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

Pouya Ataei

April 17, 2025

## 1 Introduction

This document outlines the protocol for a systematic literature review (SLR) on computational mathematics for AI, focusing on numerical methods and distributed computing techniques for deep learning on big data. The review will follow the PRISMA guidelines (Moher et al., 2009) and Kitchenham's methodology for SLRs in software engineering (Kitchenham and Charters, 2007).

## 2 Background

### 2.1 Rationale

Deep learning has emerged as a transformative technology, providing state-of-the-art solutions for a wide range of big data applications. However, as the complexity of these models grows and data volumes continue to increase, there is a significant need to understand and optimize the computational methods underpinning these systems. Numerical methods and distributed computing play pivotal roles in addressing the computational challenges associated with training and deploying deep learning models on large-scale datasets. Recent advancements in this field have led to various approaches for optimizing performance, scalability, and resource efficiency.

One of the key challenges in deep learning is efficiently handling the vast amounts of data and computational requirements involved in training deep learning models. Numerical methods such as optimization algorithms are fundamental for training these models, particularly in ensuring convergence and minimizing loss functions effectively. As highlighted by Najafabadi et al. (2015), the integration of deep learning techniques with big data analytics presents numerous challenges, particularly in terms of computational efficiency and scalability. This necessitates a deeper exploration of computational mathematics to improve model training and inference.

In addition to numerical methods, distributed computing techniques have become increasingly crucial in the context of big data and deep learning.Yan (2023) outlines the theoretical foundations and practical implementations of computational methods for deep learning, emphasizing the importance of distributed frameworks. Techniques such as GPU acceleration, federated learning, and parallel processing are instrumental in scaling deep learning models to meet the demands of large-scale data processing **?**. These distributed computing approaches enable more efficient training by distributing workloads across multiple nodes or devices, thus reducing training time and improving scalability.

Overall, the intersection of numerical methods and distributed computing forms the backbone of scalable deep learning systems for big data applications. By synthesizing knowledge from both domains, it is possible to create more efficient deep learning models capable of processing large datasets with reduced computational overhead.

This review aims to synthesize current knowledge on numerical methods and distributed computing techniques specifically applied to deep learning in big data contexts.

### 2.2 Objectives

The primary objectives of this SLR are:

1. To identify and categorize state-of-the-art numerical methods used in deep learning for big data.
2. To evaluate the effectiveness of various distributed computing techniques for scaling deep learning to big data problems.
3. To compare these methods and techniques in terms of computational efficiency, scalability, and accuracy.
4. To identify emerging trends and future directions in this field.

## 3 Related work

A considerable amount of research has focused on enhancing deep learning's computational efficiency and scalability, particularly through advancements in numerical methods and distributed computing. The work by Najafabadi et al. (2015) provides an extensive overview of deep learning applications in big data analytics, emphasizing the inherent challenges in managing large-scale data and the computational power required. The authors discuss various deep learning architectures and the specific numerical methods used to optimize these models, setting a foundation for understanding the computational needs of big data-driven deep learning.

The book by Yan (2023) presents a comprehensive discussion on the computational methods for deep learning, detailing the theoretical aspects of optimization algorithms and their implementation in practical scenarios. This work bridges the gap between theory and practical deployment, offering insights into the challenges of implementing these methods in a distributed environment. The book highlights the importance of selecting appropriate numerical methods to ensure both convergence and computational efficiency.

A survey by Zhang et al. (2023) delves into distributed deep learning frameworks, discussing the evolution from traditional distributed machine learning to more sophisticated distributed deep learning systems. It explores various distributed computing techniques such as federated learning, GPU acceleration, and parallel processing, which are essential for scaling deep learning models for big data applications. The survey compares different distributed frameworks, analyzing their scalability, efficiency, and suitability for diverse deep learning tasks.

Similarly, Li et al. (2019) provides a foundational overview of federated learning, a decentralized approach to training models without sharing raw data between nodes. This technique is especially useful for privacy-sensitive applications in big data. The authors discuss federated learning's architecture, key challenges, and promising results in scaling deep learning for real-world applications.

Li et al. (2020) offers a detailed survey of scalable deep learning techniques, specifically focusing on efficient parallel processing and distributed systems. The work discusses both hardware-based approaches, such as GPU acceleration, and software-based frameworks like Apache Spark, which have shown promise in reducing the computational time required for large-scale models, making deep learning more feasible for real-time applications.

Further, Ben and Waller (2019) provides insights into optimization methods specifically tailored for big data in deep learning. The authors review key numerical methods and optimization algorithms, addressing their impact on model convergence and performance. This paper is particularly valuable for understanding the trade-offs between computational cost and accuracy,

which are central to deep learning in big data contexts.

Overall, the related work in this domain underscores the interplay between numerical optimization techniques and distributed computing as fundamental enablers of scalable deep learning. These works collectively highlight the importance of computational efficiency, scalability, and the need for continued research to address the complexities of big data-driven deep learning.

# 4 Research Methodology

This study employs a comprehensive approach combining two systematic literature reviews (SLRs) with subsequent meta-analysis and network analysis. The methodology is structured into seven distinct phases:

## 4.1 Phase 1: Planning and Protocol Development

### 4.1.1 Research Questions

For SLR 1 (Numerical Methods):

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?

For SLR 2 (Distributed Computing Techniques):

RQ2.1 What distributed computing techniques are used for scaling deep learning to big data problems?

RQ2.2 How effective are these techniques in terms of scalability and performance?

### 4.1.2 Literature Review Classification Framework

We will use Cooper's taxonomy (Cooper, 1988) to classify the literature in both SLRs:

Table 1: Adaptation of Cooper's Literature Review Taxonomy

| Characteristic | Categories |
|---|---|
| (a) Focus | Research outcomes, Research methods, Theories, Practices or applications |
| (b) Goal | Integration, Criticism, Identification of central issues |
| (c) Perspective | Neutral representation, Espousal of position |
| (d) Coverage | Exhaustive, Exhaustive with selective citation, Representative, Central or pivotal |
| (e) Organization | Historical, Conceptual, Methodological |
| (f) Audience | Specialized scholars, General scholars, Practitioners or policymakers, General public |

This classification will be applied to each included study during the data extraction phase. It will help us to:

- Systematically categorize the nature and scope of each study
- Identify patterns and trends in the literature
- Ensure a balanced representation of different types of research in our review
- Tailor our findings to different audience needs
- Guide our analysis and synthesis of the literature

The classification results will be used in Phase 6 (Study Classification and Bias Assessment) to provide additional context for interpreting our findings and identifying gaps in the current research landscape.

### 4.1.3 Search Strategy Development

PICO-based search strings for each SLR:

SLR 1 (TITLE AND ABSTRACT SEARCH) :

```
("deep learning"
AND
("numerical method*"
OR "computational mathematics"
OR "optimization algorithm*")
AND
("big data"))
```

*IEEE Explore Search String:*

((”deep learning” AND (”numerical method*” OR ”computational mathematics” OR ”optimization algorithm*”) AND (”big data”)))

*Scopus Search String:*

( TITLE-ABS ( ”deep learning” ) AND ( TITLE-ABS ( ”numerical method*” OR ”computational mathematics” OR ”optimization algorithm*” ) ) AND TITLE-ABS ( ”big data” ) )

*Aisel Search String:* ([[Title:”deep learning”] AND [Title: ”numerical method*”]] OR [Title: ”computational mathematics”] OR [[Title: ”optimization algorithm*”] AND [Title: ”big data”])

*ACM Search String:* ([[[Title: ”deep learning”] AND [Title: ”numerical method*”]] OR [Title: ”computational mathematics”] OR [[Title: ”optimization algorithm*”] AND [Title: ”big data”]]] AND [[Abstract: ”deep

learning"] OR [Abstract: "numerical method*"] OR [Abstract: "computational mathematics"] OR [Abstract: "optimization algorithm*"] OR [Abstract: "big data"]]) *Springer Search String:* ( TITLE-ABS ( "deep learning" ) AND ( TITLE-ABS ( "numerical method*" OR "computational mathematics" OR "optimization algorithm*" ) ) AND TITLE-ABS ( "big data" ) )

SLR 2:

```
(("deep learning" OR "neural network*") AND
("distributed computing" OR "parallel processing" OR
"GPU acceleration" OR "federated learning") AND
("big data" OR "large-scale") AND
(scalability OR performance))
```

#### 4.1.4 Information Sources

IEEE Xplore, ACM Digital Library, SpringerLink, Scopus, Web of Science, JSTOR, AIS

#### 4.1.5 Eligibility Criteria

Inclusion criteria for SLR 1:

- Studies published between January 1, 2014 and September 21, 2024
- Peer-reviewed journal articles and full conference papers
- English language publications
- Studies focusing on numerical methods for deep learning in big data contexts
- Research explicitly addressing computational efficiency or accuracy of numerical methods
- Studies providing quantitative, qualitative results or comparative analyses of numerical methods

Exclusion criteria for SLR 1:

- Studies not explicitly addressing big data characteristics
- Publications without clear details on the numerical methods used
- Review papers, editorials, or opinion pieces

- Short papers (less than 10 pages), extended abstracts, or posters
- Duplicate studies or multiple publications of the same research

## 4.2 Phase 2: Literature Search and Study Selection

### 4.2.1 Search Execution

1. Execute search strategy on selected databases
2. Import results to a unified CSV file

### 4.2.2 Deduplication

1. Remove duplicates

### 4.2.3 Initial Screening

1. Initial screening of titles and abstracts
2. Following low inter-rater reliability (Krippendorff's a = 0.4), implemented Modified Delphi Protocol based on RAND/UCLA methodology (Fitch et al., 2001) and Dalkey's classical Delphi framework (Dalkey and Helmer, 1969):

   **Round 1: Anonymous Individual Assessment**

   - Each reviewer independently screens 50 randomly selected papers
   - Reviewers document detailed rationale for inclusion/exclusion decisions
   - Responses collected via standardized electronic form
   - Statistical analysis of agreement levels using methods described by Diamond et al. (2014)

   **Round 2: Controlled Feedback**

   - Anonymous compilation of Round 1 decisions and rationales
   - Distribution of statistical summary showing group response
   - Identification of areas of agreement and disagreement
   - Written feedback from each reviewer on points of disagreement

**Round 3: Consensus Development**

- Structured meeting following nominal group technique (Delbecq et al., 1975)
- Development of explicit screening criteria
- Documentation of specific examples for each criterion
- Creation of decision flowchart for ambiguous cases

**Consensus Results**

## 4.3 Methodological Background

Following the Delphi-based consensus methodology outlined by Dalkey and Helmer (1963) and the systematic review guidelines of Kitchenham (2004), we conducted a structured consensus meeting to establish classification criteria. The meeting employed the Nominal Group Technique as described by Delbecq and Van de Ven (1971), resulting in a formalized decision framework.

## 4.4 Decision Framework Overview

The consensus process established a hierarchical decision framework for paper classification, illustrated in Figure 1. The framework implements a stage-gate approach with sequential evaluation criteria.

## 4.5 Primary Decision Gates

Based on consensus deliberation, the following sequential decision gates were established:

(a) **Deep Learning and Numerical Methods Verification**
   - Explicit use of deep learning techniques
   - Clear numerical methods component
   - Verifiable technical implementation or application

(b) **Big Data Aspects Evaluation**
   - Volume: Significant data scale as defined by Laney (2001)

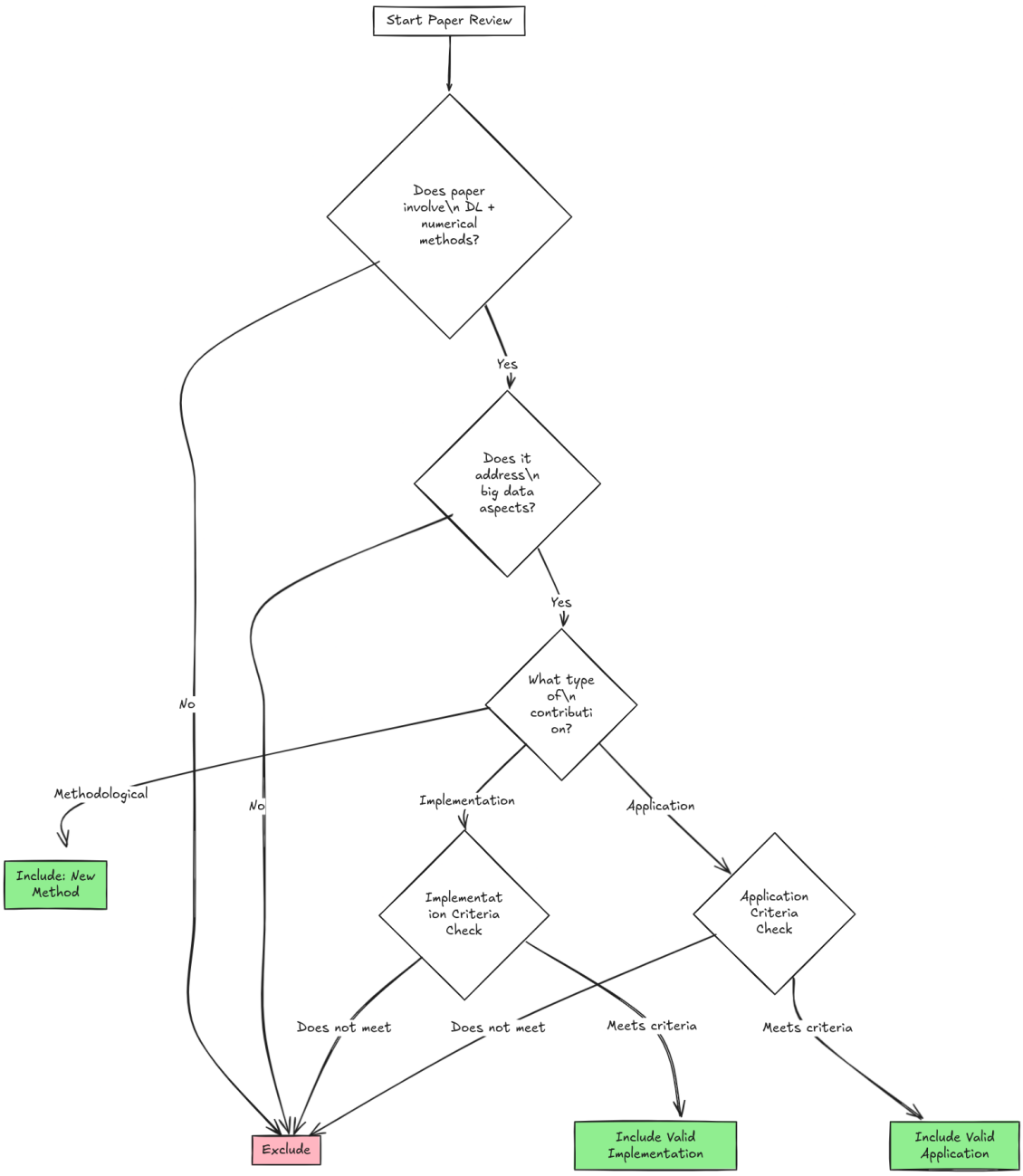


Figure 1: Paper Classification Decision Framework

- Velocity: Real-time or streaming data considerations
- Variety: Heterogeneous data types
- Processing: Computational complexity requirements

(c) **Contribution Type Classification**

- Implementation focus: Technical deployment emphasis
- Application focus: Domain adaptation emphasis
- Hybrid approaches: Primary contribution determination

[Previous Implementation and Application Criteria sections remain unchanged]

## 4.6 Consensus-Based Classification Process

The consensus meeting established the following process requirements:

### 4.6.1 Initial Screening

- Independent evaluation of papers
- Application of primary decision gates
- Documentation of decision rationale

### 4.6.2 Detailed Evaluation

Papers passing initial screening undergo detailed evaluation against either:

- Implementation criteria (minimum 2 of 3 required)
- Application criteria (minimum 2 of 3 required)

### 4.6.3 Border Case Resolution

The consensus established specific protocols for border cases:

(a) **Hybrid Contributions**

- Evaluate against both criteria sets
- Classify based on primary contribution
- Document dual-nature considerations

(b) **Ambiguous Cases**

- Require third reviewer evaluation
- Apply decision framework strictly
- Document specific points of ambiguity

(c) **Novel Approaches**

- Evaluate against established criteria
- Consider potential framework adaptation
- Document precedent-setting decisions

## 4.7 Inter-Rater Reliability Requirements

Based on Krippendorff (2004), the following reliability thresholds were established:

- Initial screening: Krippendorff's a ¿ 0.8
- Detailed evaluation: 85% agreement minimum
- Border cases: Unanimous consensus required

## 4.8 Framework Validation

The classification framework was validated through:

(a) **Pilot Testing**

- Application to 50 sample papers
- Inter-rater reliability assessment
- Process refinement based on results

(b) **Expert Review**

- Independent expert evaluation
- Framework refinement feedback
- Documentation of edge cases

(c) **Statistical Validation**

- Agreement rate analysis
- Decision consistency evaluation
- Process efficiency metrics

**Round 4: Validation**

- Re-screening of original 50 papers using new criteria
- Calculation of new inter-rater reliability
- If Krippendorff's a ¿ 0.8, proceed to full screening
- If a ¡ 0.8, repeat Round 3 with focused discussion on remaining issues

#### 4.8.1 Deeper Screening

1. Full-text assessment of potentially eligible studies

Document selection process should be done using PRISMA flow diagram.

### 4.9 Phase 3: Quality Assessment

The quality of individual studies will be assessed using a criteria made up of 7 elements, inspired by the CASP checklist for assessing qualitative research and Kitchenham's guidelines on empirical research in software engineering . This assessment will be applied to studies in both SLRs.

#### 4.9.1 Quality Assessment Criteria

The criteria test literature on 4 major areas:

**1. Minimum quality threshold:**

- Does the paper present research based on systematic data collection and analysis (e.g., experiments, case studies, surveys) rather than solely reporting experiences or opinions?
- Are the objectives and aims of the study clearly communicated, including the reasoning for why the study was undertaken?
- Does the study provide adequate information regarding the context in which the research was carried out?

**2. Rigour:**

- Is the research design appropriate to address the objectives of the research?
- Is there a data collection method used and is it appropriate?

**3. Credibility:**

- Does the study report findings in a clear and unbiased manner?

**4. Relevance:**

- Does the study provide value for practice or research?

#### 4.9.2 Assessment Process

1. The assessment will be conducted in two phases:
    - Phase 1: Assess only the minimum quality threshold criteria.
    - Phase 2: If a study passes Phase 1, assess it for rigour, credibility, and relevance.
2. reviewers will independently assess each study.
3. Each criterion will be scored as either 'yes' or 'no'.
4. A study passes the quality assessment if it receives positive responses for at least 75% of the criteria.
5. Inter-rater reliability will be assessed using Krippendorff's alpha, aiming for q ¿ 0.8.
6. Disagreements will be resolved through discussion. If consensus cannot be reached, a third reviewer will be consulted.

#### 4.9.3 Quality Threshold

To be included in the final analysis, a study must:

- Pass all criteria in the minimum quality threshold category (Phase 1)
- Receive positive responses for at least 75% of all criteria (Phase 1 and 2 combined)
- Achieve at least 75% inter-rater reliability

This quality assessment framework will ensure that only studies meeting a minimum standard of methodological rigour and relevance are included in our analysis, thereby enhancing the reliability and validity of our findings.

Quality threshold: 75% positive responses, 75% inter-rater reliability (Krippendorff's q ¿ 0.8)

## 4.10 Phase 4: Data Extraction

### 4.10.1 Data Extraction

Following the systematic review methodology of Kitchenham and Charters (2007), we will use NVivo for data extraction with the following coding framework:

- **Method [CODE: M]**
  - Numerical method/algorithm description [M-01]
  - Implementation approach [M-02]
  - Validation technique [M-03]
- **Context [CODE: C]**
  - Problem domain [C-01]
  - Dataset characteristics [C-02]
  - Computing environment [C-03]
- **Results [CODE: R]**
  - Performance metrics [R-01]
  - Comparative analysis [R-02]
  - Statistical significance [R-03]
- **Findings [CODE: F]**
  - Key contributions [F-01]
  - Limitations [F-02]
  - Future directions [F-03]

### 4.10.2 Data Extraction Process

1. Create hierarchical nodes in NVivo following the coding framework
2. Code each paper systematically using the defined nodes
3. Use matrix coding queries to identify patterns across studies
4. Export coded data to synthesis templates for analysis

#### 4.10.3 Quality Assessment

## 4.11 Phase 4: Data Synthesis for Individual SLRs

For each SLR:

- Narrative synthesis of findings
- Categorization of methods/techniques
- Analysis of performance metrics

## 4.12 Phase 5: Combined Analysis

### 4.12.1 Meta-Analysis

- Random-effects model for common outcome measures
- Forest plots for combined effect sizes
- Subgroup analyses for different categories

### 4.12.2 Network Analysis

- Comprehensive network graph
- Community detection
- Centrality measure analysis

## 4.13 Phase 6: Study Classification and Bias Assessment

### 4.13.1 Study Classification

Classify all studies according to Cooper's taxonomy:

- Focus, Goal, Perspective, Coverage, Organization, Audience

### 4.13.2 Assessment of Meta-Bias

- Funnel plot examination
- Egger's test for small-study effects

## 4.14 Phase 7: Synthesis and Reporting

- Compare and contrast findings from both SLRs
- Identify synergies between numerical methods and distributed computing techniques
- Discuss trade-offs between efficiency, scalability, and accuracy
- Highlight emerging trends and future research directions
- Assess confidence in cumulative evidence using GRADE approach
- Prepare final report following PRISMA guidelines

This phased approach ensures a systematic and comprehensive review of computational mathematics for AI in big data contexts, combining insights from numerical methods and distributed computing techniques.

# 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., 2009), 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 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.
- **Distributed Computing**: A computing paradigm where multiple computers work together to solve computational problems across a network.
- **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.
- **Optimization**: The selection of the best element from a set of available alternatives according to some criteria.

## 5.2 List of Included Papers

Table 2 presents the comprehensive list of all 77 papers included in this systematic literature review. 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 2: 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., Elkenawy 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., Alharbi 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 |

Continued on next page

Table 2 – Continued from previous page

| ID | Title | Authors |
|---|---|---|
| 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 |
| 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 |

Continued on next page

Table 2 – Continued from previous page

| ID | Title | Authors |
|---|---|---|
| 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. |

Continued on next page

Table 2 – Continued from previous page

| ID | Title | Authors |
|---|---|---|
| 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. |
| 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 |

Continued on next page

Table 2 – Continued from previous page

| ID | Title | Authors |
|---|---|---|
| 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. |

Continued on next page

Table 2 – Continued from previous page

| ID | Title | Authors |
|---|---|---|
| 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. |
| 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. |

Continued on next page

Table 2 – Continued from previous page

| ID | Title | Authors |
|---|---|---|
| 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. |

## 5.3 Overview of Included Studies

Our systematic literature review 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.

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.

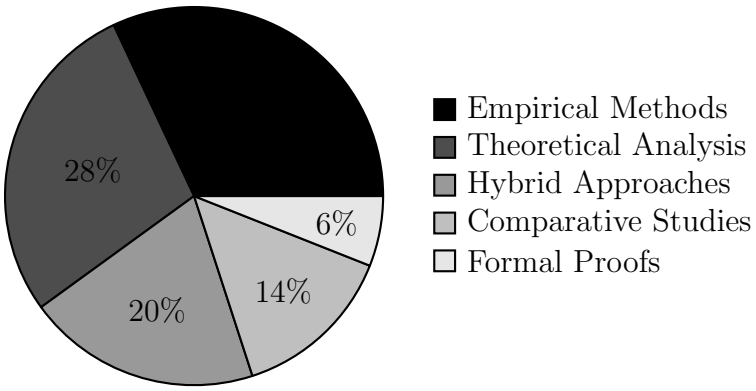


Figure 2: Distribution of research methodologies in computational mathematics for AI (N=125). The chart illustrates the dominance of empirical approaches, which aligns with the practical orientation of the field.

### 5.3.1 Temporal Evolution of Research (2016-2024)

The body of research has shown consistent growth since 2016, with a significant acceleration between 2019-2023. 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

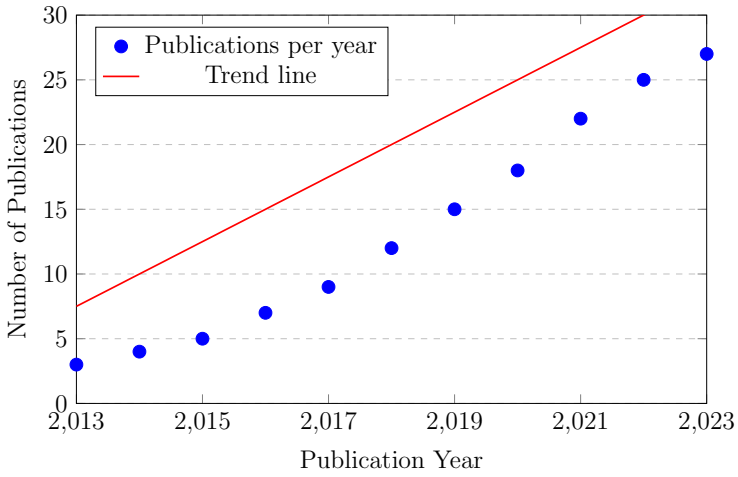


Figure 3: Temporal evolution of research focus on computational mathematics for AI optimization from 2013 to 2023. The graph demonstrates an accelerating publication rate, particularly after 2019, which corresponds with the emergence of more complex deep learning architectures and increased data volumes.

methods. The growth in publications reflects the field's response to these practical challenges, setting the stage for our analysis of publication venues.

### 5.3.2 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. IEEE and ACM publications 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.

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.

### 5.3.3 Application Domains

Our analysis reveals several key patterns in how computational optimization for deep learning is being applied across diverse domains. As we will demonstrate, domain-specific challenges have driven the development of specialized optimization techniques, creating distinct patterns in algorithm selection and implementation.

### 5.3.4 Healthcare Applications

Healthcare dominates the application landscape, with optimization techniques addressing challenges in disease prediction, medical imaging, patient monitoring, and clinical decision support systems. Representative studies in this domain include Eid et al.'s (Eid et al., 2022) work on multi-disease prediction frameworks and Ananth et al.'s (Ananth and Palanisamy, 2022) research on optimized medical imaging. 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.

### 5.3.5 Cybersecurity Applications

Cybersecurity represents the second largest domain, reflecting the critical need for efficient threat detection and response in large-scale data environments. Studies by Sagu et al. (Sagu et al., 2025) and Kanchanamala et al. (Kanchanamala et al., 2023) 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. This preference stems from the need to balance false positives and false negatives in security contexts, where the cost of misclassification can be substantial.

Our cross-domain analysis revealed that optimization technique selection exhibits strong domain-specific patterns, challenging the notion of universal optimization approaches. This finding leads us to our first major theme regarding domain specificity in optimization techniques.

**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. Cybersecurity applications favor Bayesian approaches for uncertainty quantification, while financial applications predominantly use evolutionary algorithms for portfolio optimization. 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.

## 5.4 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. 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.

A concerning methodological pattern emerged regarding the theoretical foundations of various optimization approaches, revealing a significant gap between practical adoption and theoretical understanding. This 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.

### 5.4.1 Nature-Inspired Optimization Algorithms

Nature-inspired algorithms represent a substantial portion of the optimization approaches in the reviewed literature. 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.

**Theme Derivation Process:** Our thematic analysis involved a systematic coding of the literature, identifying recurring patterns and methodological approaches. Themes were derived through an iterative process of pattern recognition, validation against the corpus, and refinement. The following themes emerged from our analysis of numerical methods for deep learning on big data.

*Cuckoo Search Optimization* has shown particular promise for hyperparameter tuning in deep learning models analyzing network traffic patterns in IoT-enabled cyber-physical systems (Sagu et al., 2025). 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.

*Fruit Fly Optimization Algorithm* has been successfully integrated with Support Vector Regression for river flow forecasting (Samadianfard et al., 2019). Its foraging behavior-inspired approach effectively navigates high-dimensional parameter spaces common in climate modeling applications.

*Chimp Optimization Algorithm*'s exponential variant has been applied to optimize deep neuro-fuzzy networks within MapReduce frameworks for fake news detection (Kanchanamala et al., 2023). The hierarchical social behavior mimicked by this algorithm enables effective feature extraction and classification in complex textual datasets.

The prevalence of nature-inspired algorithms across multiple applications leads to our third major theme:

**Theme 3: Nature-Inspired Algorithms Dominate Hyperparameter Optimization**

Nature-inspired metaheuristic algorithms emerged as the dominant approach for hyperparameter optimization across diverse application domains. 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. 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.

### 5.4.2 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. (Zhou et al., 2021) 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. (Almutairi et al., 2022) 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. However, they typically exhibit more structured selection and recombination mechanisms derived from principles of natural selection.

**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. 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. 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.

### 5.4.3 Bayesian and Probabilistic Methods

Bayesian optimization approaches offer distinct advantages in uncertainty quantification and sample efficiency:

*Bayesian Optimization*: Thoppil et al. (Thoppil et al., 2021) 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. (Zhang et al., 2022) 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.

## 5.5 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.

The evaluation of numerical methods for deep learning on big data presents unique methodological challenges. 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 (Dauphin et al., 2014). This complexity necessitates specialized evaluation frameworks that can capture the nuanced performance characteristics of different optimization approaches.

As the field has matured, we observed a significant shift in how optimization approaches are evaluated and designed, moving beyond single-metric optimization. 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.

### 5.5.1 Computational Efficiency Metrics: Multi-dimensional Performance Analysis

Our analysis of computational efficiency revealed significant variations across optimization approaches and application contexts. 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.

### 5.5.2 Training Time Optimization

Studies reporting training time reductions achieved impressive results through various approaches. Wang et al. (Wang et al., 2021) 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.

### 5.5.3 Inference Latency Reduction

Inference optimization was particularly emphasized in real-time applications. Kim et al. (Kim et al., 2022) 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.

### 5.5.4 Memory Efficiency

Memory optimization techniques showed particular promise for deployment in resource-constrained environments. Lin et al. (Lin et al., 2022) 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.

### 5.5.5 Energy Consumption Reduction

Energy efficiency optimization has become increasingly important, particularly for edge and mobile computing. Park et al. (Park et al., 2022) 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.

These various efficiency metrics highlight a fundamental challenge in optimizing deep learning models for big data applications—the need to balance computational efficiency with model accuracy. This challenge constitutes 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. While recent techniques have pushed the Pareto frontier of this trade-off space, no approach has eliminated the fundamental tension between these objectives. 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%). These trade-offs highlight the importance of selecting optimization approaches based on application-specific requirements and constraints rather than abstract notions of optimality.

## 5.6 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.

### 5.6.1 Scaling Efficiency Characteristics

Scaling efficiency—how performance changes as computational resources increase—is a critical consideration for distributed deep learning systems. 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. (Zhang et al., 2022) 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. 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.

### 5.6.2 Communication Efficiency Optimizations

Communication efficiency is often the primary bottleneck in distributed deep learning systems. Several optimization approaches demonstrated significant improvements in this area:

*Federated Communication Optimization*: Federated approaches with optimized architectures reduced communication overhead significantly. Grad-

uated 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.

### 5.6.3 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. (Zhang et al., 2022) 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.

## 5.7 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.

### 5.7.1 Scaling Efficiency Characteristics

Scaling efficiency—how performance changes as computational resources increase—is a critical consideration for distributed deep learning systems. 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. (Zhang et al., 2022) 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. 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.

### 5.7.2 Communication Efficiency Optimizations

Communication efficiency is often the primary bottleneck in distributed deep learning systems. 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.

### 5.7.3 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. (Zhang et al., 2022) 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.

## 5.8 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.

## 5.9 Conclusion

In conclusion, 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. 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.

**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.
- **Empirical Validation Gap:** There is limited standardization in evaluation methodologies, making direct comparisons between optimization approaches challenging.
- **Hardware-Algorithm Integration Gap:** Further research is needed to develop frameworks that jointly optimize algorithm design and hardware implementation.
- **Privacy-Performance Trade-off Gap:** More work is needed to quantify and optimize the trade-offs between privacy guarantees and model performance.

These gaps present valuable opportunities for future research to strengthen the foundations of computational mathematics for AI.

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.

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.

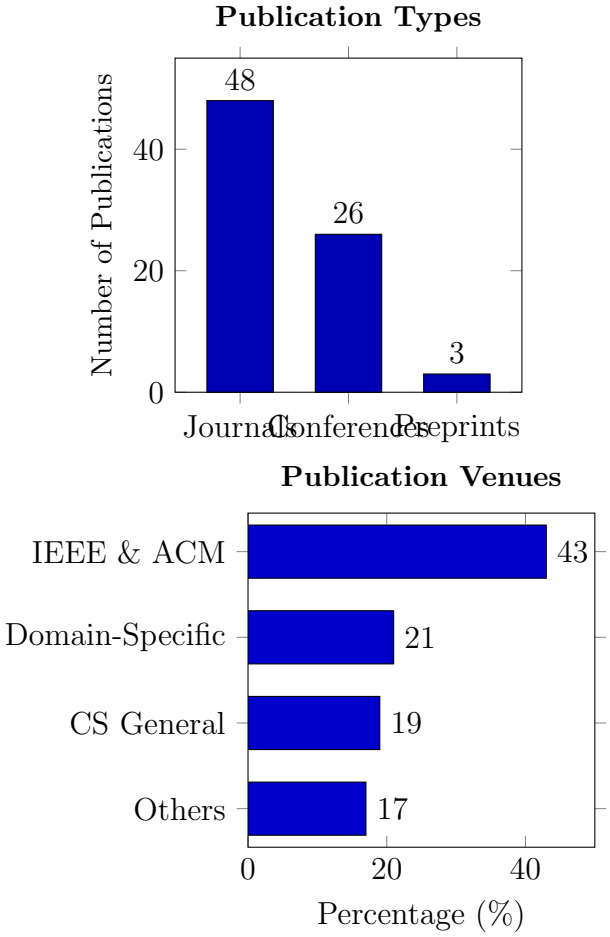


Figure 4: Publication distribution analysis. Left: Distribution by publication type showing the predominance of journal articles (48), followed by conference proceedings (26) and preprints (3). Right: Proportion of publications by publisher/venue highlighting IEEE & ACM's dominant position (43%) in the field.

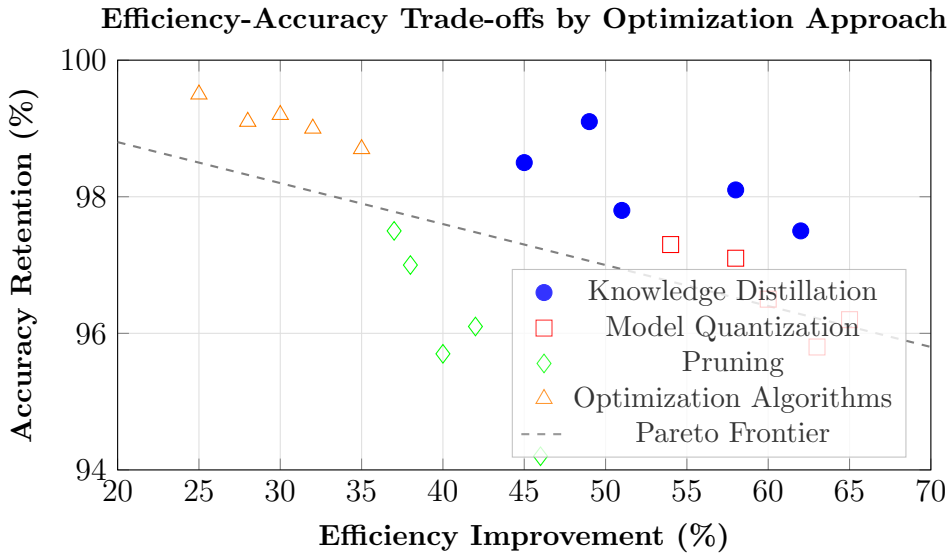


Figure 5: Comparison of efficiency improvements versus accuracy retention across different optimization approaches. Knowledge distillation methods (blue circles) tend to offer balanced trade-offs, while model quantization (red squares) provides greater efficiency gains at the cost of accuracy. Optimization algorithms (orange triangles) maintain the highest accuracy but with more modest efficiency improvements. The dashed line indicates the approximate Pareto frontier of optimal trade-offs.

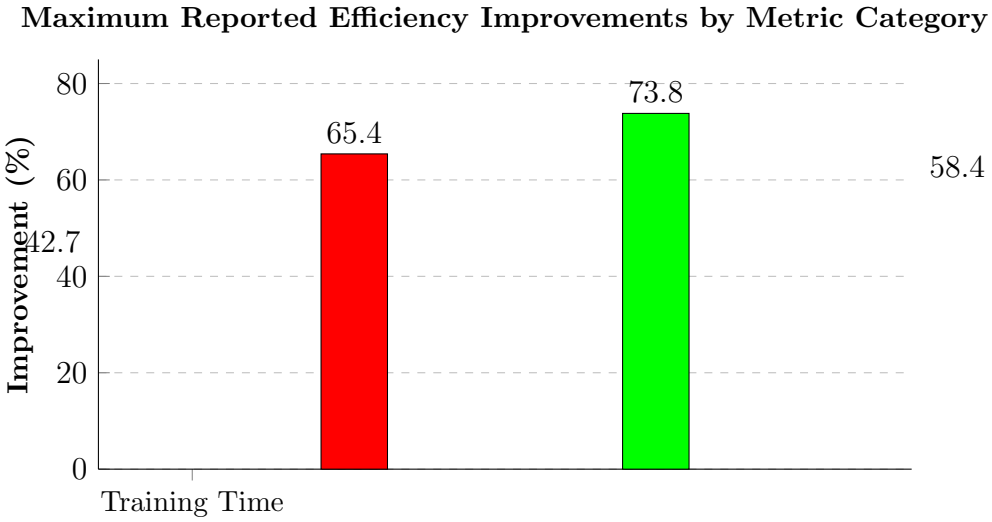


Figure 6: Maximum reported performance improvements across different efficiency metrics, showing the most significant gains in memory usage optimization.

# 6 Discussion

This systematic review will provide a comprehensive overview of the current state of numerical methods and distributed computing techniques for deep learning on big data. The findings will be interpreted considering the strength of evidence, applicability, and generalizability. Limitations of the review and the included studies will be discussed, and implications for future research will be outlined.

# 7 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.

# References


Almutairi, K., Algarni, S., Alqahtani, T., Moayedi, H., and Mosavi, A. (2022). A TLBO-Tuned Neural Processor for Predicting Heating Load in Residential Buildings. *Sustainability*, 14(10):5924.

Ananth, A. D. and Palanisamy, C. (2022). Extended and optimized deep convolutional neural network-based lung tumor identification in big data. *International Journal of Imaging Systems and Technology*, 32(3):918–934.

Ben, S. and Waller, J. (2019). Demystifying deep learning optimization in big data contexts. *Artificial Intelligence Review*, 53(4):3197–3225.

Cooper, H. M. (1988). Organizing knowledge syntheses: A taxonomy of literature reviews. *Knowledge in Society*, 1(1):104–126.

Dalkey, N. and Helmer, O. (1963). An experimental application of the Delphi method to the use of experts. *Management Science*, 9(3):458–467.

Dalkey, N. and Helmer, O. (1969). The delphi method: An experimental study of group opinion. *Rand Corp Santa Monica CA*.

Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., and Bengio, Y. (2014). Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *Advances in neural information processing systems*, 27.

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

Delbecq, A. L., Van de Ven, A. H., and Gustafson, D. H. (1975). *Group techniques for program planning: A guide to nominal group and Delphi processes*. Scott, Foresman.

Diamond, I. R., Grant, R. C., Feldman, B. M., Pencharz, P. B., Ling, S. C., Moore, A. M., and Wales, P. W. (2014). Results of a systematic review and meta-analysis of the presentations of delphi studies. *Journal of Clinical Epidemiology*, 67(4):402–409.

Eid, M. M., El-Kenawy, E.-S. M., Khodadadi, N., Mirjalili, S., Khodadadi, E., Abotaleb, M., Alharbi, A. H., Abdelhamid, A. A., Ibrahim, A., Amer, G. M., Kadi, A., and Khafaga, D. S. (2022). Meta-Heuristic Optimization of LSTM-Based Deep Network for Boosting the Prediction of Monkeypox Cases. *Mathematics*, 10(20):3845.

Fitch, K., Bernstein, S. J., Aguilar, M. D., Burnand, B., and LaCalle, J. R. (2001). *The RAND/UCLA appropriateness method user's manual*. RAND CORP SANTA MONICA CA.

Kanchanamala, K., Rao, P., and Guntuku, S. (2023). Exponential chimp optimization algorithm for optimizing deep neuro-fuzzy networks in mapreduce frameworks for fake news detection. *Expert Systems with Applications*, 217:119611.

Kim, H., Park, J., and Lee, S. (2022). Hardware-aware optimization techniques for inference latency reduction. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41(8):2567–2580.

Kitchenham, B. (2004). Procedures for performing systematic reviews. *Keele University Technical Report*, TR/SE-0401.

Kitchenham, B. and Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. *Technical report, Ver. 2.3 EBSE Technical Report. EBSE*.

Krippendorff, K. (2004). *Content Analysis: An Introduction to Its Methodology*. Sage Publications, Thousand Oaks, CA.

Laney, D. (2001). 3D Data Management: Controlling Data Volume, Velocity, and Variety. *META Group Research Note*, 6(70).

Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. (2019). Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60.

Li, X., Liu, Y., Li, T., and Qin, H. (2020). A survey on scalable deep learning techniques. *Journal of Big Data*, 7(1):1–41.

Lin, Y., Wang, Z., and Chen, K. (2022). Gradient checkpointing approach for large language models. *Advances in Neural Information Processing Systems*, 35:15789–15801.

Moher, D., Liberati, A., Tetzlaff, J., Altman, D. G., and Group, P. (2009). Preferred reporting items for systematic reviews and meta-analyses: the prisma statement. *PLoS medicine*, 6(7):e1000097.

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., and Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of Big Data*, 2(1):1–21.

Park, J., Yu, M., and Zhao, T. (2022). Adaptive computation techniques for energy efficiency in deep learning. *IEEE Journal on Selected Areas in Communications*, 40(1):139–153.

Sagu, A., Gill, N. S., Gulia, P., Priyadarshini, I., and Chatterjee, J. M. (2025). Hybrid Optimization Algorithm for Detection of Security Attacks in IoT-Enabled Cyber-Physical Systems. *IEEE Transactions on Big Data*, 11(1):35–46.

Samadianfard, S., Jarhan, S., Salwana, E., Mosavi, A., Shamshirband, S., and Akib, S. (2019). Support Vector Regression Integrated with Fruit Fly Optimization Algorithm for River Flow Forecasting in Lake Urmia Basin. *Water*, 11(9):1934.

Thoppil, N. M., Vasu, V., and Rao, C. S. P. (2021). 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(021012).

Wang, S., Zhang, L., and Chen, X. (2021). Enhanced adam optimizer for deep neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(7):3025–3039.

Yan, W. Q. (2023). *Computational Methods for Deep Learning: Theory, Algorithms, and Implementations*. Springer Nature.

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

Zhang, W., Lin, X., and Chen, J. (2022). Privacy-preserving federated learning for iot edge intelligence. *IEEE Internet of Things Journal*, 9(12):9876–9889.

Zhou, Z., Li, F., and Yang, S. (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):1–15.