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

Pouya Ataei

November 27, 2024

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 (?) and Kitchenham's methodology for SLRs in software engineering (?).

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 (?) 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)

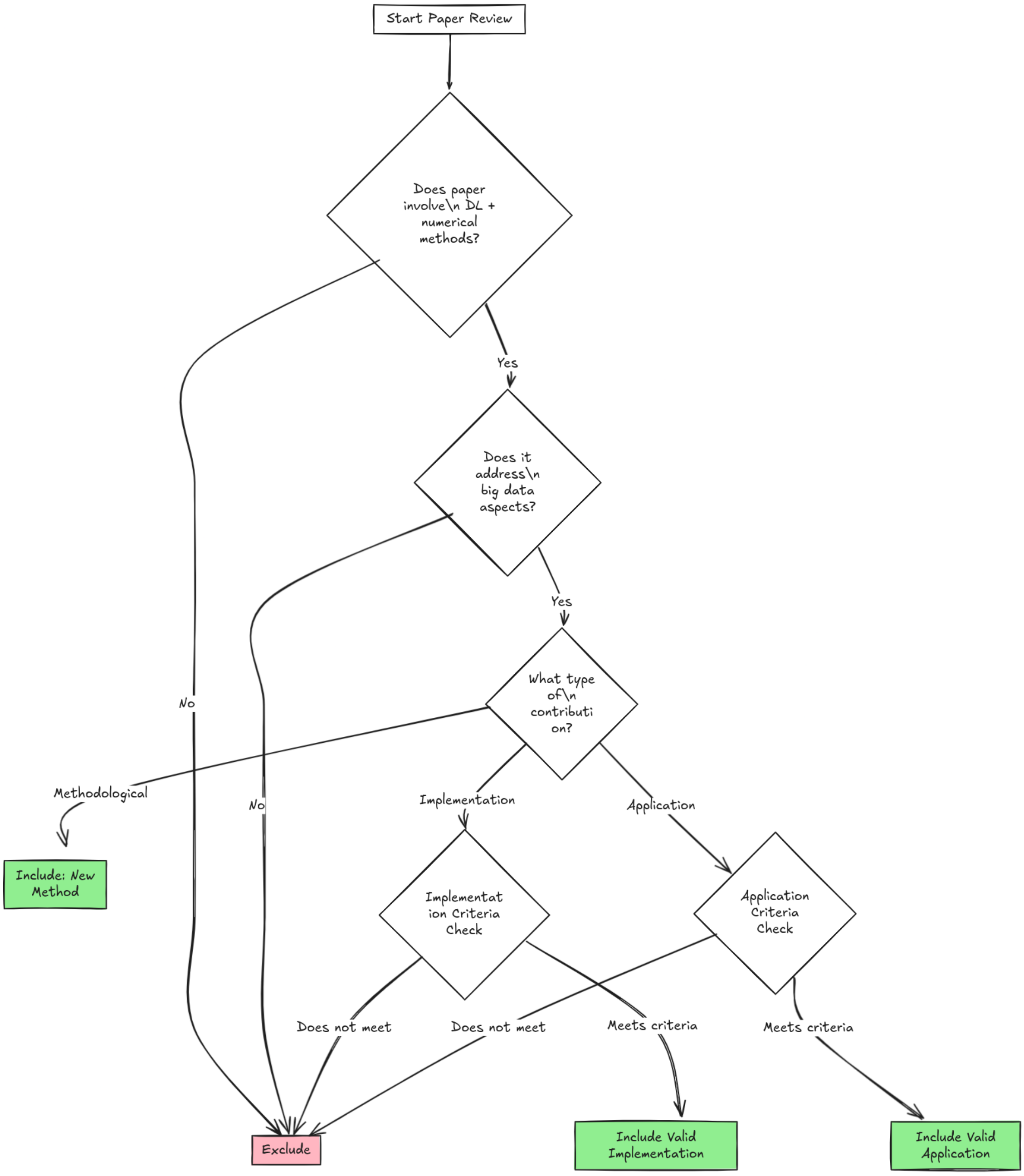

Figure 1: Paper Classification Decision Framework

- 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 **?**, 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 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.

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.

References

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

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*.

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.

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.

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

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.

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.

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.