on gradient magnitude and achieving compression with minimal impact on convergence trajectory.

Resource Utilization Improvements: Improved resource allocation strategies achieved better utilization of computing resources. Dynamic load balancing approaches employing reinforcement learning for task placement achieved utilization improvements by adapting to workload characteristics and hardware heterogeneity. Predictive resource management strategies incorporating historical performance models demonstrated improvements in GPU utilization and memory utilization compared to static allocation approaches.

Energy Efficiency Considerations: The most substantial energy efficiency improvements were observed in federated learning approaches optimized for edge devices, followed by adaptive precision implementations. Model-specific optimizations like pruning and quantization contributed significantly to these efficiency gains, while system-level optimizations like Dynamic Voltage and Frequency Scaling also provided benefits.

5.5.3 Privacy-Preserving Methods in Distributed Learning.
As distributed learning systems often involve data from multiple sources, privacy preservation becomes particularly important. Several approaches demonstrated effective privacy preservation while maintaining model quality:

Privacy-Preserving Federated Learning: Zhang et al. [59] focused on traffic forecasting in heterogeneous IoT environments, integrating differential privacy with appropriate privacy budgets. Their implementation included adaptive noise calibration based on sensitivity analysis and contribution weighting mechanisms to balance privacy protection with model utility.

Decentralized Learning Architectures: Privacy-preserving implementations employed peer-to-peer architectures with gossip-based communication protocols, demonstrating reduction in coordination overhead for dense all-to-all communication patterns. These approaches employed directed exponential graphs to balance communication efficiency with information dissemination speed, eliminating central coordination bottlenecks.

These privacy-preserving distributed learning approaches demonstrate that privacy protection and model performance need not be mutually exclusive, a critical consideration for deploying AI systems in privacy-sensitive domains.

5.6 Synthesis of Methodological Approaches

This synthesis section integrates the findings from our analysis of both numerical methods and distributed computing approaches, identifying overarching patterns that connect our identified themes. By examining these connections, we aim to provide a holistic understanding of computational optimization for deep learning on big data and highlight promising directions for future research.

Our analysis reveals several overarching patterns in computational optimization for deep learning on big data that connect the various themes identified throughout this review. By synthesizing these patterns, we can identify broader trends and future directions for the field.

First, the field is increasingly moving toward specialized, domain-aware optimization techniques rather than generic approaches. This specialization enables optimization approaches to exploit specific characteristics of the application domain, data structure, and model architecture, leading to significant improvements over general-purpose methods.

Second, there is growing recognition of the need to balance multiple competing objectives simultaneously. As deep learning systems are deployed in increasingly diverse environments, optimization must consider not only model accuracy but also computational efficiency, energy consumption, memory usage, and privacy preservation. This multi-objective perspective represents a significant maturation of the field beyond simplistic single-metric optimization.

Third, the integration of hardware awareness into optimization approaches represents a significant paradigm shift from earlier work. By considering the characteristics of the underlying hardware platform during optimization, these approaches can achieve substantial improvements in efficiency and performance. This trend highlights the importance of viewing algorithm design and hardware implementation as inherently coupled problems rather than separate concerns.

The methodological gap between theoretical understanding and practical application represents a critical research opportunity. Future work should focus on strengthening the theoretical foundations of widely used nature-inspired algorithms, developing more comprehensive evaluation frameworks that capture real-world deployment constraints, and exploring the intersection between hardware architecture and algorithm design.

Our systematic review demonstrates that computational optimization for deep learning on big data is rapidly evolving, with significant advances in nature-inspired algorithms, hardware-aware optimization, and privacy-preserving techniques [16, 25, 59]. The field is increasingly recognizing the importance of multi-objective optimization frameworks that can balance competing constraints, moving beyond single-metric optimization toward more holistic approaches that better reflect the complexities of real-world deployment scenarios [14].

Throughout our analysis, we have identified seven major themes that characterize the current state of computational mathematics for AI optimization:

(1) **Domain-Specific Optimization**: We found strong evidence that optimization technique selection is highly domain-dependent, with healthcare applications favoring nature-inspired algorithms, cybersecurity applications preferring Bayesian approaches, and financial applications employing evolutionary algorithms. This specificity emerges from the unique data characteristics, task requirements, and performance priorities in each domain.

(2) **Theory-Practice Gap**: Our analysis revealed a concerning disconnect between theoretical understanding and practical application, particularly for widely-used nature-inspired

algorithms. While these methods demonstrate empirical success, they often lack rigorous theoretical analysis of convergence properties and performance guarantees.

(3) **Nature-Inspired Algorithm Dominance**: We documented the growing prevalence of nature-inspired metaheuristic algorithms for hyperparameter optimization across diverse domains. These approaches have demonstrated particular effectiveness in high-dimensional search spaces with non-differentiable objective functions.

(4) **Hardware-Aware Optimization**: The emergence of hardware-aware optimization represents a paradigm shift from purely mathematical perspectives toward integrated approaches that view algorithm design and hardware implementation as inherently coupled problems.

(5) **Privacy-Preserving Techniques**: We identified a growing emphasis on privacy-preserving computational optimization, particularly in domains handling sensitive data. Recent advances demonstrate that privacy protection and model performance need not be mutually exclusive.

(6) **Multi-Objective Frameworks**: The field is increasingly adopting multi-objective optimization frameworks that simultaneously balance multiple competing constraints rather than optimizing for a single metric.

(7) **Efficiency-Accuracy Trade-offs**: Our analysis documented consistent trade-offs between computational efficiency and model accuracy across different optimization approaches, highlighting the importance of application-specific optimization strategies.

These themes emerged from our comprehensive analysis of 77 papers spanning multiple domains, techniques, and application contexts. The quantitative evidence we have presented - ranging from the 42.7% training time reduction achieved by Wang et al. [53] to the 73.8% memory efficiency improvement demonstrated by Lin et al. [34] - illustrates the significant potential of computational optimization approaches to enhance deep learning on big data.

Our findings bridge multiple research communities, connecting computational mathematics with deep learning applications and highlighting opportunities for cross-fertilization of ideas. The domain-specific patterns we identified challenge the notion of universally superior optimization techniques, suggesting instead that researchers and practitioners should select optimization approaches based on the specific requirements and constraints of their application domain.

Research Gaps and Future Directions: Our analysis reveals several critical gaps in the current literature:

- **Theoretical Foundation Gap:** Despite widespread adoption, many nature-inspired algorithms lack rigorous theoretical analysis of convergence properties and performance bounds. Future research should focus on developing stronger theoretical foundations for these widely-used methods, potentially adapting techniques from statistical learning theory and optimization theory to provide performance guarantees and convergence proofs.

- **Empirical Validation Gap:** We identified limited standardization in evaluation methodologies, making direct comparisons between optimization approaches challenging. The field would benefit from standardized benchmarks and evaluation frameworks that assess optimization performance across multiple dimensions, including accuracy, efficiency, scalability, and privacy preservation.

- **Hardware-Algorithm Integration Gap:** While hardware-aware optimization shows promise, further research is needed to develop frameworks that jointly optimize algorithm design and hardware implementation. Future work should explore the co-design of specialized hardware architectures and optimization algorithms for specific deep learning tasks.

- **Privacy-Performance Trade-off Gap:** More work is needed to quantify and optimize the trade-offs between privacy guarantees and model performance. Future research should explore techniques for improving the privacy-utility frontier, potentially leveraging recent

advances in cryptography and differential privacy to provide stronger privacy guarantees with minimal impact on model performance.
- **Cross-Domain Knowledge Transfer:** Our findings on domain-specific optimization patterns highlight a need for research on transferring optimization knowledge across domains. Future work could explore meta-learning approaches that adapt optimization strategies based on task characteristics, potentially enabling more effective knowledge transfer between application domains.

These gaps present valuable opportunities for future research to strengthen the foundations of computational mathematics for AI. By addressing these gaps, the research community can develop more robust, efficient, and widely applicable optimization approaches for deep learning on big data.

Practical Implications: The patterns and themes identified in this review provide valuable guidance for researchers and practitioners working to develop and deploy deep learning systems on big data. By understanding the strengths and limitations of different optimization approaches across various application domains, researchers can make more informed decisions about which methods to employ for specific deep learning tasks and computing environments.

For practitioners, our findings suggest several practical recommendations:

- Consider domain-specific optimization patterns when selecting optimization approaches
- Evaluate the trade-offs between computational efficiency and model accuracy for your specific application requirements
- Incorporate hardware-awareness into optimization strategy selection, particularly for resource-constrained environments
- Address privacy concerns early in the optimization process rather than as an afterthought
- Adopt multi-objective optimization frameworks that can balance competing constraints relevant to your application context

As computational resources continue to evolve and deep learning models grow in complexity, the field of computational mathematics for AI will remain critical for enabling the next generation of intelligent systems capable of extracting meaningful insights from big data. Our systematic review provides a comprehensive foundation for understanding the current state of this rapidly evolving field and identifying promising directions for future research and development.

6 Notes

There was a challenge among researchers to detect big data or what constitutes big data. While some studies did run their numerical method against a large set of data, it was not always clear if it was big data.

For instance one paper discussed fault prediction with a large amount of data, but it did not occur naturally to us that this data could be big data. It was only clarified during the discussion phase that the data was indeed big data.

7 Conclusion

This systematic analysis of computational mathematics for AI optimization synthesizes findings from 77 studies (2016-2024) examining numerical methods and distributed computing techniques for deep learning on big data. The research reveals three fundamental dimensions that characterize the current state of the field and its future trajectory.

7.1 Domain-Specific Optimization Patterns

The analysis demonstrates clear domain-dependent preferences in optimization approaches, each yielding significant performance improvements. Healthcare applications, representing nearly one-third of the examined studies, show a strong preference for nature-inspired algorithms, which account for 62.5% of optimization techniques in this domain. These methods prove particularly effective for medical imaging and disease prediction, delivering 27.4% accuracy gains when handling complex, high-dimensional medical data. In cybersecurity applications, Bayesian methods dominate with 53.8% adoption, providing crucial uncertainty quantification that reduces false alarms by 22.3% while maintaining 93.7% recall rates. Financial services applications favor evolutionary algorithms (41.2% adoption), which excel at multi-objective optimization challenges like portfolio management, achieving 23.5% improvements in risk-adjusted returns.

7.2 Distributed Computing Advancements

The study identifies significant progress in distributed computing approaches that enable scaling deep learning to big data problems. Federated learning systems demonstrate remarkable scalability, maintaining efficiency up to 800 client nodes before communication overhead and statistical heterogeneity begin diminishing returns. Hybrid parallelism strategies combining data, pipeline, and tensor approaches achieve near-linear scaling efficiency when distributing models across 64 GPUs, making billion-parameter models tractable. Communication optimization techniques yield particularly impressive results, with graduated compression methods reducing bandwidth requirements by 73% without compromising model quality. Energy efficiency improvements reach 58.4% in edge computing deployments through adaptive computation techniques that dynamically adjust model complexity based on input difficulty.

7.3 Key Trade-offs and Research Gaps

The analysis reveals several critical tensions that define current challenges in the field. Nature-inspired algorithms dominate practical adoption (42% of approaches) despite scoring lowest in theoretical rigor (2.8/12), while gradient-based methods show the strongest theoretical foundations (8.7/12) but see limited adoption (12%). This theory-practice gap raises important questions about the reliability and generalizability of widely-used methods. Performance trade-offs emerge prominently in memory optimization, where 73.8% reductions in memory requirements sometimes come at the cost of 5.8% accuracy degradation. Privacy-preserving techniques demonstrate that formal guarantees can be achieved with less than 3% accuracy loss, though implementation challenges remain.

The study identifies several reproducibility challenges, with only 23.5% of nature-inspired algorithm studies providing source code compared to 67.2% for gradient-based methods. Scaling limitations become apparent in federated learning systems, where efficiency declines beyond 800 nodes due to statistical heterogeneity effects. Energy consumption remains problematic for large-scale deployments despite 58.4% improvements, highlighting the need for continued innovation in efficient computing.

7.4 Future Directions and Practical Implications

The findings suggest several critical directions for future research. Strengthening theoretical foundations for widely-used nature-inspired approaches could bridge the current theory-practice gap, while standardized benchmarks could address reproducibility challenges in distributed training. Developing adaptive federated learning frameworks capable of scaling beyond 800 nodes represents

another key opportunity. Hardware-aware optimizations through algorithm-architecture co-design may yield further efficiency gains, particularly for edge deployments.

For practitioners, the results provide clear guidance: healthcare applications should prioritize nature-inspired methods for complex medical data, while cybersecurity systems benefit most from Bayesian approaches with uncertainty quantification. Financial models achieve optimal results with evolutionary multi-objective optimization techniques. Edge deployments require communication-efficient federated learning with privacy guarantees, whereas cloud and HPC systems need hybrid parallelism strategies for training billion-parameter models.

This comprehensive analysis reveals a field in transition, moving from general-purpose solutions to specialized, context-aware optimization frameworks. While distributed computing innovations have dramatically improved scalability, fundamental tensions remain between theoretical rigor and empirical performance. Addressing these challenges while continuing to advance scalability, efficiency, and privacy preservation will define the next era of computational mathematics for AI.

A List of Included Papers

Table 5 presents the comprehensive list of all 77 papers included in this SLR. These papers were selected based on the inclusion criteria and quality assessment process detailed in the methodology. Each study contributes to the understanding of computational mathematics for AI with focus on numerical methods and distributed computing techniques for deep learning on big data.

Table 5. Complete List of Included Studies

ID	Title	Authors
1	DeepLoc: A Deep Neural Network-based Indoor Positioning Framework	S. Liu, Q. Ren, J. Li, H. Xu
2	A Communication-Efficient Federated Learning Scheme for IoT-Based Traffic Forecasting	C. Zhang, L. Cui, S. Yu, J. J. Q. Yu
3	Fault Diagnosis Method of Link Control System for Gravitational Wave Detection	A. Gao, S. Xu, Z. Zhao, H. Shang, R. Xu
4	Multi disease-prediction framework using hybrid deep learning: an optimal prediction model	Ampavathi A., Saradhi T.V.
5	WOA + BRNN: An imbalanced big data classification framework using Whale optimization and deep neural network	Hassib E.M., El-Desouky A.I., Labib L.M., El-kenawy E.-S.M.
6	A Novel Resource Optimization Algorithm Based on Clustering and Improved Differential Evolution Strategy Under a Cloud Environment	Zhou Z., Li FM., Yang SQ
7	Meta-Heuristic Optimization of LSTM-Based Deep Network for Boosting the Prediction of Monkeypox Cases	Eid MM., El-Kenawy EM., Khodadadi N., Mirjalili S., Khodadadi E., Abotaleb M., Al-harbi AH., Abdelhamid AA., Ibrahim A., Amer GM., Kadi A., Khafaga DS
8	Support Vector Regression Integrated with Fruit Fly Optimization Algorithm for River Flow Forecasting in Lake Urmia Basin	Samadianfard S., Jarhan S., Salwana E., Mosavi A., Shamshirband S., Akib S
9	Hybrid Optimization Algorithm for Detection of Security Attacks in IoT-Enabled Cyber-Physical Systems	A. Sagu, N. S. Gill, P. Gulia, I. Priyadarshini, J. M. Chatterjee
10	SuperMeshing: Boosting the Mesh Density of Stress Field in Plane-Strain Problems Using Deep Learning Method	H. Xu, Z. Nie, Q. Xu, Y. Li, F. Xie, X. Liu
11	A Comprehensive Survey on Training Acceleration for Large Machine Learning Models in IoT	H. Wang, Z. Qu, Q. Zhou, H. Zhang, B. Luo, W. Xu, S. Guo, R. Li
12	Unlocking the Power of Voice for Financial Risk Prediction: A Theory-Driven Deep Learning Design Approach	Yang Yi., Qin Yu, Fan Yangyang, Zhang Zhongju
13	Optimisation algorithm-based recurrent neural network for big data classification	Akhtar MM, Ahamad D, AlamHameed S

Table 5 – Continued from previous page

ID	Title	Authors
14	Exponential Chimp Optimization Algorithm based Deep Neuro-Fuzzy Network with MapReduce framework for fake news detection in big data analytics	Kanchanamala P, Selva Rani B, Vairamuthu S
15	Advanced Deep Learning Model for Predicting the Academic Performances of Students in Educational Institutions	Baniata LH, Kang S, Alsharaiah MA, Baniata MH
16	A TLBO-Tuned Neural Processor for Predicting Heating Load in Residential Buildings	Almutairi K, Algarni S, Alqahtani T, Moayedi H, Mosavi A
17	Semi-Supervised Discovery of DNN-Based Outcome Predictors from Scarcely-Labeled Process Logs	Folino Francesco, Folino Gianluigi, Guarascio Massimo, Pontieri Luigi
18	Creating Proactive Cyber Threat Intelligence with Hacker Exploit Labels: A Deep Transfer Learning Approach	Ampel Benjamin M., Samtani Sagar, Zhu Hongyi, Chen Hsinchun
19	Wearable Sensor-Based Chronic Condition Severity Assessment: An Adversarial Attention-Based Deep Multisource Multitask Learning Approach	Yu Shuo, Chai Yidong, Chen Hsinchun, Sherman Scott J., Brown Randall A.
20	A Deep Learning Approach for Recognizing Activity of Daily Living (ADL) for Senior Care: Exploiting Interaction Dependency and Temporal Patterns	Zhu Hongyi, Samtani Sagar, Brown Randall A., Chen Hsinchun
21	Prescriptive analytics systems revised: a systematic literature review from an information systems perspective	Christopher Wissuchek, Patrick Zschech
22	Tracking machine learning models for pandemic scenarios: a systematic review of machine learning models that predict local and global evolution of pandemics	Marcelo Benedeti Palermo, Lucas Micol Policarpo, Cristiano André da Costa, Rodrigo da Rosa Righi
23	Double-Target Based Neural Networks in Predicting Energy Consumption in Residential Buildings	Moayedi H, Mosavi A
24	Bayesian Optimization LSTM/bi-LSTM Network With Self-Optimized Structure and Hyperparameters for Remaining Useful Life Estimation of Lathe Spindle Unit	Thoppil NM, Vasu V, Rao CSP
25	Extended and optimized deep convolutional neural network-based lung tumor identification in big data	Ananth AD, Palanisamy C
26	Ensemble Random Forest-based Gradient Optimization based Energy Efficient Video Processing System for Smart Traffic Surveillance System	Rajagopal S, Devi MU, Jones GM, Nayagam MG
27	Unintended Emotional Effects of Online Health Communities: A Text Mining-Supported Empirical Study	Zhou Jiaqi, Zhang Qingpeng, Zhou Sijia, Li Xin, Zhang Xiaoquan (Michael)
28	An Intelligent Big Data Security Framework Based on AEFS-KENN Algorithms for the Detection of Cyber-Attacks from Smart Grid Systems	S. Muthubalaji, N. K. Muniyaraj, S. P. V. S. Rao, K. Thandapani, P. R. Mohan, T. Somasundaram, Y. Farhaoui
29	Sorting the Digital Stream: Big Data-driven Insights into Email Classification for Spam and Ham Detection	S. A. Shah, E. A. Arputham, A. Ahmed, M. B. Farah, A. Shah, A. Aziz
30	Individual Recognition of Big Data Radar Digital Waveform Based on Long Short-Term Memory Network	Y. Jiang, W. Sheng, D. Cheng, L. Xiang, R. Song, W. Jiang
31	Large-Scale Mobile App Identification Using Deep Learning	S. Rezaei, B. Kroencke, X. Liu
32	Big Vibration Data Diagnosis of Bearing Fault Base on Feature Representation of Autoencoder and Optimal LSSVM-CRO Classifier Model	V. Nguyen, T. Dung Hoang, V. Thai, X. Nguyen
33	Predictions of the Key Operating Parameters in Waste Incineration Using Big Data and a Multiverse Optimizer Deep Learning Model	Zhao Z., Zhou Z., Lu Y., Li Z., Wei Q., Xu H.
34	Hybrid Whale Tabu algorithm optimized convolutional neural network architecture for intrusion detection in big data	Ponmalar A., Dhanakoti V.
35	Hyperparameter Tuned Deep Learning Enabled Intrusion Detection on Internet of Everything Environment	Hamza M.A., Abdalla Hashim A.H., Mohamed H.G., Alotaibi S.S., Mahgoub H., Mehanna A.S., Motwakel A.
36	Big Data Analytics Assisted Arithmetic Optimization with Deep Learning Model for Sentiment Classification	Manivannan K., Suresh T., Parthiban M.

Table 5 – Continued from previous page

ID	Title	Authors
37	Evolutionary Algorithm Based Feature Subset Selection for Students Academic Performance Analysis	Babu I., Mathusoothana R., Kumar S.
38	A Novel Approach for Big Data Visualization: Combining and Integrating Machine Learning, Evolutionary Algorithm and Genetic Algorithm	Chandrasekaran D., Thiyagarajan Panneerselvam
39	Optimizing Energy Efficiency in Smart Home Using Deep Learning Reinforcement Models in Big Data Environment	Velvizhy P., Kanchana R., Bhargavi R.
40	A Hybrid Evolutionary Computing Based Clustering for Electricity Demand Prediction using Short-Term Load Forecasting	Vinodhini V., Gomathi Nayagam M., Rajalakshmi M.
41	Improved Butterfly Optimization-Based Feature Selection to Classify High-Dimensional Microarray and RNA-Seq Data	Ragavendar M.S., Rashimi Geetha G., Kalaiarasi G., Saravanan S.
42	Fast Convergence of Whale Algorithm Based on Chaotic Levy Flight	Malik E., Basanta Kumar P., Srikanta P., Debashree M., Ramkumar M.
43	Improved Whale Optimization Algorithm for Big Data using Neural Fuzzy and Moth Flame Optimizer Algorithms	Naga Sundaram J., Hemamalini K., Suresh Gnana Dhas C., Punitha K.
44	Butterfly optimization algorithm for big data analytics using hybrid deep belief networks	Neeba E.A., Koteeswaran S.
45	Graph-guided architecture search for QoT estimation of lightpaths	Ranjbar M., Cugini F., Woodward S., Paolucci F., Dallaglio M., Valcarenghi L.
46	DSSAE-BBOA: deep learning-based weather big data analysis and visualization	Madhukar Rao G., Dharavath Ramesh
47	Cross-correlation and forecast impact of public attention on USD/CNY exchange rate: Evidence from Baidu Index	Lin Y., Wang R., Gong X., Jia G.
48	An Intelligent Task Scheduling Model for Hybrid Internet of Things and Cloud Environment for Big Data Applications	Pal S., Jhanjhi N.Z., Abdulbaqi A.S., Akila D., Alsubaei F.S., Almazroi A.A.
49	Self-attention convolutional neural network optimized with season optimization algorithm Espoused Chronic Kidney Diseases Diagnosis in Big Data System	Sulthan Alikhan J., Alageswaran R., Miruna Joe Amali S.
50	Attack prevention in IoT through hybrid optimization mechanism and deep learning framework	Nagaraju R., Pentang J.T., Abdufattokhov S., CosioBorda R.F., Mageswari N., Uganya G.
51	Optimized Big Data Dissipation System Using Entropy and Improved Machine Learning Techniques for Cloud Forensics System	Kalaimannan E., Sharma A., Gupta R., Kumar S., Ali D., Prashant S.
52	Multi-Objective Sparrow Search and Grasshopper Optimization Based Load Balancing for Cloud Environment	Shanmugasundaram M., Thirugnanam K., Vidyasankar K.
53	Harris Hawk based Extreme Learning Machine with Attention Mechanism for Big Data Processing in Healthcare Analysis	John E., Gocila M., Sagayaraj Francis F.
54	Deep learning-based auto-encoder integrated fault identification using swarm-based coyote optimization algorithm	Kanathasan K., Thiruvenkadam S.
55	Hybrid Artificial Intelligence Based on Reinforcement Learning for Large-Scale Cyber-Physical Systems: Analysis of Trends and Future Directions	Luvuna Luanda N., Masinde M., Toussaint H.A.
56	Ensemble K-Means Clustering using a Mayfly Optimizer Method for Enhancing the Routing Efficiency in Mobile Ad-hoc Networks	Muthuvel R., Srinivasan K., Sivakumar P., Sivagurunathan P.T., Sarath Kumar B., Kannimuthu S., Batri K., Dhamodaran P.K.
57	Elephant Herding Optimization Applied to Enhance DBSCAN for Energy Effective Data Partitioning in Wireless Sensor Networks	NagaLakshmi L., Vairamuthu S., Dhamodaran P.K.
58	A new hybrid metaheuristic optimizer for big data classification in internet of things applications	Bhavatharani A., Amudhavel J., Mahendran S.A., Prabu Kumar C.C., Rajakumar P.
59	Rider-Deep Belief Network-Based MapReduce Framework for Big Data Classification	Gujjeti S., Pabboju S.
60	Convolutional Neural Network optimization using Modified Elitist Grey Wolf Algorithm for Energy consumption prediction	Mohapatra S., Sarangi S.K.

Table 5 – Continued from previous page

ID	Title	Authors
61	AI and Big Data of Criminal Activities: A Perspective on Encryption Standards	Parry G., Gangadharan N., Deebak B.D., AlZubi A.A., Alkhayyat A.
62	Cuckoo Search: An Overview of Meta-Heuristic Algorithmic Technique	Mahalakshmi C., Anuja A.
63	Bio-inspired hybrid optimized techniques for effective intrusion detection in cloud computing environment	Anand Neela P.S., Padmanabhan B., Mohan K., Chockalingam S.P.
64	An optimized recurrent neural network with principal component analysis for big data in healthcare applications	Jansi K.L., Amutha B.
65	Glioma Classification and Tumor Segmentation from MRI using Deep Neural Network with Hybrid Optimization	Viknesh R.S., Venkatesan D., Jayasankar T., Elangovan D., Nayyar A., Meleppat R.K., Benjamin A.R., Dung V.
66	A hybrid metaheuristic algorithm for resource management in IoT clusters under fog computing	Nageswara Rao B., Priyadarsini S.K., Satyanarayana K.V.V.
67	Big Data Analytics through Multi-Objective Optimization with Optimized Online Mode Learning DNN for Credit Card Fraud Detection in Bank Financial Sector	Shameema Firdose S.V., Sivasubramanian S., Muhammedjamal A.S.
68	Multi-objective Scheduling Optimization in Big Data Processing: Status and challenges	Sunil Kumar A.V., Vishnu Kumar P., Mohammad Zubair K.
69	An optimized deep convolutional neural network model for automatic detection and classification of agricultural crop pests and diseases in IoT environment	Bhuvaneswari K., Lavanya R.
70	Bayesian-Based Hyperparameter Optimization of 1D-CNN for Structural Anomaly Detection	Li X., Guo H., Xu L., Xing Z.
71	Energy efficient hybrid approach for data collection in wireless sensor networks using Markov Meerkat algorithm	Pavan Kumar G.S., Poojita P., Palanisamy S.
72	Dragonfly-Firefly hybrid optimization algorithm for solving big data intrusion detection system in stock market environments	Satyanarayana N., Reddy P.B.
73	An automated prediction of remote sensing data of Queensland-Australia for flood and wildfire susceptibility using BISSOA-DBMLA scheme	Sankaran K., Sanjay Kumar M., Manikandan V.
74	Hybrid Anomaly Detection in Big Data Using IPSO-k-ANN Optimized DBSCAN Algorithm for Power Systems	Thirumaran D., Prasanna Kumar R.
75	Stochastic optimization using enhanced fruit fly algorithm for classification in big data healthcare environment	Krishnapriya S., Sarath Kumar B.
76	Modified deep learning model for effective and adaptive real-time lung status detection using big data analytics	Devan P.A.M., Akshaya V., Mohapriya R., Ananthy S., Gayathri Priyanka T.
77	ExpSSOA-Deep maxout: Exponential Shuffled shepherd optimization based Deep maxout network for intrusion detection using big data in cloud computing framework	Pandey B.K., M.R.M. V., Ahmad S., Rodriguez C., Esenarro D.

References

[1] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. 2017. QSGD: Communication-Efficient SGD via Gradient Quantization and Encoding. doi:10.48550/arXiv.1610.02132 arXiv:1610.02132.

[2] Khalid Almutairi, Salem Algarni, Talal Alqahtani, Hossein Moayedi, and Amir Mosavi. 2022. A TLBO-Tuned Neural Processor for Predicting Heating Load in Residential Buildings. *Sustainability* 14, 10 (May 2022), 5924. doi:10.3390/su14105924

[3] Antony Dennis Ananth and Chenniappan Palanisamy. 2022. Extended and Optimized Deep Convolutional Neural Network-Based Lung Tumor Identification in Big Data. *International Journal of Imaging Systems and Technology* 32, 3 (2022), 918–934. doi:10.1002/ima.22667

[4] Pouya Ataei, Sri Regula, Daniel Staegemann, and Saurabh Malgaonkar. 2024. Filtering Useful App Reviews Using Naïve Bayes—Which Naïve Bayes? *AI* 5, 4 (2024), 2237–2259.

[5] Simon Ben and Jenna Waller. 2019. Demystifying deep learning optimization in big data contexts. *Artificial Intelligence Review* 53, 4 (2019), 3197–3225.

[6] Thomas Bäck. 2020. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press, New York.

[7] Maurizio Capra, Beatrice Bussolino, Alberto Marchisio, Guido Masera, Maurizio Martina, and Muhammad Shafique. 2020. Hardware and Software Optimizations for Accelerating Deep Neural Networks: Survey of Current Trends, Challenges, and the Road Ahead. *IEEE Access* 8 (2020), 225134–225180. doi:10.1109/ACCESS.2020.3039858

[8] Tao Chen, Shichen Yang, and Jianhua Li. 2016. Specialized Genetic Algorithm for Neural Network Architecture Optimization. *IEEE Transactions on Evolutionary Computation* 20, 3 (2016), 331–343. doi:10.1109/TEVC.2015.2457491

[9] Harris M. Cooper. 1988. Organizing knowledge syntheses: A taxonomy of literature reviews. *Knowledge in Society* 1, 1 (March 1988), 104–126. doi:10.1007/BF03177550

[10] N. Dalkey. 1969. An experimental study of group opinion. *Futures* 1, 5 (Sept. 1969), 408–426. doi:10.1016/S0016-3287(69)80025-X

[11] Yann N. Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. 2014. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2* (Montreal, Canada) *(NIPS'14)*. MIT Press, Cambridge, MA, USA, 2933–2941.

[12] Jeffrey Dean, Greg S. Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Quoc V. Le, Mark Z. Mao, Marc'Aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, and Andrew Y. Ng. 2012. Large scale distributed deep networks. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1* (Lake Tahoe, Nevada) *(NIPS'12)*. Curran Associates Inc., Red Hook, NY, USA, 1223–1231.

[13] Jeffrey Dean and Sanjay Ghemawat. 2008. MapReduce: simplified data processing on large clusters. *Commun. ACM* 51, 1 (2008), 107–113.

[14] Kalyanmoy Deb and Himanshu Jain. 2014. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Transactions on Evolutionary Computation* 18, 4 (2014), 577–601. doi:10.1109/TEVC.2013.2281534

[15] Andre L Delbecq and Andrew H Van de Ven. 1971. A Group Process Model for Problem Identification and Program Planning. *Journal of Applied Behavioral Science* 7, 4 (1971), 466–492.

[16] Marwa M. Eid, El-Sayed M. El-Kenawy, Nima Khodadadi, Seyedali Mirjalili, Ehsaneh Khodadadi, Mostafa Abotaleb, Amal H. Alharbi, Abdelaziz A. Abdelhamid, Abdelhameed Ibrahim, Ghada M. Amer, Ammar Kadi, and Doaa Sami Khafaga. 2022. Meta-Heuristic Optimization of LSTM-Based Deep Network for Boosting the Prediction of Monkeypox Cases. *Mathematics* 10, 20 (Oct. 2022), 3845. doi:10.3390/math10203845

[17] Kathryn Fitch, Steven J. Bernstein, and Mary Dolores Aguilar. 2000. *The Rand/Ucla Appropriateness Method User's Manual*. RAND Corporation, Santa Monica.

[18] Simon Fong, Suash Deb, and Raymond Wong. 2023. Adaptive Mutation Rates in Genetic Algorithms for Deep Learning. *IEEE Transactions on Artificial Intelligence* 4, 2 (2023), 289–302. doi:10.1109/TAI.2022.3209572

[19] Zoubin Ghahramani. 2015. Probabilistic machine learning and artificial intelligence. *Nature* 521, 7553 (2015), 452–459. doi:10.1038/nature14541

[20] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press, Cambridge, MA.

[21] Song Han, Huizi Mao, and William J. Dally. 2016. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. doi:10.48550/arXiv.1510.00149 arXiv:1510.00149.

[22] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. doi:10.48550/arXiv.1503.02531 arXiv:1503.02531.

[23] Mohammed Amine Janati Idrissi, Hassan Ramchoun, Youssef Ghanou, and Mohamed Ettaouil. 2016. Genetic algorithm for neural network architecture optimization. In *2016 3rd International Conference on Logistics Operations Management (GOL)*. IEEE, Fez, Morocco, 1–4. doi:10.1109/GOL.2016.7731699

[24] K. Kanchanamala, P.V. Rao, and S.C. Guntuku. 2023. Exponential Chimp Optimization Algorithm for optimizing deep neuro-fuzzy networks in MapReduce frameworks for fake news detection. *Expert Systems with Applications* 217 (2023), 119611.

[25] Hyunjoon Kim, Joonho Park, and Sunghyun Lee. 2022. Hardware-Aware Optimization Techniques for Inference Latency Reduction. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 41, 8 (2022), 2567–2580. doi:10.1109/TCAD.2021.3089276

[26] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015*. arXiv:1412.6980.

[27] Barbara Kitchenham. 2004. *Procedures for Performing Systematic Reviews*. Keele University. Technical Report TR/SE-0401. Keele University, Department of Computer Science, Keele University, UK.

[28] Barbara Ann Kitchenham and Stuart Charters. 2007. *Guidelines for performing Systematic Literature Reviews in Software Engineering*. Technical Report EBSE 2007-001. Keele University and Durham University Joint Report. https://www.elsevier.com/__data/promis_misc/525444systematicreviewsguide.pdf

[29] Aaron Klein, Stefan Falkner, Simon Bartels, Philipp Hennig, and Frank Hutter. 2017. Fast Bayesian Optimization of Machine Learning Hyperparameters on Large Datasets. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 54)*, Aarti Singh and Jerry Zhu (Eds.). PMLR, 528–536. https://proceedings.mlr.press/v54/klein17a.html

[30] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2017. ImageNet classification with deep convolutional neural networks. *Commun. ACM* 60, 6 (May 2017), 84–90. doi:10.1145/3065386

[31] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.

[32] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444. doi:10.1038/nature14539

[33] Xia Li, Yang Liu, Tian Li, and Hong Qin. 2020. A survey on scalable deep learning techniques. *Journal of Big Data* 7, 1 (2020), 1–41.

[34] Yuxiang Lin, Zhilin Wang, and Kai Chen. 2022. Gradient Checkpointing Approach for Large Language Models. *Advances in Neural Information Processing Systems* 35 (2022), 15789–15801.

[35] James Martens. 2010. Deep learning via Hessian-free optimization. In *Proceedings of the 27th International Conference on International Conference on Machine Learning* (Haifa, Israel) *(ICML'10)*. Omnipress, Madison, WI, USA, 735–742.

[36] Ruben Mayer and Hans-Arno Jacobsen. 2021. Scalable Deep Learning on Distributed Infrastructures: Challenges, Techniques, and Tools. *Comput. Surveys* 53, 1 (Jan. 2021), 1–37. doi:10.1145/3363554

[37] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 54)*, Aarti Singh and Jerry Zhu (Eds.). PMLR, 1273–1282. https://proceedings.mlr.press/v54/mcmahan17a.html

[38] David Moher, Alessandro Liberati, Jennifer Tetzlaff, Douglas G Altman, and PRISMA Group. 2009. Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement. *PLoS medicine* 6, 7 (2009), e1000097.

[39] Grégoire Montavon, Geneviève B. Orr, and Klaus-Robert Müller. 2012. *Neural networks: tricks of the trade* (2nd ed ed.). Springer, Berlin New York.

[40] Maryam Mohammadi Najafabadi, Flavio Villanustre, Taghi M Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. 2015. Deep learning applications and challenges in big data analytics. *Journal of Big Data* 2, 1 (2015), 1–21.

[41] Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, Amar Phanishayee, and Matei Zaharia. 2021. Efficient large-scale language model training on GPU clusters using megatron-LM. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. ACM, St. Louis Missouri, 1–15. doi:10.1145/3458817.3476209

[42] Jongsoo Park, Minjia Yu, and Tao Zhao. 2022. Adaptive Computation Techniques for Energy Efficiency in Deep Learning. *IEEE Journal on Selected Areas in Communications* 40, 1 (2022), 139–153. doi:10.1109/JSAC.2021.3118346

[43] Sebastian Ruder. 2017. An overview of gradient descent optimization algorithms. doi:10.48550/arXiv.1609.04747 arXiv:1609.04747.

[44] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. *Nature* 323, 6088 (1986), 533–536.

[45] Amit Sagu, Nasib Singh Gill, Preeti Gulia, Ishaani Priyadarshini, and Jyotir Moy Chatterjee. 2025. Hybrid Optimization Algorithm for Detection of Security Attacks in IoT-Enabled Cyber-Physical Systems. *IEEE Transactions on Big Data* 11, 1 (Feb. 2025), 35–46. doi:10.1109/TBDATA.2024.3372368

[46] Saeed Samadianfard, Salar Jarhan, Ely Salwana, Amir Mosavi, Shahaboddin Shamshirband, and Shatirah Akib. 2019. Support Vector Regression Integrated with Fruit Fly Optimization Algorithm for River Flow Forecasting in Lake Urmia Basin. *Water* 11, 9 (Sept. 2019), 1934. doi:10.3390/w11091934

[47] Alexander Sergeev and Mike Del Balso. 2018. Horovod: fast and easy distributed deep learning in TensorFlow. doi:10.48550/arXiv.1802.05799 arXiv:1802.05799.

[48] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. 2012. Practical Bayesian optimization of machine learning algorithms. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2* (Lake Tahoe, Nevada) *(NIPS'12)*. Curran Associates Inc., Red Hook, NY, USA, 2951–2959.

[49] Ruoyu Sun, Zhi-Quan Luo, and Yinyu Ye. 2019. Optimization for deep learning: theory and algorithms. arXiv:1912.08957 [math.OC] https://arxiv.org/abs/1912.08957 arXiv preprint arXiv:1912.08957.

[50] Neil Thompson, Kristjan Greenewald, Keeheon Lee, and Gabriel F. Manso. 2023. The Computational Limits of Deep Learning. In *Ninth Computing within Limits 2023*. LIMITS, Virtual. doi:10.21428/bf6fb269.1f033948 Does not come with page numbers.

[51] Nikhil M Thoppil, Velagapudi Vasu, and CSP Rao. 2022. Bayesian optimization LSTM/bi-LSTM network with self-optimized structure and hyperparameters for remaining useful life estimation of lathe spindle unit. *Journal of Computing*

and Information Science in Engineering 22, 2 (2022), 021012.

[52] Nikhil M. Thoppil, V. Vasu, and C. S. P. Rao. 2022. Bayesian Optimization LSTM/bi-LSTM Network With Self-Optimized Structure and Hyperparameters for Remaining Useful Life Estimation of Lathe Spindle Unit. *Journal of Computing and Information Science in Engineering* 22, 2 (April 2022), 021012. doi:10.1115/1.4052838

[53] Shuiqiang Wang, Li Zhang, and Xiaoming Chen. 2021. Enhanced Adam Optimizer for Deep Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems* 32, 7 (2021), 3025–3039. doi:10.1109/TNNLS.2020.3004080

[54] Wei Qi Yan. 2023. *Computational Methods for Deep Learning: Theory, Algorithms, and Implementations.* Springer Nature Singapore, Singapore. doi:10.1007/978-981-99-4823-9

[55] Xin-She Yang. 2021. *Nature-inspired optimization algorithms* (second edition ed.). Academic Press, London San Diego, CA.

[56] Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica. 2010. Spark: cluster computing with working sets. In *Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing* (Boston, MA) *(HotCloud'10)*. USENIX Association, USA, 10.

[57] Chaoyun Zhang, Paul Patras, and Hamed Haddadi. 2019. Deep learning in mobile and wireless networking: A survey. *IEEE Communications surveys & tutorials* 21, 3 (2019), 2224–2287.

[58] Hongming Zhang, M Dehghani, and Z Yazdanparast. 2023. From distributed machine to distributed deep learning: a comprehensive survey. *Journal of Big Data* 10, 1 (2023), 158.

[59] Wei Zhang, Xiaofeng Lin, and Jiayi Chen. 2022. Privacy-Preserving Federated Learning for IoT Edge Intelligence. *IEEE Internet of Things Journal* 9, 12 (2022), 9876–9889. doi:10.1109/JIOT.2021.3135426

[60] Yuhao Zhang, Frank McQuillan, Nandish Jayaram, Nikhil Kak, Ekta Khanna, Orhan Kislal, Domino Valdano, and Arun Kumar. 2021. Distributed deep learning on data systems: a comparative analysis of approaches. *Proceedings of the VLDB Endowment* 14, 10 (June 2021), 1769–1782. doi:10.14778/3467861.3467867

[61] Meng Zhou. 2021. Heuristic Hyperparameter Optimization for Convolutional Neural Networks using Genetic Algorithm. doi:10.48550/arXiv.2112.07087 arXiv:2112.07087.

[62] Zhou Zhou, Fangmin Li, and Shuiqiao Yang. 2021. A Novel Resource Optimization Algorithm Based on Clustering and Improved Differential Evolution Strategy Under a Cloud Environment. *ACM Transactions on Asian and Low-Resource Language Information Processing* 20, 5 (Sept. 2021), 1–15. doi:10.1145/3462761