Neural Network-based Framework for understanding Machine deep learning systems' open issues and future trends: A systematic literature review

Yaser Mohammed Al-Hamzi[a], Shamsul Bin Sahibuddin[b]

[a,b] Razak Faculty of Technology and Informatics,University of Technology Malaysia (UTM),54100 Kuala Lumpur – Malaysia
[a]mayaser1975@graduate.utm.my, [b] shamsul@utm.my

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 23 May 2021

Abstract: Nowadays, we live in the fourth industrial revolution era, where artificial intelligence, big data with machine learning engineering, and its subfield, the deep learning approach, uses a massive amount of data. This enormous amount of data must be analysed and computed efficiently. In this study, we present BiMDLs (Big machine deep learning systems), which contains state-of-the-art interfaces, frameworks, and libraries. To the best of our knowledge, significant limitations exist in several open aspects of BiMDLs and interfaces, and their ability to analyse, compute, and efficiently develop enormous data. Each of these aspects represents a framework issue that is interlinked in one way or another. This paper's goal is to summarize, organize and examine current BiMDLs and their technologies via a comprehensive review of recent research papers, to provide a synthesis and discuss observations of current open issues and future trends. Therefore, a systematic literature review (SLR) was developed, and 284 solid studies were conducted, analysed and discussed. Furthermore, we highlight several significant challenges and missing requirements of existing big machine deep learning engines and future extension directions. We believe that this SLR could benefit big machine deep learning researchers, developers, and specialists for further improvement; especially in parallel computing environments.

Keywords: Neural Network, BiMDLs, Machine Learning, Deep Learning, Open issues, Big Data, Systematic Review

1. Introduction

In today's digital world, computer science pays considerable attention to new generations of artificial intelligence; which is rapidly expanding, and has again become an attractive research topic [1]. Big data and software machine deep learning approaches, where ‘big data’ is a large set of distributed source data that is challenging to handle and evaluate using conventional methods[2]; [3]; [4], [5]Nowadays, data is power that leads to an organization becoming successful and big data analytics force industries to diagnose, forecast, and understand potential growth, leading them to achieve business value[6].

Big machine deep learning systems (BiMDLs), and their related technologies, are relatively new and are still evolving. Only a few works have attempted to tackle the shortcomings and complexities of artificial architectures; especially machine and deep learning techniques [1],[2],[3],[4]. In this systematic review, we presented a BiMDLs’ framework that contains state-of-the-art interfaces, frameworks, and libraries, such as Facebook’s Torch/PyTorch and Caffe2[4],[5], [6];**[7]**,[8],[9], University of Montreal’s Theano, Google’s TensorFlow, [10], Apache’s MxNet, and Microsoft’s CNTK[8],[9],[10],[11] . The size of tensor in deep learning is huge, possibly reaching more than 200 million dimensions with 8 billion points; or what has become known as “the curse of dimensions” [7], [12]–[14]; [8]; [9]; [10]. To solve this high-dimensional problem, developers, researchers, programmers, and even software companies, designed a myriad of frameworks (known as big machine deep learning systems), to handle the complex matrices and mathematical operations involved[11]; [15]. For instance, Facebook’s Torch/PyTorch and Caffe2 [16], [17]; [18], University of Montreal’s Theano, Google’s TensorFlow[19]–[22], Apache’s MxNet, and Microsoft’s programming libraries with fixed user interfaces[23]. However, most machine deep learning frameworks are converging towards a common pipeline design; similar in terms of purpose, goal, and mission [22], [24]–[27].

Researchers face severe challenges in terms of incompatibility among these software systems, [15];[16]; [9]; [8]; [17]. We present and discuss several other relevant open issues, such as the difficulty of code conversion[18], [17], the lack of benchmarks [1],[19], and the difficulty of selecting a proper framework from all big machine deep learning frameworks [15], [2]. The literature review of this study reveals that these issues affect computing efficiency and effectiveness in parallel, in terms of increased computing and development time, difficulty in organizing computing tasks, increase in computing process costs, and decreased computing accuracy due to goal mismatches, which makes the process of computing, training and performance extremely complicated[15],[18] .

Machine deep learning frameworks, based on the above-mentioned open issues, are still in need of appropriate solutions. Empirical research indicates that achieving a state of effective combination is a crucial success factor for BiMDLs’ projects [2].Therefore, after analysing the literature on big machine deep learning systems, we demonstrate that it is indeed an active study area, and that real challenges exist that need more intense in-depth study to analyse and identify potential solutions.

A combination of various machine learning models can boost the parallel computing process's accuracy; this could be achieved through a unified model that will undoubtedly lead to improved performance of the industry's machine deep learning techniques. However, to achieve the best accuracy, decreased time-consumption and reduced computing costs, and a combination of two or more of these methods is required[22]; [28]; [25]. We believe that this paper proposes a promising technology model to enhance BiMDLs' frameworks and their related libraries' compatibility.

2. Related Work

Many reviews, studies and various surveys have been conducted in the last six years on various topics of big machine deep learning systems (BiMDLs). Moreover, the machine learning approach has the potential to improve many business functions and meet a wide range of organizational needs[29]. For instance, BiMDLs capabilities can be leverage to recommend products to users based on previous purchases, provide image recognition for video monitoring, identify spam emails, and predict courses of action, paths, or diseases, amongst other things. However, Except for big high-tech firms such as Microsoft and Google, most organizations' development of ML capabilities is still primarily a research activity or a standalone project. Furthermore, there is a scarcity of existing guidance to help organizations develop these capabilities. The fragility of ML components and their algorithms limits their integration into applications. They are vulnerable to changes in data, which may cause their predictions to shift over time. Mismatches between system components also hamper them.

This paper's fundamental goal is to conduct a thorough investigation of state-of-the-art big machine deep learning systems BiMDLs interfaces and their libraries. It includes an in-depth discussion of current BiMDLs technologies in terms of features offered, categorization, and classification. Additionally, many important open issues and further research opportunities will be presented for the next step of big Machine deep learning technologies development. While producing this paper, no other systematic study has been found on BiMDLs and their related technologies covering most of the existing open issues to the best of our knowledge. On the other hand, most of the published reviews in big machine deep learning systems did not address the lack of compatibility, the lack of code conversion, the lack of benchmark, and the difficulty of choosing among the big machine deep learning systems BiMDLs.

However, more research is needed to establish the unique advantages obtained by combining these technologies and understanding how AI can be further improved with the increasing availability of Big Data with its volume, variety, and velocity [1], [30]. Therefore, there is a necessity to fully understand the synergy of AI systems and Big Data methods and its implications for AI research and practice. Furthermore, the literature overlooks the significant need to discuss or make mention of the importance of big data, machine learning, and deep learning in terms of their ecosystems and frameworks and the scarcity of AI, ML, DL, and BD resources despite the urgent need for more of research to match with the incredible acceleration and development of the fourth industrial revolutions. On the other hand, a lot of reviews and surveys have been done in the last years on various topics of big data.

3. Literature review and A Systematic Literature Review SLR

A systematic literature review SLR is a sequential methodological step that guides researchers by identifying the research objective and preparing how papers will be retrieved and reported. It simply aims to gather all empirical data that satisfies pre-specified eligibility requirements to address a specific research question. It employs explicit, Specific systematic methods chosen to reduce bias, resulting in more accurate results from which conclusions can be formulated and decisions can be drawn. Moreover, Systematic literature reviews provide comprehensive datasets, making them a primary resource when referring to evidence in the studied research area. This research followed a set of measures to produce a systematic, transparent, and repeatable result. In other words, it is a form of secondary study that uses a well-defined methodology to identify, analyze, and impartial and (to some extent) repeatable interpretation of all relevant facts related to specific research questions [31].

This SLR represents a significant contribution to the researchers through providing opportunities for further improvement on BiMDLs big Machine deep learning systems environments. Moreover, it guides the researchers and developers for successful BiMDLs by evaluating the factors and their related dimensions that influence the parallel computing process. The contributions of this SLR in response to the research questions raised are described in detail. (RQ) are as follows: For RQ1: What are the most common characteristics, similarities, differences, attributes, advantages, and disadvantages among the big machine deep learning systems (BiMDLs) in terms of their goal, tasks, and function?To answer this question, the SLR provides deep knowledge and informative review about using the existing Big machine deep learning systems in several different levels of their performance, such as big data analysis, big data storage, big data process, and parallel computing. We also provide some of the multiple comparisons that we found served the purpose of the research.

RQ2: What are the main open issues and challenges of the current big machine deep learning systems BiMDLs? Therefore, the answer to this question will identify the big machine deep learning factors and dimensions and their impacts and identify the current BiMDLs challenges. On the other hand, the factors that affect the BiMDLs, in which the answer for this question will help the researchers, developers, and organizations to process the precision, accuracy, efficiency, and quality needed with less time-consuming and with low cost.

RQ3: What critical factors and dimensions affect the existing big machine deep learning systems BiMDLs? This research has focused on several open issues and limitations: After analysing the systematic literature review of the existing big machine deep learning systems BiMDLs and their related frameworks, we found that this is still an open research domain, and some issues need further exploration, discussion to find the appropriate solution.

The significant issues associated with this domain, such as the lack of *c*ompatibility among machine Frameworks, the difficulty of code generation and conversion, the lack of benchmarks within big machine deep learning frameworks, and the difficulty of selecting the proper Framework and library.The existent SLR provides a comprehensive review on the current BiMDLs open Challenges and limitations, especially in Parallel computing aspect.

RQ4: What are the empirical approaches that overcome the current challenges of BiMDls?The answer focuses on and addresses the best strategies for a more straightforward solution by proposing a Unified platform and design an appropriate prototype in our future work, capable of combining the big machine deep learning systems features in one framework to enhance the BiMDLs techniques Compatibility and overcome their critical challenges.

Planning the review

Identify the need for this review

Before undertaking this SLR, we needed to identify and review the existing SLR of the phenomenon of interest to clarify whether our review has already been done and provide a rationale for conducting an updated review.

The checklist suggested by previous studies when reviewing the SLR:

- What were the main systematic review’s objectives?
- What source wereleveraged to collect the primary studies? Were they imposing significant limitations?
- What are the inclusion and exclusion criteria, and how are they applied?
- How were data from primary studies extracted?
- What methods were used to investigate the differences between studies? How was the information combined? Does the evidence lead to the conclusions?

Determine research questions

In this stage, the research questions are elaborated clearly and precisely, and the search protocol is defined. The questions identification stage includes the following methodological elements:

*Search process:*Aims to Identify the primary studies that should be address RQs.

*Data extraction:*Aims to Extract the information required to respond to the RQs.

Data analysis: Aims to synthesize the information in such a way that the RQs can be answered.

*Population:*The essential elements here are: Artificial Intelligence (AI), Machine Learning (ML), Deep Learning (DL), Data Science, and Big Data (BD).

Intervention: Planning models, tools, Techniques, Libraries, factors, and dimensions for BiMDLs methods and their current open issues and challenges.

Comparison: literature provides a various comparison among BiMDLs Frameworks, methods, models, interfaces, libraries, algorithms, and systems in different levels.

Outcomes: Frameworks, Porotypes, Tools, Techniques, Interfaces, Factors, Dimensions, Planning models, Comparative results, and a comprehensive obvious and applicable strategy for BiMDLs big machine deep learning system evaluation and its related open issues and challenges.

Context: Any high-quality previous work, findings relevant to the BiMDLs and the current open issues.

*Our Approach is to:*Separate the question into its various facts and components.

The research questionsRQ are:

RQ1: What are the most common characteristics, similarities, differences,attributes, advantages and disadvantages among the big machine deep learning systems (BiMDLs) in terms of their goal, tasks and function?

RQ2: What are the main open issues and challenges of the current big machine deep learning systems BiMDLs?

RQ3: What critical factors and dimensions affect the existing big machine deep learning systems BiMDLs?

RQ4: What are the empirical approaches that overcome the current challenges of BiMDLs and reduce high-cost, time-consuming parallel computing process?

Since this systematic review includes several comparisons of various methods, systems, frameworks and Approaches, research objectives presented as follows:

The research Objectives RO are:

RO 1: To identify, categorize, classify the big machine deep learning systems BiMDLs and their components to determine the advantages and disadvantages of the included interfaces and libraries. that occurred due to their differences in parallel computing goals.

RO 2: To identify, investigate and filter the synthesis research evidence of the Existing big machine deep learning systems open issues and related challenges.

RO 3: To Identify the critical factors and dimensions that affect the existing big machine deep learning systems BiMDLs?

RO 4: To reduce the high-cost, time-consuming parallel computing process by increasing the compatibility, speed up code generating & conversion, and enhance accuracy and training quality among big machine deep learning systems by Proposing a practical, low-cost Model.

Develop a review protocol

A pre-defined protocol is necessary to reduce the possibility of researcher bias, such as selection or analysis of studies that researcher expectations may drive. Therefore, regarding development of the revision, defined protocol is applied, and the primary articles are obtained according to the established criteria. The protocol includes the following:

Background: The rationale.

Research questions: Questions that the review intended to answer.

Search strategy: Including search terms and resources/databases to be searched, such as digital libraries, specific journals, and conference proceedings.

*Data extraction strategy:*Aims to define how each primary study's information will be obtained.

Synthesis strategy: Meta-analysis / Quantitative Synthesis, define synthesis strategy and the techniques to be used.

The following questions about the factors and dimensions are asked to answer the RQs that influence the success of the big machine deep learning systems BiMDLs:Q1: What is success for BiMDLs?Q2: What factors and dimensions affectBiMDLs?Q3: What are the classifications, and how are factors and dimensions classified?

Selection strategy

As shown in Table 1, best represented as a flow diagram with inclusion and exclusion criteria based on the research question and quality assessment.

Inclusion criteria: The following inclusion criteria were contributed to include previous work:

TABLE 1: Inclusion Criteria

Inclusion Criteria	**Reason for Exclusion**
Research focus	Research papers that identify the critical success factors and desirable outcomes,including their critical success dimensions those that categorize the factors and show the

	development phases.
Quantitative empirical studies	These publications are included because they provide existing empirical evidence, which is the primary focus of this review.
Impact factor	Only articles from journals with considerable impact factor such as SJR are taken into account.
Language	English language studies isonly considered.
Theories	We selected only the related, critical and inclusive theories, then we associated each factor and dimension to its theory
Date of publication	Only from 2015 to 2021, as we focused only on the recent open issues and challenges.
Participants	The essential elements here are: Artificial Intelligence (AI), Machine Learning (ML), Deep Learning (DL), Data Science, and Big Data (BD). experts, and specialists
Literatures	The review was limited toarticles in high-quality peer-reviewed, high-indexed journals and high quality international scientific conferences
For any duplicate studies,	Only the latest version we used.
Evaluation	All studies that have empirical evaluation was involved.

Exclusion criteria: Table 2 illustrates theinclusion criteria that were used to exclude the literature review are:

TABLE 2: Exclusion Criteria

Inclusion Criteria	**Reason for Exclusion**
Publication type	We excluded books, book chapters, dissertations, low-quality conference, short articles and non-indexed journal
Unit of Analysis	Exclude studies that do not consider Artificial intelligence, Machine learning, Deep learning, Big data, parallel computing and technology-based
Research focus	Research papers that fail to include research methodology or numerical test results (descriptive statistics), benchmarking, visible tangible results and analysis or discussion
Language	We excluded all languages except English language.
Theory	Excluded the unrelated and inconclusive theories
Date of publication	We excluded studies before 2015 as our discussed issues
For any duplicate studies,	We excluded all duplicate studies except last version.

4. Methodology

Conducting the review

Following the planning and establishment of a predefined and well-defined plan, it is time to execute the literature by following the procedures and tasks outlined in the plan as follows:

Search strategy

It is commonly assumed that the more explicit and meticulous the search strategy, the more likely it is that a SLR will find all of the relevant papers. The majority of high-quality primary studies can be found in systematic reviewsof this research by seeking standard electronic databases. Furthermore, informal methods such as web surfing, "asking around," and being alert to serendipitous discovery can significantly boost the search efforts' yield and efficiency. As well, Methods that "snowball," such as seeking references and electronic citation tracking, effectively locating high-quality sources in remote locations. The search strategy includes the data source and the search string.

Identify search strings and Search resources

Before approaching to online searching process, we should determine all possible terms that were extracted from the research questions to find a relevant primary study. The terms combine together to form the statement which known as search string. As a result, we first identified the main idea of the big machine deep learning software and reviewed the keywords; Then, using the abstract and title of some identified primary studies, we identified alternative keywords and terms. Finally, we used the or/and Boolean operators to form the search string. As a result, we discovered many terms divided into fifteen categories, each of which contains alternative terms, and we listed synonyms, abbreviations, and alternative spellings as follows.:

- Artificial Intelligence, Machine Learning, Machine Learning Sycle, Algorithms, Deep Learning, The Fourth Revolution Industry, Software Engineering.
- Systems, Tool, Techniques, Technologies, Architectures, Package, Software, Toolkits, State of The Art, Ecosystems.
- Training, Modelling, Computing, Parallel Computing, Non-Parallel Computing.
- Tensor, TensorFlow, Torch, PyTorch Caffe, Caffe2, CNTK, Theano, Keras)
- Big Data, Big Data Characteristics, Big Data Analytics, Big Data Tools, Data Science, Data Mining, Dataset
- Framework, Frameworks, System, Model, Prototype, Method, Frame, Design, Typical
- Comparison, Comparative, Compare, Differentiation, Differences, Different, Similarities, Likeness
- Enhance, Enhancing, Improve, Improving, Develop, Developing, Optimize, Optimizing
- Advantages, Disadvantages, Negatives, Positives, Lack, Lacking, Shortage, Limitation, Pros, Cons
- Factors, Variable, Determinants, Components, Facts, Reasons, Categories, Aspects, Agent, Representative
- Dimensions, Subfactors, Measure, Measurement, Extent, Pointers
- Review, Survey, SLR, Systematic Literature Review, Literature Review, Comparative Study, Case Study, Challenges, Open Issues, Future Trends, Future Work
- Benchmark, Benchmarking, Benchmarks, Datasets, Evaluation, Evaluating
- Coding, Code, Code Generation, Code Conversion
- Compatibility, Interoperability, Matching, Match, Convenient, Appropriate, Suited, Consistent.

Use Boolean ANDs and ORs (when a particular keyword produces too many results): We used the following search string in the titles, abstract and keywords as presented in Figure 1.

(Artificial Intelligence OR Machine Learning OR Machine Learning Sycle OR Algorithms OR Deep Learning OR The Fourth Revolution Industry OR Software Engineering) **AND** (Systems, Tools OR Techniques OR Technologies OR Architectures OR Package OR Software OR Toolkits OR State Of The Art OR Ecosystems) **AND** (Enhance OR Enhancing OR Improve OR Improving OR Develop OR Developing OR Optimize OR Optimizing)**AND** (Training OR Modelling OR Computing OR Parallel Computing OR Non-Parallel Computing) **AND** (Tensor OR TensorFlow OR Torch OR PyTorch Caffe OR Caffe2 OR CNTK OR Theano OR Keras)**AND** (Big Data OR Big Data Characteristics OR Big Data Analytics OR Big Data Tools OR Data Science OR Data Mining OR Dataset) **AND** (Framework OR Frameworks OR System OR Model OR Prototype OR Method OR Frame OR Design OR Typical) **AND** (Comparison OR Comparative OR Compare OR Differentiation OR Differences OR Different OR Similarities OR Likeness) **AND** (Advantages OR Disadvantages OR Negatives OR Positives OR Lack OR Lacking OR Shortage OR Limitation OR Pros OR Cons) **AND** (Factors OR Variable OR Determinants OR Components OR Facts OR Reasons OR Categories OR Aspects OR Agent OR Representative) **AND** (Dimensions OR Subfactors OR Measure OR Measurement OR Extent OR Pointers) **AND** (Review, Survey OR SLR OR Systematic Literature Review OR Literature Review OR Comparative Study OR Case Study OR Challenges OR Open Issues OR Future Trends OR Future Work) **AND** (Benchmark OR Benchmarking OR Benchmarks OR Datasets OR Evaluation OR Evaluating) **AND** (Coding OR Code OR Code Generation OR Code Conversion) **AND** (Compatibility OR Interoperability OR Matching OR Match OR Convenient OR Appropriate OR Suited OR Consistent).

Figure 1: Search string and Keywords

We manually searched for other sources of evidence such as through:

Forward search (Snowballing technique): studies that have cited initially identified studies according to [32].

Backward search: from reference lists of initially identified studies, based on (Ali et al., 2018). For articles that were unavailable but pivotal for this study, it was considered and required**.**

The review results: The number of results found for each keyword was recorded, including the sources. So, we present the findings of the search and then analysis of the studies that were chosen. The Analysis section will go over this analysis.

Literature selection: The review was limited to articles in peer-reviewed, high indexed/quality journals and high quality international scientific conferences, leaving out books, book chapters, and low-quality papers. We decided on our approach regarding criteria selection during the protocol definition to minimize the likelihood of bias, although they may be refined during the search process.

The search sources are: ISI WOS, IEEE Xplore, Science Direct, Scopus, ACM Digital Library, Springer Link, Digital Library, and others (Emerald, Wiley& SPIE). The search period begins in the year 2015 until 2021.

*Theories:*In this study we have associated all of the identified factors and their related dimensions to what theory it suits them. The selected theories are listed in table 3.

TABLE 3: Summary of used theories in SLR

No	**Theory**
1	Computational or (ML) Machine learning theory
2	Complexity theory
3	Structured process modelling theory
4	Computational complexity theory
5	Stakeholder theory
6	Delone and McLean IS success model
7	Coding theory
8	Transaction cost economics (TCE)
9	Information processing Theory
10	Transactive memory theory
11	Theory of computation
12	programming language Theory
13	Theory of technology Dominance (TTD)

Assessing the quality of studies

According to the guidelines in [33], we conduct the quality assessment criteria shown in table 4. To evaluate the papers and select high-quality studies. We have created **nine** questions answered by 'Yes' **Y**, 'Partly' P and 'No' **N** answer to address the quality assessment and are presented in Table 4. The scoring points are Y = 1, P = **0.5** and **N = 0**. Furthermore, each primary study should get from **0** to **13** score points.

TABLE 4: Quality assessment form

ID	Question	Score
1	Do the Studies provide models, Tools, Frameworks and libraries related with Big machine deep learning systems BiMDLs?	Y|P |N
2	Do the Studies provide technique to select metrics to evaluate BiMDLs?	Y|P |N
3	Do the Studies provide factors that affect the BiMDLs?	Y|P |N
4	Do the studies provide dimensions influence the BiMDLs?	Y|P |N
5	Do the Studies provide mitigation strategies to overcome the challenges of BiMDLs	Y|P |N
6	Do the studies provide a reasonable technical comparison?	Y|P |N
7	Do the results of the studies is generalizable and applicable?	Y|P |N

8	Does the data extracted adequately described?	Y|P |N
9	Are the inclusion and exclusion criteria of the studies adequately described?	Y|P |N

Appraisal of quality ensures that only the most appropriate, trustworthy, and relevant studies are used to develop the review's conclusions.

Data extraction: To gather all of the primary study information required to answer the review questions and meet criteria for determining study quality. Therefore, data extraction is form need to be designed.Electronic forms are useful and can facilitate subsequent analysis. So, to ensure data extraction consistency, two techniques can be employed:

Supervisor: extracts data from random sample of the primary studies and cross-checked with students' results.

Researcher: can perform second extraction from a random selection of studies.After selecting the SLR primary data studies, we designed the data extraction form presented in Table 5 to extract the primary search process correctly. The studies papers' information has been identified using the form of data extraction and should contain fields corresponding to each research question. Then, the quality assessment questions, which are listed in Table 5, are evaluated for each primary study. The primary studies that are selected answer some or all of the research questions.

TABLE 5: Form of data extraction

Search interest	**Extracted Data**
General information	Paper title, Paper type, author(s) name(s), publication Year, publication index
RQ1	Systems, Frameworks, Methods, Tools, Models, practices, characteristics, Attributes, and applications, Similarities, Advantages, Disadvantages
RQ2	Limitations, Open issues, Challenges, Software, platforms, mechanisms, Techniques, Design, Lacking, Precision, accuracy, Quality, Parallel computing, AI, Big data, ML, DL, Data science
RQ3	Factors, Dimensions, Techniques, Benchmarks, Coding, Complexity, Differences, Features,
RQ4	Solutions, Mitigate strategy, Evaluations, Metrics, Models, Performance, efficiency

The data that is related to the all four research questions has been extracted to get more knowledge about it. This SLR has performed the data extraction for the 284 primary studies. All the extracted data and information has been analysed and discussed until the result achieve the objectives of this SLR study.

Reporting the review: Format report

Eventually, the results of the literature are reporter and conclusion are driven from the identified data and material. This section will detail out the SLR process by following the guidelines provided byKitchenham in [31] by which includes the research questions that achieve the objectives of the SLR.

Structure and contents of systematic review report

Title: Based on the question being asked and indicate that the study is a systematic review

Abstract: Include context, objectives, methods, results, conclusions.

*Background:*includesJustification for the need for review, synopsis of previous reviews, and review question specification.

*Review Methods:*This contains Search strategy,data sources, study selection, quality assessment, data extraction and data synthesis.

The included and Excluded Studies: The sub-section includes the inclusion and the exclusion criteria, set of excluded studies, and justification for exclusion.

References and Appendices: To list included and excluded studies or to the raw data from the included studies

Results: Description of primary studies, summaries or details of analysis.

*Discussion:*The strengths and weaknesses of the evidence are included in the review compared to other reviews, taking into account. any differences in results and meaning of findings.

Conclusions: Practical implications and unanswered questions and implications for future research.

Multi-Criteria Analysis and AHP Technique

MCA Approach is a complementary method to cost-benefit analysis (CBA). It consists of a two-phases decision-making process. In the first phase we identify a set of our research objectives. It then attempts to identify trade-offs between those goals, different policies, or different ways of achieving a given procedure.The second stage attempts to determine the "best" policy by assigning weights (scores) to the various objectives. The Analytic Hierarchy Process (AHP) offers a comprehensive analysis and rational framework for structuring a decision problem. In the present research, an empirical study was conducted to determine the relative importance of the factors affecting the Big machine deep learning systems.

The result's input was used for pair-wise comparison of the factors influencing the BiMDLs. In AHP, the comparison of alternatives is based on the input of an expert team and the SLR findings. Finally, a comprehensive, precise analysis must be conducted to assess the impact of factors and their related dimensions on our model. The steps to be followed while implementing the AHP technique are described below:

Step1:Make a hierarchical decision structure by dividing the entire BiMDLs problem into parameters or criteria.

Step2:Create a series of judgments based on pair-wise comparisons to establish priorities among the hierarchy's parameters or criteria. Preferences for parameters are rated on a scale in this step.

Step3:In this step we synthesize these assessments to create a set of overall priorities for the hierarchy. Weighted criteria scores are calculated in this step, yielding a relative ranking of parameters or criteria.

Step4:To check the consistency of the judgments, we basically compare the quantitative and the qualitative results using decisions that are well-informed to derive weights and priorities.

Step5:Choosing the best alternative based on Using the available sample data, calculate the total score for each potential alternative.

Finding

In this section, we have analyzed the data that was extracted. We started withSearch process resultsby presenting the overall finding of these SLR papers presented in Figure 2, and then the answer for each RQ is provided based on the collection data analysis, as shown in Table 7**.** Furthermore, Table 6 provides a list of the most selected High-Indexed Journals whose papers were cited and their Number of cited articles.

The results show that the top twenty cited journals in this research (shown in the upper coloured part in table6) were as follows: Journal of Business Research, Journal of systems architecture, IEEE Access journal, International Journal of Information Management, Procedia Computer Science, International Journal of Electrical and Computer Engineering, Education and Information Technologies, Information Fusion, Neurocomputing, The Journal of Systems and Software, Applied soft computing, IEEE Computer Society, Artificial Intelligence Review, Machine Learning Journal, Artificial Intelligence in Medicine, Journal of Big Data, Knowledge-Based Systems,IEEE MICRO, IEEE Transactions on Parallel and Distributed Systems, Computers & Chemical Engineering.

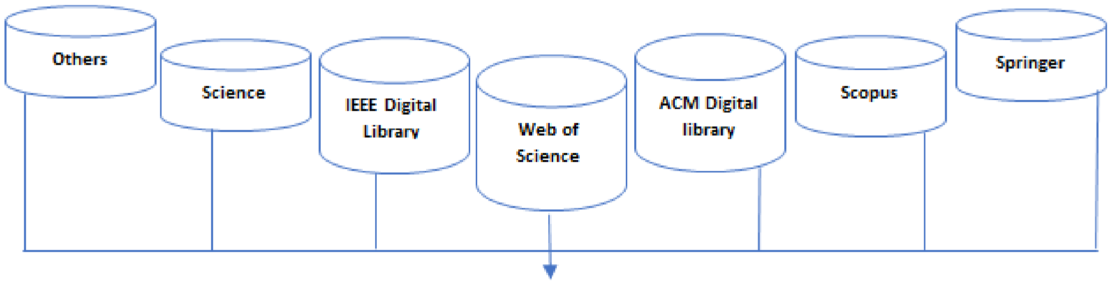
TABLE 6: List of High-Indexed Journals and Their Number of cited articles

No	Journal name	Number of cited articles	Indexed by
	Journal of Business Research	5 Articles	SCOPUS/WOS/SCIENCEDIRECT
	Journal of systems architecture	4 Articles	SCOPUS/WOS/SCIENCEDIRECT
	IEEE Access journal	4 Articles	SCIE/WOS
	International Journal of Information Management	3 Articles	SCIENCE DIRECT
	Procedia Computer Science	3 Articles	SCIENCE DIRECT

	International Journal of Electrical and Computer Engineering	3 Articles	SCOPUS
	Education and Information Technologies	3 Articles	SPRINGER and SSCI
	Information Fusion	3 Articles	SCIE/WOS
	Neurocomputing	2 Articles	SCOPUS/SCIENCEDIRECT/WOS
	The Journal of Systems and Software	2 Articles	ISI, SCOPUS, SCIENCE DIRECT
	Applied soft computing	2 Articles	SCIE/WOS
	IEEE Computer Society	2 Articles	WOS
	Artificial Intelligence Review	2 Articles	SCIE/WOS
	Machine Learning Journal	2 Articles	SPRINGER
	Artificial Intelligence in Medicine	2 Articles	SCIENCE DIRECT
	Journal of Big Data	2 Articles	WOS/ESCI
	Knowledge-Based Systems	2 Articles	SCIENCE DIRECT
	IEEE MICRO	2 Articles	IEEE, SCIE/WOS
	IEEE Transactions on Parallel and Distributed Systems	2 Articles	IEEE Xplore, WOS/SCIE
	Computers & Chemical Engineering	2 Articles	ISI/SCIE/WOS
	Concurrency and computation-practice & experience journal.	2 Articles	SCIE/WOS
	Journal of machine learning & knowledge extraction	1 Article	SCOPUS
	Engineering Applications of Artificial Intelligence	1 Article	SCIENCE DIRECT
	IEEE Computer Society Technical Committee on Data Engineering	1 Article	IEEE Digital library
	Urban Water Journal	1 Article	WOS/SCIE
	IEEE Transactions on Knowledge and Data Engineering	1 Article	WOS/SCIE
	Parallel Computing	1 Article	SCIENCE DIRECT
	Knowledge and Information Systems	1 Article	WOS/SCIE
	MRS Communications	1 Article	WOS/SCIE
	Information and Software Technology	1 Article	SCIENCE DIRECT
	Technological Forecasting and Social Change	1 Article	SCIENCE DIRECT
	European Scientific Journal	1 Article	WOS/ESCI
	Government Information Quarterly	1 Article	SCIENCE DIRECT
	Technological Forecasting & Social Change	1 Article	SCIENCE DIRECT
	Journal of Computer Communications	1 Article	SCIE/WOS
	Journal of Archives of Computational Methods in Engineering	1 Article	SPRINGER/SCIE
	Journal of Metals and Materials International	1 Article	IEEE Xplore
	Economic Analysis and Policy	1 Article	SCIENCE DIRECT
	Engineering	1 Article	SCIENCE DIRECT
	Journal of EEE Transactions on Visualization and Computer Graphics	1 Article	WOS/SCIE
	Computer Vision and Image Understanding	1 Article	SCIENCE DIRECT
	Soft Computing Journal	1 Article	SPRINGER/SCIE
	International Journal of Computer Vision	1 Article	SCIE/WOS
	Drug Discovery Today	1 Article	SCOPUS
	REMOTE SENSING	1 Article	ISI/SCIE/WOS
	JOURNAL OF DIGITAL IMAGING	1 Article	ISI/SCIE/WOS
	International Journal of Innovative Technology and Exploring Engineering	1 Article	IEEE Digital library
	Decision Support Systems	1 Article	SCIENCE DIRECT
	Journal of Manufacturing Systems	1 Article	SCIENCE DIRECT
	Journal of Computer Science and Technology	1 Article	WOS/SCIE
	Journal of Computers in Human Behaviour	1 Article	SCIENCE DIRECT
	IEEE Communications Surveys and Tutorials	1 Article	WOS/SCIE
	PLoS ONE	1 Article	WOS/SCIE

	Information Sciences	1 Article	WOS/SCIE
	Computer Journal	1 Article	IEEE Digital library/WOS/SCIE
	Futures	1 Article	SCIENCE DIRECT
	International Journal of Management Education	1 Article	SCIENCE DIRECT
	Materials Today: Proceedings	1 Article	SCIENCE DIRECT
	Computers in Biology and Medicine	1 Article	SCIENCE DIRECT
	Journal of WSEAS	1 Article	SCOPUS
	Journal of Development Policy	1 Article	Wiley online Library
	Applied Science Journal	1 Article	SCOPUS
	Ain Shams Engineering Journal	1 Article	SCIENCE DIRECT
	International Journal of Information System Modeling and Design	1 Article	WOS/ESCI
	Multimedia Tools and Applications	1 Article	SCIENCE DIRECT
	Proceedings of the IEEE	1 Article	WOS/SCIE
	Annals of internal medicine	1 Article	WOS/SCIE
	Business Horizons	1 Article	SCIENCE DIRECT
	Journal of Neuroscience Methods	1 Article	SCIENCE DIRECT
	Computers and Operations Research	1 Article	SCIENCE DIRECT
	Expert Systems with Applications	1 Article	SCIENCE DIRECT
	Frontiers in genetics	1 Article	SCIE/WOS
	Computer Methods and Programs in Biomedicine	1 Article	SCIENCE DIRECT
	Computational and Structural Biotechnology	1 Article	WOS/SCIE
	Journal of Imaging	1 Article	ESCI/WOS
	IEEE Computer Society Technical Committee on Data Engineering	1 Article	IEEE Digital library
	Journal of King Saud University - Computer and Information	1 Article	WOS/SCIE
	IEEE Internet Computing	1 Article	IEEE Digital computing/ SCIE
	Journal of Product Innovation Management	1 Article	WOS/SCIE
	IEEE Transactions on Services Computing	1 Article	SCIE/WOS
	International Journal of Engineering Research & Technology	1 Article	SCIE/WOS
	Neurosurgery	1 Article	SCIE/WOS
	IIOAB Journal	1 Article	ESCI/WOS
	IEEE Transactions on Multi-Scale Computing Systems	1 Article	IEEE Digital library

Search process resultsby presenting the overall finding of these SLR papers presented in Figure 2



Total of References Using Search Strings for Potentially Eligible Studies	
Google Scholar	2158
ISI WOS/SCIE/ESCI/SSCI	1125
IEEE Digital Library	976
Science Direct	661
SCOPUS	556
ACM	525
Springer	360
Others	67
Total	**6428**
References Using Search Strings After Applying Inclusion and Exclusion Criteria	
Google Scholar	-
ISI WOS/SCIE/ESCI/SSCI	563
IEEE Digital Library	488
Science Direct	330
SCOPUS	278
ACM	263
Springer	180
Others	33
Total	**2135**
References Using Search Strings After Reading Abstract and Conclusion	
ISI WOS/SCIE/ESCI/SSCI	282
IEEE Digital Library	244
Science Direct	330
SCOPUS	165
ACM	132
Springer	90
Others	17
Total	**1260**
References Using Search Strings After Reading the Whole Text Including the Finding and Related Results	
ISI WOS/SCIE/ESCI/SSCI	75
IEEE Digital Library	65
Science Direct	46
SCOPUS	35
ACM	35
Springer	24
Others	4
Total	**284**

Figure 2:Search string Process results

General results and discussion

Based on the search process result Table 8, the 284 selected papers are analyzed based on the relevant RQs as shown in Table 7. It has been established that the majority of the selected papers provide answers to RQ1, and RQ3, as 195 out of 284 papers answer the RQ1, whereas 221 papers give answers to RQ3. However, 176 extracted papers offer answers for RQ2. Around 40% (about 113 papers) of the extracted papers contributed to the answers for RQ4, which is the lowest among the rest of the research questions. This could be due to fewer research efforts have been contributed to the literature compared to the other questions, which requires more research to be done to provide more answers on the RQ4.

Discussion and analyses related research questions

After identifying the data extraction strategy and selecting the primary data studies, we designed the data extraction form presented in Table 5 to obtain the data from the primary search process correctly. The studies' articles' information has been identified using data extraction application and should contain fields corresponding to each research question. Then, the quality assessment questions listed in Table 4 are evaluated for each primary study. The primary studies that are selected answer some or all of the research questions. Table 8 and figure3 illustrate Search Process Results while Table 9 and figure 4 show the distribution of publications per year.

RQ1: What are the most common characteristics, similarities, differences, attributes, advantages and disadvantages among the big machine deep learning systems (BiMDLs) in terms of their goal, tasks and function?

The primary objective of this review is to identify, categorize, classify the big machine deep learning systems BiMDLs and their components to determine the advantages and disadvantages of the included interfaces and libraries that occurred due to their differences in parallel computing goals by conducting SLR. The results of our comprehensive investigation were informative.

A list of existing BiMDLs investigation outcomes and the comparison results in several aspects have identified the limitations, common features, the similarities, differences, the advantages, and disadvantages that could add beneficial knowledge to Artificial intelligence AI, Machine learning ML, Deep learning DL, and Data science Researchers and Developers.

TABLE 7: The most related studies to the research topic mapped to RQs

Paper	Year	Type of Publication	Indexed by	Frequencies	RQ1	RQ2	RQ3	RQ4
[34]	2019	Journal	ISI WOS/SCIE	99	√	√	√	
[35]	2020	Journal	ScienceDirect	98	√		√	√
[25]	2019	Journal	Springer	90	√	√	√	
[36]	2019	Journal	ISI WOS/SCIE	87	√	√		√
[37]	2020	Journal	ScienceDirect	84	√	√		
[38]	2018	Journal	ISI WOS/SCIE	80		√	√	√
[39]	2018	Journal	ScienceDirect	80	√		√	
[40]	2019	Journal	ISI WOS/SCIE	74		√	√	
[41]	2019	Journal	ISI WOS/SCIE	72	√		√	√
[42]	2021	Journal	Scopus	70	√	√		√
[43]	2021	Journal	ScienceDirect	70		√	√	
[44]	2018	Conference	IEEE	68	√	√	√	√
[5]	2019	Journal	ISI WOS/SCIE	67	√	√		
[45]	2018	Journal	ISI WOS/SCIE	65		√		√
[46]	2017	Journal	ISI WOS/SCIE	65		√	√	√
[47]	2020	Journal	ISI WOS/SCIE	61	√		√	
[48]	2018	Journal	ScienceDirect	60	√	√		√
[49]	2017	Journal	ISI WOS/ESCI	59	√	√		
[50]	2019	Journal/Conference	ScienceDirect/ACM	58	√		√	√
[51]	2020	Journal	ISI WOS/SCIE	58	√	√	√	
[52]	2019	Conference	IEEE	56	√	√		√
[53]	2020	Journal	ScienceDirect	53	√			
[54]	2018	Journal	IEEE	52		√	√	√
[55]	2020	Journal	ScienceDirect	52	√	√		
[56]	2019	Conference	IEEE/ACM	50	√		√	√
[57]	2020	Journal	ISI WOS/SCIE	50	√	√		
[22]	2019	Journal	Springer	49		√	√	√
[58]	2021	Journal	ScienceDirect	49	√	√	√	
[59]	2021	Journal	ISI WOS/SCIE	48	√			√
[60]	2018	Conference	IEEE	48	√	√	√	
[61]	2020	Journal	Springer	47	√	√	√	
[62]	2019	Journal	ISI WOS/SCIE	47	√		√	√
[63]	2018	Journal	ISI WOS/SCIE	47	√	√	√	
[64]	2018	Conference	IEEE/ACIS	46	√	√		√
[65]	2017	Journal	ScienceDirect	46	√	√	√	
[66]	2019	Journal	ScienceDirect	44	√		√	

[67]	2020	Journal	ScienceDirect	43		√	√	√
[26]	2017	Conference	ACM	43	√	√		√
[68]	2019	Conference	ACM	42	√		√	
[69]	2018	Conference	ISI WOS/SCIE	41	√	√		√
[70]	2020	Journal/Conference	Scopus	41	√	√		√
[71]	2017	Conference	IEEE	41	√		√	√
[72]	2020	Journal	Scopus	40	√	√		
[73]	2020	Journal	ISI WOS/SCIE	40	√		√	√
[74]	2019	Journal	ISI WOS/SCIE	39		√	√	√
[75]	2019	Conference	IEEE	39	√	√		
[76]	2020	Journal	ScienceDirect	39		√	√	√
[77]	2018	Journal	IEEE	38	√		√	
[78]	2017	Conference	IEEE	38	√	√		
[79]	2020	Conference	ACM	37	√		√	√
[80]	2018	Journal	ISI WOS/SCIE	37		√	√	
[81]	2020	Journal	ScienceDirect	36	√			√
[82]	2018	Journal	ProQuest	34		√	√	
[83]	2017	Journal	ISI WOS/ESCI	34	√			√
[2]	2019	Journal	ScienceDirect	34	√	√	√	
[84]	2019	Journal	ISI WOS/SCIE	32	√	√	√	
[85]	2018	Journal	ScienceDirect	32	√	√		√
[86]	2017	Conference	IEEE	32	√		√	
[87]	2020	Journal	ScienceDirect	31		√	√	√
[88]	2016	Journal	ScienceDirect	31	√	√	√	√
[89]	2016	Conference	USENIX	30	√			
[90]	2015	Journal	ISI WOS/SCIE	30		√	√	√
[91]	2020	Journal	ScienceDirect	30		√	√	√
[92]	2020	Journal	IEEE	29	√		√	
[93]	2020	Conference	USENIX/ACSA	29	√	√		√
[94]	2020	Journal	ScienceDirect	29	√		√	√
[95]	2020	Journal	ScienceDirect	29	√	√	√	
[96]	2020	Journal	ScienceDirect	28	√	√		√
[97]	2018	Journal	ISI WOS/SCIE	28	√	√		√
[98]	2020	Journal	ScienceDirect	28	√		√	
[99]	2018	Journal	ISI WOS/SCIE	28	√	√		√
[100]	2018	Conference	ACM	27	√		√	√
[101]	2017	Conference	IEEE	27	√	√	√	
[102]	2019	Conference	ACM	26			√	√
[103]	2019	Journal	ISI WOS/SCIE	26	√	√		
[104]	2018	Journal	ISI WOS/SCIE	26	√		√	√
[105]	2017	Conference	IEEE	25		√	√	
[106]	2017	Conference	NIPS/Stanford Uni	25	√			√
[107]	2021	Journal	Scopus	25	√	√	√	
[108]	2019	Journal	ISI WOS/SCIE	25	√		√	
[11]	2020	Journal	ISI WOS/SCIE	24	√	√	√	√
[109]	2019	Conference	ACM	24	√	√	√	
[110]	2020	Journal	ScienceDirect	23	√	√		√
[111]	2019	Conference	IEEE	23			√	
[112]	2020	Conference	Scopus	22	√	√		√
[113]	2018	Journal	IEEE	22	√		√	
[114]	2020	Journal	ISI WOS/SCIE	22		√	√	√
[115]	2018	Conference	ACM/IEEE	22	√		√	
[116]	2018	Journal	ISI WOS/SCIE	22	√	√		√
[117]	2020	Conference	IEEE	21	√		√	
[118]	2020	Conference	IEEE	21		√	√	

[119]	2020	Journal	ISI WOS/SSCI	21		√	√	√
[120]	2018	Journal	ScienceDirect	20	√		√	
[121]	2019	Conference	NeurIps/Canada	19	√	√	√	
[122]	2019	Journal	IEEE/SCIMAGO	18	√		√	√
[123]	2019	Journal	ISI WOS/ESCI	18	√	√	√	
[124]	2020	Journal	Scopus	17	√			
[125]	2017	Conference	IEEE	17	√	√	√	
[126]	2020	Journal	Springer	17	√		√	√
[127]	2019	Conference	HAL	17	√	√		
[128]	2020	Journal	ScienceDirect	17	√		√	√
[129]	2018	Journal	ISI WOS/SSCI	17	√	√	√	
[130]	2016	Journal	ISI WOS/SCIE	17	√	√	√	
[131]	2020	Journal	Scopus	16		√	√	√
[132]	2020	Journal	ScienceDirect	16	√	√	√	
[133]	2016	Conference	Springer	16		√	√	√
[134]	2019	Conference	IEEE	16	√		√	
[135]	2017	Journal	ISI WOS/SCIE	16	√	√		√
[136]	2018	Journal	ISI WOS/SCIE	16	√	√	√	
[137]	2018	Journal	ISI WOS/SCIE	15	√		√	
[138]	2017	Conference	IEEE	15		√	√	
[139]	2020	Journal	Springer	15	√	√	√	√
[140]	2020	Journal	ScienceDirect	15	√		√	
[141]	2016	Conference	IEEE	15	√	√	√	
[142]	2020	Conference	IEEE/ACM	15		√	√	√
[143]	2019	Journal	Scopus	15	√			
[144]	2018	Journal	ScienceDirect	14		√	√	√
[145]	2018	Journal	IEEE	14	√		√	
[146]	2019	Conference	ACM	14		√	√	
[147]	2018	PhD Thesis	California Uni	14	√	√		√
[148]	2020	Conference	ACM/IEEE	14		√	√	
[149]	2016	Conference	Scopus /EMNLP	14	√		√	√
[150]	2020	Journal	ScienceDirect	14	√	√	√	
[151]	2017	Conference	ACM	13	√	√	√	
[152]	2020	Journal	ISI WOS/SCIE	13	√	√	√	
[153]	2020	Journal	ScienceDirect	13	√		√	
[154]	2018	Journal	ACM	13		√	√	√
[155]	2018	Conference	ACM	13	√			
[156]	2020	Journal	ISI WOS/ESCI	13	√	√	√	
[157]	2020	Conference	Scopus	12	√			√
[158]	2017	Journal	Springer	12		√	√	
[24]	2016	Conference	ACM/USENIX	12	√			
[159]	2019	Conference	Springer	12	√	√	√	
[160]	2018	Conference	Springer	12	√		√	
[161]	2018	Conference	IEEE	12		√	√	√
[162]	2020	Journal	ISI WOS/ESCI	12	√		√	
[163]	2019	Conference	SPIE/Scopus/ WOS	12		√		√
[164]	2018	Conference	Springer	12	√	√	√	
[165]	2020	Journal	ScienceDirect	12	√		√	
[166]	2020	Journal	ACM	11		√	√	√
[167]	2019	Conference	ACM/IEEE	11		√	√	
[168]	2017	Conference	IEEE	11	√		√	
[169]	2020	Conference	IEEE	11		√	√	√
[170]	2020	Journal	Scopus/AAAI	11	√	√		
[171]	2020	Conference	SPIE/Scopus/WOS	11	√		√	√

[172]	2018	Conference	IEEE	11		√	√	
[173]	2017	Journal	WOS/CEEOL	11	√	√		√
[29]	2019	Journal	ISI WOS/SCIE	11	√		√	
[174]	2018	Conference	IEEE	11		√	√	
[175]	2018	Journal	IEEE/JOSS	10	√		√	
[176]	2018	Conference	ACM	10	√	√		
[177]	2019	Conference	ACM	10			√	√
[178]	2019	Conference	ACM	10		√	√	
[179]	2020	Journal	ISI WOS/SCIE	10	√		√	√
[180]	2017	Conference	IEEE	10	√	√		
[181]	2020	Journal	ScienceDirect	10			√	√
[182]	2016	Journal	ScienceDirect	10	√	√		
[183]	2016	Journal	ScienceDirect	10		√	√	√
[184]	2020	Conference	ISI WOS/SCIE	10	√		√	
[185]	2018	Journal	ScienceDirect	10		√	√	
[186]	2020	Journal	ISI WOS/SCIE	9	√		√	
[187]	2020	Conference	IEEE	9	√	√		
[188]	2017	Conference	IEEE	9		√	√	√
[189]	2019	Journal	ScienceDirect	9	√		√	
[8]	2020	Journal	ACM	8		√	√	
[190]	2018	Conference	ACM	8	√	√		
[191]	2018	Journal	Scopus	8			√	√
[192]	2020	Conference	PMLR/ Vienna Austria	8	√	√	√	
[193]	2019	Conference	Springer	8		√		√
[194]	2016	Journal	ISI WOS/SCIE	8	√		√	
[195]	2018	Journal	ProQuest	8		√	√	√
[196]	2019	Conference	Springer	8	√		√	
[197]	2018	Journal	ISI WOS/ESCI	8		√	√	√
[198]	2018	Journal	Springer	8		√	√	
[199]	2019	Journal	Springer	8	√		√	
[200]	2018	Journal	Scopus	8	√			
[201]	2015	Journal	ISI WOS/ESCI	8		√	√	√
[202]	2019	Journal	Scopus	7	√			
[203]	2019	Journal	Springer	7		√	√	√
[23]	2018	Journal	PROQUEST/ISI WOS/SCIE	7	√		√	
[204]	2019	Journal	EMERALD	7		√		√
[205]	2017	Conference	Springer	7	√		√	
[206]	2020	Journal	ISI WOS/SCIE	7	√	√	√	
[207]	2017	Journal	ISI WOS/SCIE	7			√	√
[208]	2016	Journal	ISI WOS/SCIE	7		√	√	
[209]	2019	Journal	ScienceDirect	7	√		√	√
[210]	2021	Journal	ScienceDirect	7	√	√	√	
[211]	2018	Conference	IEEE	6	√		√	
[212]	2017	Conference	IEEE/ACM	6		√	√	
[213]	2018	Conference	IEEE	6	√		√	√
[214]	2019	Journal	ISI WOS/SCIE	6		√	√	
[215]	2019	Journal	ISI WOS/SCIE	5	√		√	
[216]	2020	Journal	Scopus	5		√		√
[217]	2017	Conference	IEEE	4	√		√	
[218]	2016	Journal	ISI WOS/SCIE	4		√	√	√
[219]	2018	Conference	IEEE	4	√	√	√	
[220]	2016	Journal	ISI WOS/ESCI	4	√		√	√
[221]	2019	Journal	IEEE	4		√	√	
[222]	2019	Conference	ISI WOS/SCIE	4	√		√	

[223]	2016	Journal	ISI WOS/SCIE	4		√	√	
[224]	2020	Conference	ACM	4			√	√
[225]	2017	Conference	IEEE/ACM	3	√	√	√	
[18]	2018	Journal	Scopus	3		√		√
[20]	2018	Conference	IEEE	3	√	√	√	
[226]	2019	Journal	ScienceDirect	3	√		√	√
[7]	2020	Journal	Scopus / PNAS	3		√	√	
[14]	2019	Journal	Scopus	3	√		√	√
[227]	2016	Journal	ISI WOS/SCIE	3		√	√	
[228]	2019	Journal	Scopus	3	√		√	√
[229]	2015	Conference	ACM	3		√	√	
[230]	2017	Journal	Springer	3	√		√	√
[231]	2019	Journal	ScienceDirect	3		√	√	
[232]	2016	Conference	IEEE/ACM	3	√		√	√
[233]	2018	Conference	IEEE	2		√	√	
[234]	2019	Journal	Scopus	2	√		√	
[235]	2019	Conference	IEEE	2	√	√	√	
[236]	2016	Conference	IEEE	2		√	√	
[237]	2018	Conference	IEEE	2	√	√		√
[238]	2018	Journal	ISI WOS/SCIE	2	√		√	
[239]	2017	Conference	Scopus / NIPS	2		√	√	
[240]	2019	Journal	Springer	2	√		√	
[241]	2021	Journal	ISI WOS/SCIE	2		√	√	
[242]	2019	Journal	ScienceDirect	2	√		√	√
[243]	2018	Conference	IEEE	2	√	√	√	
[244]	2016	Journal	ISI WOS/SCIE	2			√	√
[245]	2019	Journal	ISI WOS/SCIE	2	√	√	√	
[246]	2019	Conference	IEEE	2	√		√	√
[247]	2018	Journal	Scopus	2		√	√	
[9]	2017	Journal	ISI WOS/SCIE	2	√		√	
[248]	2020	Journal	ISI WOS/SCIE	2	√	√		√
[249]	2020	Journal	ISI WOS/SCIE	2	√		√	
[250]	2018	Conference	IEEE	2	√	√	√	
[251]	2016	Journal	ScienceDirect	2		√		√
[252]	2020	Journal	Springer	2	√		√	
[253]	2020	Journal	ISI WOS/SCIE	2	√	√	√	
[254]	2020	Journal	ISI WOS/SCIE	2		√	√	
[255]	2017	Journal	ISI WOS/SCIE	2	√		√	
[256]	2018	Conference	Springer	2	√	√		√
[257]	2015	Journal	ScienceDirect	2	√	√	√	
[258]	2019	Journal	IEEE	2	√		√	
[259]	2019	Journal	Scopus	2		√	√	
[6]	2016	Journal	Wiley	2	√		√	√
[260]	2019	Journal	Springer	2		√	√	
[261]	2019	Journal	ISI WOS/SCIE	2		√	√	
[262]	2019	Conference	IEEE	2	√		√	√
[263]	2018	Journal	Scopus	2	√	√	√	
[264]	2020	Journal	Scopus	1	√	√	√	
[265]	2018	Journal	ScienceDirect	1	√		√	
[266]	2016	Journal	Scopus	1	√	√		√
[267]	2019	Journal	Scopus	1	√		√	
[268]	2020	Journal	Scopus	1	√	√	√	
[269]	2018	Conference	IEEE	1	√			√
[13]	2016	Conference	ISI WOS/SCIE	1	√	√	√	
[270]	2016	eBook	Springer	1	√	√	√	

[271]	2019	Journal	ACM	1			√	√
[272]	2018	Conference	ACM	1		√	√	
[273]	2019	Journal	Springer	1	√	√	√	
[274]	2016	Conference	IEEE	1	√		√	
[275]	2017	Conference	IEEE	1	√	√	√	
[276]	2019	Conference	IEEE	1		√	√	
[277]	2017	Journal	ScienceDirect	1	√	√		√
[278]	2018	Journal	ISI WOS/SCIE	1	√		√	
[279]	2017	Conference	IEEE	1	√	√	√	
[280]	2018	Journal	Springer	1	√		√	√
[281]	2017	Conference	ACM	1		√	√	
[282]	2017	Conference	IEEE	1	√		√	
[283]	2019	Journal	ISI WOS/SCIE	1		√	√	√
[284]	2017	Conference	IEEE	1	√		√	
[285]	2015	Conference	ACM	1		√	√	
[286]	2019	Conference	ACM	1	√	√		√
[10]	2019	Journal	ISI WOS/SCIE	1			√	
[287]	2020	Journal	ISI WOS/SCIE	1	√	√		√
[288]	2019	Conference	IEEE	1	√		√	
[289]	2019	Journal	IEEE	1		√		√
[290]	2016	Conference	Scopus	1	√	√	√	
[291]	2019	Conference	ACM	1	√		√	
[292]	2020	Conference	ACM	1		√	√	
[293]	2020	Conference	Scopus	1	√		√	√
[294]	2019	Conference	ACM	1		√	√	
[12]	2019	Journal	Scopus	1	√		√	√
[295]	2020	Journal	ISI WOS/SCIE	1		√	√	
[296]	2018	Conference	IEEE	1	√		√	√
[297]	2018	Conference	IEEE	1		√	√	
[19]	2020	Journal	Scopus	1	√	√	√	

TABLE 8: Search Process Results

Research Resources	Potentially eligible studies	Selected Studies
Google Scholar	2158	-
ISI WOS/SCIE/ESCI/SSCI	1125	75
IEEE Digital Library	976	65
Science Direct	661	46
SCOPUS	556	35
ACM	525	35
Springer	360	24
Others	67	4
Total	6428	284

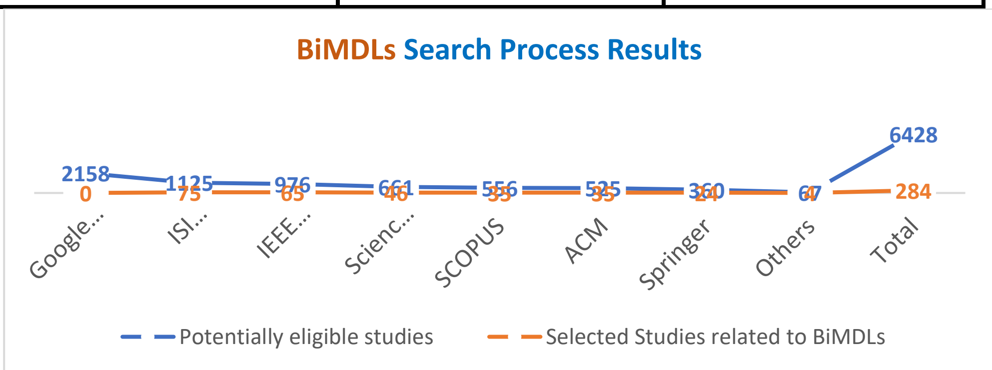


Figure 3: BiMDLs Search Process Results diagram

TABLE 9: Distribution of publications \ per year

Year	Number of studies	Percentage (%)
2015	5	2 %
2016	25	9 %
2017	35	12 %
2018	66	23 %
2019	75	26 %
2020	72	25 %
2021	6	2 %

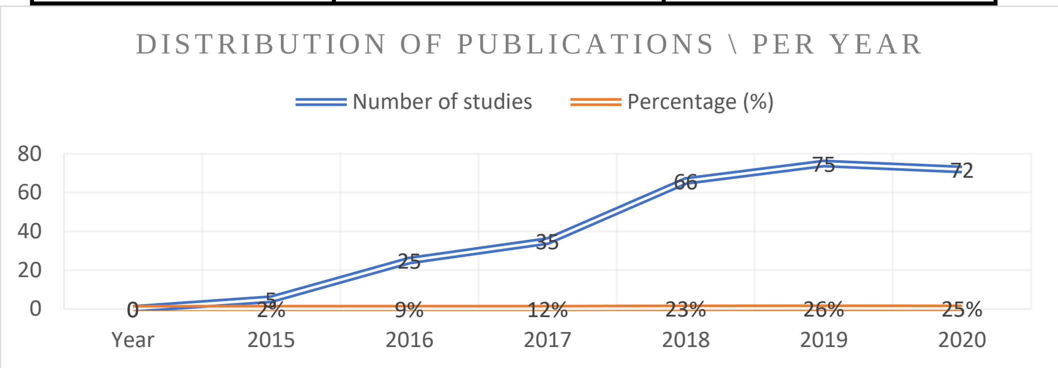


Figure 4: Distribution of publications diagram \ per year

Comparisons

Comparisonof the most BiMDLs popular Frameworks

Undoubtedly, BiMDLs framework has met with tremendous success in a broad range of fields such as artificial vision, voice recognition, big data processing, and immersive comprehension and the latest state-of-the-art ML and DL implementations.Moreover, the size of tensor in deep learning is huge, this huge amount could reach more than 200 million dimensions with 8 Billion points, or what became known as (The curse of dimensions) [7], [12]–[14]; [8]; [9]; [10]. And to solve this problem (High-Dimensionally Problem), Developers, Researchers, Programmers and even Software companies have designed a myriad of frameworks under the name of big machine/Deep learning Systems, to handle the complex matrices and mathematical operations [11]; [15].For instance, Facebook's Torch/PyTorch and Caffe2 [16], [17]; [18],[298]; [25], [121], [299]; University of Montreal's Theano, Google's TensorFlow [19]–[22], Apache's MxNet, and Microsoft's programming libraries with fixed user interfaces [23] and Microsoft's CNTK [23], [137], [138] as shown in Table 8. However, the most machine deep learning frameworks are converging to common pipeline design, similar in terms of purpose, goal and mission [22], [24]–[27].

The first comparison was conducted between the mostthe most BiMDLs popular Frameworks**.** Therefore,based on Table10, We found that when the framework works within the environment to which it belongs, the compatibility is excellent, and the code can be reused or even reversed and converted smoothly. However, when the environment differs, there is less or no compatibility with other frameworks.

TABLE 10:Comparison results of the most the most BiMDLs popular Frameworks

Framework	creator	Purpose	Core Language	Platform	Interface	CUDA Support	Pretrained Model	Multi-GPU	Multi-Threaded CPU	Compatibility within the same environment	Compatible with the other frameworks

Cafe/Cafe2	Berkeley Vision and Learning Center 2013 caffe 2017 caffe2	Caffe, a Convolutional Architecture for Fast Feature Embedding, is a Deep Learning framework that focuses on image classification and segmentation and can be deployed on both CPU and GPU.	C++	Linux, Mac OS X, Windows	Python, MATLAB	Yes	Yes	Yes (only data parallel)	Yes (BLAS)	Yes	No
TensorFlow	Google Brain team 2015	It is a high-performance numerical computation software library with support for Machine Learning and Deep Learning architectures deployed in CPU, GPU, and TPU.	C++	Linux, Mac OS X, Windows	Python (Keras), C/C++, Java, Go, R	Yes	Yes	Yes (Most flexible)	Yes (Eigen)	Yes	No
Theano	University de Montréal 2007	Smooth mathematical operations on multidimensional arrays are performed by a GPU-compatible Python library that is tightly integrated with NumPy.	Python	Cross-platform	Python	Yes	Through Lasagne's model zoo	Not perfect	Yes (Blas, conv2D, limited, OpenMP)	Yes	No
Torch/PyTorch	Adam Paszke, Sam Gross, SoumithChintala, Gregory Chanan (Facebook) 2016	A Python-based platform for distributed training and performance evaluation that is supported by main cloud applications.	Python, C, C++, CUDA	Linux, macOS, Windows	Python, C++, Julia	Yes	Yes	Yes	Yes (widely Used)	Yes	No
CNTK	Microsoft Research 2016	Microsoft Cognitive Toolkit (CNTK) is a DL Framework that describes computations using directed graphs.	C++	Windows, Linux (OSX via Docker on roadmap)	Python, C++, Command line	Yes	Yes	Yes	Yes	Yes	No
MXNet	Apache Software Foundation 2015	MxNet is a free and open-source deep learning software framework for training and deploying deep neural networks.	Small C++ core library	Linux, Mac OS X,	C++, Python, Julia, MATLAB, JavaScript, Go, R, Scala, Perl	Yes	Yes	Yes	Yes (OpenMP)	Yes	No
Deeplearning4j	Skymind engineering team; Deeplearning4j	DL4j is a Java programming library for the Java virtual machine	Java	Linux, Mac OS X, Windows,	Java, Scala, Clojure, Python (Keras)	Yes	Yes	Yes	Yes (OpeMP)	Yes	No

	community; originally Adam Gibson 2014	(JVM). It is a framework wi th comprehensive support for deep learning algorithms.		Androi d (Cross platfor m)							
Chainer	Preferred Networks and supported by IBM, Intel, NVidia 2015	Supports CUDA computation and multiple GPU implementation	Python	Linux, macOS	Python	Yes	Yes	Yes, with a little effo rt	Yes	Yes	No
BigDL	Jason Dai (Intel) 2016	BigDL is aApache Spark distributed deep learning library that supports programming languages Scala and Python.	Scala	Apache Spark	Scala, Python	No	Yes	No	Yes	Yes	No
ONNX	Facebook, Microsoft 2017	(ONNX) The Open Neural Network Exchange is an artificial intelligence ecosystem that is open source.	C++, P ython	Project Brainw ave pla tform	C++, Pyt hon	Yes	Yes	Yes	Yes	yes	Compatible with some operatorsan d few BiMDLs frameworks, such as:PyTorch and Caffe2. With significant latency

Comparison in terms of distribution and parallelization

Distributed BiMDLs can naturally solve the problem of algorithm complexity and memory limitation in large scale machine learning.Distributed machine learning can achieve efficiency by loading data in parallel and fault tolerance by replicating data between machines. On the other hand, Big machine deep learning systems come with many possibilities for parallelization. The main advantage of data parallelism is that it is applicable to any ML or DL model architecture without further domain knowledge of the model. It scales well for compute-intensive operations but has only a few parameters, such as CNNs. However, data parallelism is limited for operations with many parameters, as parameter synchronization becomes the bottleneck[254].Besides, the use of different training processes to train different classifiers from distributed data sets increases the possibility of achieving greater precision, especially in a large domain. However, the design and implementation of efficient and clearly correct parallel algorithms represent a major challenge[65]. Table11 literature review Illustrated thesecond comparisonresults in terms of the distribution and parallelization:

TABLE 11: BiMDLs Comparison results in terms of distribution and parallelization

Framework	Comparison Description
Cafe	• There's no native support for distribution
Caffe2	• Only decentralized • Only synchronous • Only Model Quantization • Only Gradient Quantization • Only Communication Scheduling [254]
TensorFlo w	• Centralization • Terms synchronous and asynchronous that used interchangeably. • Supported model quantization • No support for gradient quantization • No support for communication scheduling [254]
Theano	• No native support for distribution. [254]; [283]

Torch/PyTorch	• Centralization and decentralization • Synchronized and asynchronous • Model quantization and gradient quantization are not supported • Communication scheduling is not supported [254]
CNTK	• Centralization and decentralization • BMUF-based bounded asynchronous training[274] • Model quantization is not supported; however, 1-bit gradient quantization[300]is supported • Communication planning is not supported.
MXNet	• Only centralized. • Synchronized and asynchronous • Supported model quantization • Error-feedback with 2-bit gradient quantization are supported [283]. • Communication scheduling not supported [254]
Deeplearning4j	• Centralization and Decentralization • Synchronized and asynchronous • Supported Model quantization. • Strom's modified 1-bit gradient quantization is supported. [301]; [302] ; [254] • Communication planningis not supported
Keras	• Model quantization is supported • Keras-based DL framework is a proper environment for the implementation of a Higher-level concept[254].
Chainer	• Only decentralized • Only synchronous • Model quantization is not supported • Gradient quantization is not supported • Communication Scheduling is not supported
SINGA	• Centralization and decentralization • Synchronized and asynchronous • Model quantization is not supported • Gradient quantization is not supported • Communication Scheduling is not supported

The third comparison includes two of the most famous, meaningful, and influential libraries within big machine deep learning systems, BiMDLs: TensorFlow and Keras.

Comparison ofTensorFlow and Keras Frameworks

Keras: Keras is an Open-Source neural network Framework. It is intended to be modular, quick, and simple to use. It is a valuable library for building any DL algorithm. Keras is a TensorFlow, CNTK, or Theano compatible and can be deployed in GPU and CPU.

TensorFlow:TensorFlow framework is an open-source machine deep learning Interface. It was built to run on multiple CPUs or GPUs and even mobile operating systems. Table 12 illustrates the results of the comparison review as flowing:

TABLE 12: Comparison results example Between TensorFlow and KERAS

Comparisons	Keras	TensorFlow
Founder	• It began as a project by Chollet François and was developed by a group of people.	• The Google Brain team developed it.
Definition	• KERAS is an Open-Source Neural Network library. It is intended to be modular, quick, and simple to use. It is a valuable library for building any deep	• TensorFlow is an open-source deep-learning library. It was created to run on multiple CPUs or GPUs and even mobile operating systems

	learning algorithm	
APIs level	• KERAS is an API with a high level of abstraction. It runs on top of TensorFlow, CNTK, or Theano and can be deployed in CPU and GPU.	• TensorFlow is a framework that provides both high-level and low-level APIs.
Ease of use	• KERAS is simple to use if we are familiar with Python.	• It needs to learn the syntax of using various TensorFlow functions.
Perfection	• KERAS is Perfect for quick implementations.	• It is ideal for deep learning research, complex networks.
Debugging	• It Uses another API debug tool such as TFDBG.	• We can use Tensor board visualization tools for debugging.
Language	• It is Written in Python, a wrapper for Theano, TensorFlow, and CNTK.	• It is written mostly in C++, Java, and Python
Hardness and flexibility	• KERAS has a straightforward architecture that is both readable and concise.	• TensorFlow is not very simple to use.
Debugging challenge	• In the KERAS framework, there is a significantly less frequent need to debug simple networks.	• It is pretty challenging to perform debugging in TensorFlow.
Dataset size	• KERAS is usually used for small datasets.	• TensorFlow used for large datasets
Community support	• Community support is minimal, (There is little community support)	• A large community of tech companies backs TensorFlow
Performance level	• It can be used for low-performance models.	• It is used for high-performance models
Advantage	• It reduces the number of user actions required for repeated use cases. • Provide actionable feedback in the event of a user error. • KERAS provides a simple, consistent interface optimized for everyday use cases. • It assists the user in writing custom building blocks to express new research ideas. • Create new layers, metrics, and models that are cutting-edge. • Provide a quick and straightforward prototyping service.	• Provides Python and APIs, making them easier to work with. • Should be used to train and serve models in live mode to actual customers. • The TensorFlow framework is compatible with both CPU and GPU computing devices. • It enables us to execute a subset of a graph, allowing us to retrieve discrete data. • Provides a faster compilation time than other deep learning frameworks. • It enables automatic differentiation, which benefits gradient-based machine learning algorithms.
Disadvantage	• Keras is more complex and a less flexible framework to use • There are no RBM (Restricted Boltzmann Machines), for instance, • There are fewer enterprises available online than TensorFlow • The Multi-GPU is not fully functional.	• TensorFlow is slow and inefficient when compared to other Python frameworks. • Nvidia GPU support is limited to language support. • User must have a solid understanding of advanced calculus and linear algebra and prior experience with

		machine learning. • Because TensorFlow has a unique structure, it is difficult to find and debug errors. • It is a shallow level with a steep learning curve.
Features	• Focusing on the user experience. • Multi-backend and multi-platform support. • Models can be created quickly. • Allows for simple and quick prototyping • Supports convolutional networks. • Support for recurrent networks • KERAS is expressive, adaptable, and ideal for exploratory research. • KERAS is a Python-based framework that facilitates debugging and exploration. • A highly modular neural network library written in Python •Designed with the goal of allowing for rapid experimentation	• Python tools for faster debugging • Dynamic models with Python control flow • Support for custom and higher-order gradients • TensorFlow provides multiple levels of abstraction, making it easier to build and train models. • TensorFlow enables us to quickly train and deploy our model, regardless of the language or platform we use. • TensorFlow offers flexibility and control through features such as the KERAS Functional API and Model • Well-documented and simple to understand • Probably the most popular Python library that is simple to use.

Comparison ofPyTorch 0.3.0, TensorFlow 1.4.0, and Cafe2 0.8.1

The obtained results from previous work demonstrated that many benchmarks were conducted. However, there is a need for more efforts to aid in select the appropriate BiMDLs framework or find the next deep learning hot issues from the evolution of the framework. As an example of the earlier benchmarks, Wang et al.[25] Provided a benchmark test as shown in Table 13. Benchmark was performed for image processing between three deep learning frameworks, which are: TensorFlow1.4.0, PyTorch 0.3.0, and Caffe2 0.8.1 with different datasets and GPUs. Test environment's: CUDA 9.0.176, CuDNN 7.0.0.5, NVIDIA driver 387.34, a 1080Ti GPU with 11GB GDDR5. Except where noted. Network structure: VGG16 and Resnet152. The results are based on the models being run with images of size 224 224 3 and a batch size of 16. "Eval" displays the average duration of a single forward pass over 20 passes. "Train" displays the average duration of a pair of forwarding and backward passes over 20 runs. The varied heterogeneous results illustrated that there are apparent mismatches between BiMDLs Frameworks. Therefore, if we want to integrate the existing machine deep learning algorithms into the product, then we recommend starting with the BiMDLs unified framework, although the need to use the Java language, while the results will be satisfactory.

TABLE13: Benchmark on tree BiMDLs frameworks and GPUs

	Precision	Vgg16 eval (ms)	Vgg16 train (ms)	ResNet152 eval (ms)	ResNet152 train (ms)	Compatible with other frameworks
PyTorch 0.3.0	32-bit	39.3	131.9	57.8	206.4	No
	16-bit	33.5	117.6	46.9	193.5	
TensorFlow 1.4.0	32-bit	43.4	131.3	69.6	300.6	No
	16-bit	38.6	121.1	53.9	257.0	

Cafe2 0.8.1	32-bit 16-bit	47.0 40.1	158.9 137.8	77,9 61.7	223.9 184.1	No

Table 14, Provides another benchmark test among Caffe, TensorFlow, and Torch frameworks was conducted (in four convolutional models) test use a six-core Intel Core i7-5930K CPU at 3.5 GHz and an NVIDIA Ti-tan X GPU[25]. TensorFlow achieves faster step times than Cafe and high performance that is within 6% of the most recent Torch version. TensorFlow and Torch have comparable performance because they both use the same version of the cuDNN, a GPU accelerated library, which executes the convolution and pooling operations on the critical path for training. Caffe performs these operations with open-source implementations that are simpler but less efficient than cuDNN. TensorFlow is also user-friendly, while TensorFlow-based high-level APIs, such as TFlearn and Tensor Layer, have existed in the same way that Theano has.

Therefore, it is beneficial to note that high-level APIs are developed to aid developers in building complex interfaces more efficiently. At some point, the use of high-level APIs may result in longer runtimes. The varied heterogeneous results shows that there is apparent incompatibility between the three BiMDLs Frameworks, and this is further evidence that these systems need more work to enhance the compatibility between them.

TABLE 14:

Training four convolutional models with different BiMDLs libraries on a single GPU in milliseconds.

	GoogleNet	OxfordNet	Overfeat	AlexNet	Compatible with other frameworks
Caffe	1935	1068	823	324	No
TensorFlow	445	540	279	81	No
Torch	470	529	268	81	No

All results in this table are for training with 32-bit floats

Analysis thecurrent open issues and the related challenges

This sub-section will provide informative details that effectively cover the RQ2 and RQ3 answers and partly provide some indicative directions related to the RQ4 answer.

RQ2: What are the main open issues and challenges of the current big machine deep learning systems BiMDLs?

The research has focused on several open issues: After analyzing the literature of big machine deep learning systems and their related frameworks, we found that this is still an open research domain, and some issues need further exploration, discussion to find the appropriate solution. Moreover, in this study, we extracted and analyzed roughly 43 factors affecting Big Machine deep learning systems challenges in different ways, followed by a more significant number of dimensions representing clear indications of the open issues under study. Then we supported them by linking them to the relevant theories to strengthen the quality, reliability, and realism of evidence. We believe that these valuable factors and related dimensions gathered and analyzed from several aspects will provide a satisfactory answer to RQ3.

Furthermore,about *RQ3: What are the critical factors and dimensions that affect the existing big machine deep learning systems BiMDLs?*The answer of RQ2 and RQ3 which associated to each other includes several significant issues, such as the lack of compatibility among machine Frameworks, the difficulty of code generation and conversion, and the lack of benchmarks within big machine deep learning frameworks, as well as the difficulty of selecting the proper techniques among big machine deep learning systems. In this sub-section, the answers related to research questions are provided as follows:

First Issue:Lack of Compatibility among Big machinedeep learning Systems (BiMDLs)

One of the most critical aspects of this section is to address the compatibility level among big machine deep learning systems in terms of the type of programming language. Some frameworks can use the host language's objects instead of tensors as the operator arguments. For example, the beginning argument of MXNet's slice operator is a Python tuple". "When outputting the model, only the final value of such argument will be written while its full computation history is lost." Therefore, "this hybrid programming could result in conversion failure if the source framework (e.g., TensorFlow) defines the same argument as a tensor.

"However, the complete solution relies on whether the target framework has a built-in tensor-to-value capability such that we should directly convert the subgraph representing the argument's computation history. Table 15 provides a summary of the literatures that indicate to the compatibility level in terms of the type of programming language associated to the related theory, factor and open issue dimension.

TABLE 15: ***LoC***(Lack of compatibility) within big machine deep learning systems BiMDLs in terms of Programming language

Factors ID	Theory	Open issues Dimensions	References	Total of Ref.
LoC (Lack of compatibility within big machine deep learning systems BiMDLs (Lacking in terms of Programming language)	Theory of computation/programming language Theory	Compatibility level in terms of programming language and its related libraries (e.g., Python libraries for ML: SciPy, NumPy, Keras, Theano, etc..)	[79]; [74]; [22]; [137]; [25]; [78]; [97]; [121]; [125]; [83]**;** [144]; [93]; [80]; [138]; [151]; [158]; [145]; [75]; [26]; [82]; [191]; [190]; [118]; [24]; [61]; [202]; [89]; [102]; [37]; [35]; [59]; [38]; [110]; [41]; [42]; [45]; [34]; [72]; [70]; [105]; [71]; [60]; [54]; [5]; [44]; [52]; [46]; [63].	48 Papers

The type of random-access memory RAM and memory size directly affect the level of compatibility within the big machine deep learning frameworks. Therefore, the second dimension of the compatibility among BiMDLs is in terms ***(***Lack of memory type/size), as shown in Table 16.

TABLE 16: ***LoC*** (Lack of compatibility) within big machine deep learning systems BiMDLs in terms of memory type/size)

Factors ID	Theory	Open issues Dimensions	References	Total of Ref.
LoC (Lack of compatibility within big machine deep learning systems BiMDLs (In terms of memory type/size)	Information processing Theory (IP theory, IPT)/Transactive memory theory	Compatibility level in terms of memory RAM type, and RAM capacity (also, In terms of – CPEs, MPEs- and LDM, DMA)	[79]; [22]; [25]; [78]; [83]**;** [144]; [225]; [84]; [93]; [36]; [176]; [151]; [157]; [39]; [26]; [82]; [100]; [105]; [64]; [121]; [111]; [89]; [102]; [37]; [107]; [62]; [35]; [98]; [103]; [59]; [41]; [42]; [45]; [34]; [72]; [108]; [77]; [70]; [71]; [60]; [54]; [5]; [49]; [44]; [52]; [46]; [50]; [63].	48 Papers

Moreover, the type of computer device also affects the level of compatibility among BiMDLs as shown in Table 17. That is, if the devices on which codes, software, or datasets are exchanged are from the same type, then the compatibility level will be higher such as the exchange between intel with intel, Nvidia with Nvidia, and AMD, with AMD, and vice versa.

TABLE 17: ***LoC***(Lack of compatibility) within big machine deep learning systems BiMDLs in terms of device type)

Factors ID	Theory	Open issues Dimensions	References	Total of Ref.
LoC (Lack of compatibility within big machine deep learning systems BiMDLs	Computational complexity theory	Compatibility level in terms of Device Type (e.g., IBM/NVIDIA/INTEL/AMD etc.) / Platform Compatibility	[79]; [22]; [25]; [78]; [83]**;** [144]; [126]; [93]; [80]; [36]; [151]; [158]; [157]; [26]; [82]; [190]; [100]; [118]; [24]; [105]; [64]; [121]; [111]; [89]; [113]; [146]; [37]; [85]; [87]; [35]; [103]; [59]; [41]; [45]; [34]; [77]; [70]; [105];	49 Papers

(Lacking in terms of type of computer device)			[71]; [60]; [54]; [5]; [56]; [49]; [44]; [52]; [46]; [50]; [63].	

The compatibility among the deep learning techniques and how to fully discover the configuration space from all existing work remains an open question [82]. However, a type of graphic card is also called a display card or video card. A circuit board in the computer with specialized hardware optimized to display high-quality graphics at a high-speed rate. The better the graphic card, the greater the compatibility among big machine deep learning open sources. Table 18:Provides more details and a summary of the literature that indicates the compatibility level in terms of the type of graphic card associated with the related theory, factor, and open issue dimension.

TABLE 18:*LoC* (Lack of compatibility) within big machine deep learning systems BiMDLs in terms of graphics and graphic card

Factors ID	Theory	Open issues Dimensions	References	Total of Ref.
LoC (Lack of compatibility within big machine deep learning systems BiMDLs (In terms of graphics and graphic card)	Computational complexity theory	Compatibility level in terms of Graphic card type and (e.g., graph-based optimization and knowledge graph)	[22]; [25]; [78]; [84]; [93]; [80]; [36]; [151]; [238]; [169]; [157]; [39]; [26]; [82]; [100]; [105]; [64]; [89]; [113]; [106]; [37]; [66]; [107]; [35]; [98]; [81]; [103]; [59]; [53]; [48]; [38]; [43]; [58]; [41]; [45]; [34]; [108]; [77]; [70]; [40]; [74]; [71]; [60]; [54]; [5]; [44]; [52]; [46]; [50]; [63].	50 Papers

Furthermore, the previous work showed that the shortage is in interoperability among the big machine and deep learning systems is because of the open sources themselves as they have differences in terms of the mission, the goal, and the purpose for which it was designed[26], [82], [137].

The lack of interoperability often rapidly testing potential various machine deep frameworks. IT will invest time building new ecosystems or make attempts to re-implement them within their systems[26], [229]. Table 19 illustrates further details in terms of the lack of interoperability among BiMDLs.

TABLE 19: ***LoC*** (Lack of compatibility) within big machine deep learning systems BiMDLs in terms of the lacking in interoperability among tools and techniques themselves.

Factors ID	Theory	Open issues Dimensions	References	Total of Ref.
LoC (Lack of compatibility within big machine deep learning systems BiMDLs (Lacking inInteroperability)	Computational complexity theory	Lack of Interoperability among BiMDLs frameworks Techniques and libraries.	[79]; [77]; [157]; [263]; [124]; [8]; [22]; [78]; [121];[264]; [125]; [137]; [82]; [83]**;** [144]; [40]; [92]; [166]; [175]; [112]; [84]; [117]; [80]; [36]; [186]; [151]; [67]; [26]; [190]; [100]; [118]; [211]; [122]; [25]; [47]; [37]; [62]; [87]; [35]; [59]; [42]; [45]; [61]; [34]; [70]; [71]; [5]; [56];	51 Papers

			[44]; [52]; [46].	

As shown in Table 20, Most DL-frameworks are based on CUDA and CuDNN are Non-compatible CUDA libraries and in both the Frameworks and CUDA 's accelerated update speed, both are often misaligned and thus incompatible. For example, the pre-built TensorFlow version 1.5 package requires CUDA 9.0 while the documentation for CUDA 8.0 is still written[22], [176] as shown in Table 6. Moreover, deep learning approaches have indicated some great outcomes when all is said in done object identification and that as it may, there still stay principal difficulties to be tended to, especially in distinguishing o various scales [82].

TABLE 20: ***LoC*** Lack of compatibility within big machine deep learning systems in terms of type of computing platform

Factors ID	Theory	Open issues Dimensions	References	Total of Ref.
LoC (Lack of compatibility within big machine deep learning systems BiMDLs) (In terms of type of computing platform)	Computational complexity theory	Compatibility level in terms of parallel computing platform (e.g., Nvidia CUDA platform & deep neural network library CuDNN)	[22]; [25]; [78]; [125]; [83]**;** [144]; [232]; [167]; [112]; [84]; [93]; [117]; [80]; [36]; [168]; [176]; [151]; [177]; [26]; [82]; [100]; [118]; [24]; [64]; [121]; [202]; [111]; [89]; [113]; [106]; [146]; [37]; [35]; [98]; [59]; [38]; [110]; [42]; [34]; [77]; [70]; [105]; [97]; [71]; [60]; [54]; [5]; [2]; [56]; [44]; [52]; [46]; [50]; [63].	54 Papers

Recently [303] have demonstrated that there are emerging requirements for the interoperability between previously mentioned ML and DL frameworks, such that available model files and training/serving programs implemented with one framework could be easily ported and reused with another framework. This is because using multiple frameworks is becoming a standard practice for achieving the best development results, including Model learning performance and programming Skills and DevOps productivity. Existing BiMDLs framework, on the other hand, prioritize performance and expressivity over composability and portability, making framework compatibility difficult.Tables21providesfurtherdetails and explanation of the compatibility level in terms of open sources, interfaces and libraries.

TABLE 21: ***LoC*** (Lack of compatibility within big machine deep learning systems (Lacking in terms of open sources, interfaces and libraries)

Factors ID	Theory	Open issues Dimensions	References	Total of Ref.
LoC (Lack of compatibility within big machine deep learning systems BiMDLs (Lacking in	Computational complexity theory	Mismatch Level among ML and DL frameworks, systems, Techniques, and libraries. (e.g.Despite the fact that only PyTorch is directly supported, the sufficiency of PyTorch-to-ONNX model conversion (in spite its constraints in functionality) allows for model compatibility	[79]; [22]; [25]; [78]; [121]; [83]**;** [144]; [40]; [84]; [80]; [36]; [186]; [176]; [151]; [145]; [75]; [26]; [82]; [100]; [118]; [64]; [131];[89]; [113]; [106]; [37]; [62]; [87]; [35]; [81]; [103]; [59]; [38]; [43]; [41]; [42]; [45]; [51]; [61];	56 Papers

terms of open sources, interfaces and libraries)		between deep learning frameworks. [131]). We have not yet established default policies that are suitable for all users. [89].	[34]; [70]; [105]; [74]; [97]; [71]; [60]; [54]; [5]; [2]; [56]; [49]; [44]; [52]; [46]; [50]; [63]	

The CPU, GPU, and TPU differ in that the CPU handles all of the computer's logic, calculations, and input/output; it is a general-purpose processor. In contrast, a GPU is an additional processor used to improve the graphical interface and run high-end tasks. TPUs are powerful custom-built processors that are used to run projects built on a specific framework i.e., TensorFlow.

- CPU: Central Processing Unit. Manage all the functions of a computer.
- GPU: Graphical Processing Unit. Enhance the graphical performance of the computer.
- TPU: Tensor Processing Unit. Custom build ASIC to accelerate TensorFlow projects.

Therefore, the compatibility among big machine deep learning systems is varies based on the type of processor, as illustrated in Table 22.

TABLE 22:***LoC*** (Lack of compatibility) within big machine deep learning systems BiMDLs in terms of type of processor (CPUs, GPUs or TPUs)

Factors ID	Theory	Open issues Dimensions	References	Total of Ref.
LoC (Lack of compatibility within big machine deep learning systems BiMDLs in term of type of processor)	Computational complexity theory	Compatibility level in terms of CPUs, GPU and TPUs) and their performance/ Platform Compatibility	[79]; [22]; [25]; [78]; [83]**;** [144]; [214]; [84]; [93]; [80]; [36]; [168]; [176]; [151]; [86]; [158]; [177]; [157]; [39]; [26]; [82]; [100]; [118]; [24]; [105]; [64]; [121]; [111]; [89]; [113]; [106]; [146]; [140]; [37]; [35]; [98]; [81]; [103]; [59]; [48]; [38]; [110]; [58]; [41]; [42]; [34]; [72]; [77]; [70]; [40]; [71]; [60]; [54]; [5]; [2]; [56]; [49]; [44]; [52]; [46]; [50]; [63].	62 Papers

Undoubtedly, the BiMDLs framework has met with tremendous success in a broad range of fields, including computer vision/artificial vision, voice recognition, natural language processing, immersive comprehension, and the latest state-of-the-art ML DL implementations.

Moreover, almost anywhere and the success of these technologies is attributed by the use of various layers of artificial neurons to its high representational capacity of input data and big data processing such as texts, Audios or videos). using various methods, diverse algorithms, and several neural networks to deal with these types of data. For instance:

*ML algorithms:*SVM (Support Vector Machine), LR (Linear Regression), NB (Naïve Bayes), LR (Logistic Regression), K-MEAN, K-Nearest Neighbors, RF (Random Forest), DT Decision Tree.

DL algorithms: ANN, DNNs (e.g., ImageNet, AlexNet, MNIST and CIFAR), CNNs (Its Networks: e.g., AlexNet, VGGNet, RestNet, Inception, and GoogleNet), RNNs, LSTMs, MLPNN, DBM, DBN.

Therefore, to select the proper open-source for a particular mission, it must be compatible with the type of algorithm, type of data/dataset, and the type of neural network. Table 23 illustrates the issue dimension associated with its factor and the related theory supported by the related literature references.

TABLE 23:***LoC*** (Lack of compatibility) within big machine deep learning systems BiMDLs in terms of type of data inputs, algorithm and network

Factors ID	Theory	Open issue Dimensions	References	Total of

				Ref.
LoC (Lack of compatibility within big machine deep learning systems BiMDLs In terms of type of : -data inputs. -Algorithms -Networks	Computational complexity theory	-Compatibility level in different types of inputs (Text, Audio or Video) and types of ML/DL networks (e.g., SVM, LR, NB, K-MEAN, RF, ANN; DNNs, RNNs, CNNs, GANs, LSTMs, GRUs, RCNN, DBNs and RBMsin terms of the gap between object detection and image processing, on the other hand. - Inconsistent tensor layouts: "Some operators of the target framework may not support the original input tensor layout (NCHW vs. NHWC. Therefore, the target model generator must carefully transpose tensors at the proper places, modify such operators' learnable parameters/attributes, or even reimplement them to ensure faithfulness.	[79]; [22]; [25]; [78]; [121]; [125]; [83]**;** [144]; [86]; [84]; [93]; [80]; [36]; [151]; [224]; [177]; [147]; [139]; [157]; [92]; [39]; [75]; [26]; [82]; [100]; [114]; [118]; [24]; [105]; [64]; [202]; [111]; [89]; [113]; [146]; [140]; [37]; [55]; [107]; [94]; [62]; [35]; [76]; [81]; [103]; [59]; [53]; [120]; [48]; [57]; [38]; [43]; [110]; [58]; [42]; [61]; [34]; [77]; [70]; [105]; [40]; [74]; [97]; [71]; [60]; [54]; [5]; [2]; [56]; [49]; [44]; [46]; [50]; [63]; [303]	74 Papers

Second issue: The difficulty of code generation and conversion

The conversion of generated code is technically feasible because most of the current ML and DL frameworks take the same abstraction to represent models as similar tensor-oriented computation graphs whose syntax and semantics are well defined [303]. However, it is not easy due to the following major challenges:Various machine learning and deep learning algorithms help to improve learning performance, broaden application scopes, training, computing, performance accuracy and simplify the calculation process and the exceptionally long training cycle of the deep learning models remains a big problem for researchers [4]. Table 24 illustrates more details about code generation and conversion in terms of training process, computing and accuracy.

TABLE 24: Code generation and conversion in terms of training process, computing and accuracy.

Factors ID	Theory	Open issues Dimensions	References	Total of Ref.
LCG (Lack of code Generation &LCC (Lack of code conversion) in Big machine deep learning systems BiMDLs) in terms of Accuracy)	Computational complexity theory	Training process, computing and Performance accuracy	[79]; [22]; [83]**;** [40]**;** [117]; [36]; [137]; [166]; [177]; [169]; [67]; [99]; [39]; [75]; [114]; [131]; [127]; [102]; [25]; [113]; [47]; [106]; [37]; [66]; [91]; [132]; [107]; [94]; [90]; [96]; [62]; [69]; [35]; [76]; [81]; [103]; [59]; [53]; [120]; [48]; [57]; [38]; [43]; [110]; [58]; [41]; [42]; [45]; [73]; [68]; [51]; [61]; [34]; [72]; [108]; [77]; [70]; [71]; [60]; [54]; [5]; [56]; [49]; [44]; [52]; [46]; [50]; [63].	68 Papers

Time-consuming, latency and performance efficiency: as shown in Table 25: Literature reviews show that there is a significant increase in time-consuming and some latency occurs during the code generation and the code conversion process from one environment to another different environment; this affects overall performance efficiency.

TABLE 25:LCG(Lack of code Generation)& LCC(Lack of code conversion) inBiMDLsin terms of Time-Consuming

Factors ID	Theories	Open issues Dimensions	References	Total of Ref.
LCG (Lack of code Generation & LCC Lack of code conversion) in Big machine deep learning systems BiMDLs) in terms of Time Consuming)	Coding theory/Computational complexity theory	Time-consuming, latency and performance efficiency	[79]; [22]; [125]; [97]; [83]**;** [40]**;** [86];[117]; [36]; [168]; [176]; [216]; [166]; [147]; [67]; [99]; [39]; [75]; [26]; [114]; [105]; [64]; [131]; [121]; [127]; [111]; [102]; [25]; [113]; [106]; [37]; [132]; [107]; [94]; [90]; [62]; [35]; [76]; [98]; [59]; [48]; [57]; [38]; [58]; [41]; [42]; [45]; [73]; [68]; [51]; [34]; [156]; [72]; [108]; [70]; [74]; [71]; [54]; [5]; [56]; [49]; [44]; [52]; [46]; [50]; [63].	66 Papers

There are obvious discrepancies in big machine deep learning lifecyclein both the computation graph constructs and supported features between BiMDLs frameworks [303]. Additionally, the exceptionally long training cycle of the deep learning models remains a big problem for researchers [4]. Table 26Provides more details and a summary of the literature that indicates the training life-cycle of machine learning and deep learning models.

TABLE 26: Provides more details and a summary of the literature that indicates the training life-cycle of machine learning and deep learning models.

Factors ID	Theories	Open issues Dimensions	References	Total of Ref.
LCG (Lack of code Generation) & LCC (Lack of Code conversion) in Big machine deep learning systems) in terms of BiMDLs life-cycle)	Coding theory/Computational complexity theory	Training life-cycle of machine learning and deep learning models.	[79]; [22]; [125]; [97]; [83]**;** [40]**;** [36]; [203]; [186]; [166]; [147]; [67]; [99]; [39]; [75]; [82]; [191]; [100]; [114]; [118]; [102]; [25]; [47]; [146]; [37]; [85]; [55]; [107]; [90]; [96]; [62]; [69]; [87]; [35]; [76]; [81]; [59]; [53]; [120]; [48]; [57]; [38]; [58]; [39]; [42]; [45]; [88]; [73]; [68]; [51]; [61]; [34]; [77]; [70]; [74]; [71]; [60]; [54]; [5]; [49]; [44]; [52]; [50]; [63].	64 Papers

Code generation - Codebase accessibility- error-prone code and code quality: The source model syntax encoded in the file can be quite different and complicate to that expressed by developers in the original training program due to framework optimization or runtime execution model. This makes semantic comprehension much harder and results in conversion failure or a non-optimal target model. [303]. For instance, TensorFlow translates neural network layers to low-level optimized tensor operations. Another example is that the PyTorch model file stores a dynamic computation graph defined by a real run, which could lose the conditional or loop information in code. Hence, reverse syntax transformation to recover the original ML or DL constructs is needed for the source model. Table 27 Provides more details and a summary of the literature that indicates the lacking in code Generation itself and its quality)

TABLE 27: LCG (Lack of code Generation) &LCC (Lack of Code conversion) in terms of(the Lacking in code Generation itself and its quality)

Factors ID	Theories	Open issues Dimensions	References	Total of Ref.
LCG (Lack of code	Coding	Code	[79]; [22]; [97]; [125]; [137];[83]**;**	62

Generation) & LCC (Lack of Code conversion) in Big machine deep learning systems (BiMDLs) in terms of (the Lacking in code Generation itself and its quality)	theory/Computational complexity theory	generation - Codebase accessibility-error-prone code and code quality.	[40];[166]; [92]; [117]; [36]; [138]; [212]; [216]; [100]; [158]; [145]; [67]; [75]; [26]; [82]; [191]; [190]; [114]; [118]; [105]; [131]; [121]; [127]; [102]; [25]; [106]; [37]; [107]; [62]; [35]; [76]; [103]; [59]; [120]; [48]; [38]; [43]; [58]; [39]; [41]; [45]; [68]; [51]; [34]; [108]; [77]; [70]; [60]; [54]; [5]; [56]; [49]; [44]; [52]; [46]; [50].	Papers

Furthermore, based on table 28, and since several implementations are available, modifications to a current algorithm can not necessarily be applied on the same method as the original, such as when introducing a new object detection algorithm, benchmarking it against a standard algorithm in this area, Faster R-CNN, implemented in Caffe / caffe2 is normal [26]; [304]. Sometimes, a fair analogy is only necessary if the latest idea is also applied in Caffe / caffe2, but the simplicity of using a deep learning system is limited.

On the contrary, in the time of using a specific system, the head-to-head analysis contributes to output uncertainty: It is impossible to assess how much of it stems from algorithmic improvements vs. that from the actual application, and this lack of interoperability also impedes researchers from quickly evaluating proposed new models in different deep learning frameworks. They will either expend time building different systems or attempt to re-implement them within their system. [26].

TABLE 28: LCG (Lack of code Generation) &LCC (Lack of Code conversion) in terms of the implementation process

Factors ID	Theories	Open issues Dimensions	References	Total of Ref.
LCG (Lack of code Generation) & LCC (Lack of Code conversion) in Big machine deep learning systems BiMDLs) (Lack of code Generation & conversion in terms of implementation process) (Cost should be economical)	Coding theory /Computational complexity theory/Transaction cost economics (TCE)	Implementation Time, Implementation cost Benchmark test and the (parallel computing total cost)	[79]; [22]; [125]; [83]; [40]; [117]; [36]; [168]; [138]; [216]; [166]; [158]; [67]; [99]; [75]; [26]; [114]; [118]; [105]; [64]; [131]; [89]; [102]; [25]; [47]; [106]; [37]; [132]; [62]; [35]; [76]; [59]; [53]; [48]; [57]; [38]; [58]; [39]; [41]; [45]; [73]; [68]; [51]; [61]; [34]; [156]; [71]; [5]; [44]; [52]; [46]; [50]; [63], [212];[186]; [72]	56 Papers

Deep Learning (DL), a new class of machine learning algorithms that aims to learn multiple levels of representation and abstraction that aid in inferring knowledge from data sources such as multimedia data (e.g., text, image, audio, and video), has recently made astounding advances in machine vision, speech recognition, multimedia analysis, and drug design. While current tools such as Theano, Torch/PyTorch, Caffe/Caffe2, Computational Network Toolkit (CNTK), and TensorFlow are efficient in their respective domains, they are essentially application libraries with some inherent limitations. They are basically application libraries with some limitations. Like all programming libraries, the DL libraries have fixed bindings for key data structures such as tensors and tensor-related computations. Users must adhere to the data structure, limiting their ability to perform application-specific optimization or port it to target runtime platforms. The internal representation of their control flow logic is not visible to users. [138].

CNTK and TensorFlow, for example, represent DL network computation using directed acyclic graphs and generate runtime binaries from the graphs. However, because these graphs are not intended for user-level access, the runtime platforms of DL applications are restricted to what the libraries provide.Generally, current libraries must be built for the platforms for which they are created, which can be difficult on Windows platforms. Furthermore, changing the implementation of specific layers or data structures is difficult without a thorough understanding of the underlying implementation. This reduces their reusability and portability. Therefore, to address these limitations we present BiMDLs model to provide (1) intuitive constructs to support deep network compact encoding; (2) symbolic gradient derivation of the networks; and (3) static analysis for memory consumption and error detection.

For a better understanding of the difficulty of code generation and conversionproblem, we can provide this example: Google developed and supports TensorFlow whereas PyTorch developed by Facebook AI Research lab (FAIR), Let's assume that there is a software problem in Google (TensorFlow) related to question answering, and the researcher who wants to handle this problem belongs to FAIR which owns (PyTorch), then, certainly, in this case, the researcher have to convert his entire code from PyTorch to TensorFlow just to handle one single problem as the two codes are different, this will lead to consuming a lot of time, more researcher suffering during coding and more cost.

Furthermore, while Python is the most widely used programming language in data science and parallel computing, C++, Java, and Julia are also used in some cases.

However, another limitation regarding code conversion is that the BiMDLs frameworks and their libraries do not generate Java source code, this leads to incurring lengthy start-up time That will replicate the same pre-processing steps every time a BiMDLs system is begun and the Java software created also has minimal dependencies and can operate on all major operating systems like Windows, macOS, and Linux without any changes or rebuild. The lack of these two advantages makes DL frameworks less portable than other techniques [137].

Furthermore, since it is not compilation-based BiMDLs frameworks and their libraries statically and analyzing memory consumption for determining deep Learning network whether run on a computer platform without actually deploying the code and the Java program generated is easy to debug with an IDE such as Eclipse or IntelliJ, where users can set breakpoints and inspect intermediate results, which is very difficult for most DL libraries which we already know [137]. Table 29provides further details in terms of BiMDLs complexity aspects.

TABLE 29: LCG (Lack of code Generation) &LCC (Lack of Code conversion) in terms of Tensor dimensions, Language type,coding, encoding, decoding andEncryption complexity

Factors ID	Theories	Open issues Dimensions	References	Total of Ref.
LCG (Lack of code Generation) & LCC (Lack of Code conversion) in Big machine deep learning systems BiMDLs) in terms of the Lack of code conversion.	Coding theory/Computational complexity theory	Code conversion, Code duplication and Code Importing/Code Exporting	[79]; [125]; [83]**;** [40]**;** [166]; [92]; [176]; [216]; [158]; [67]; [75]; [26]; [191]; [100]; [118]; [24]; [105]; [131]; [89]; [102]; [37]; [55]; [35]; [98]; [103]; [59]; [120]; [48]; [38]; [58]; [39]; [68]; [51]; [34]; [77]; [70]; [5]; [56]; [44]; [52].	40 Papers
LCG (Lack of code Generation) & LCC (Lack of Code conversion) in Big machine deep learning systems BiMDLs) in terms of the Lack of Encoding.	Coding theory/Computational complexity theory	Encoding /Decoding complexity	[79]; [125];[137];[83]**;** [40]**;** [166]; [92]; [126]; [167]; [117]; [145]; [99]; [75]; [191]; [190]; [114]; [131]; [122]; [102]; [113]; [37]; [55]; [62]; [35]; [76]; [81]; [53]; [120]; [48]; [38]; [58]; [39]; [41]; [45]; [68]; [51]; [34]; [77]; [74]; [5].	40 Papers
LCG (Lack of code Generation) & LCC (Lack of Code conversion) in Big machine deep learning systems BiMDLs in terms of language type and language mismatch).	Coding theory/Computational complexity theory	Type of programming language (Difficulty of JAVA code generation)	[79]; [22]; [125]; [83]**;** [138]; [137]; [212]; [39]; [75]; [82]; [191]; [114]; [131]; [102]; [25]; [35]; [59]; [38]; [41]; [42]; [45]; [34]; [54]	23 Papers
LCG (Lack of code Generation) & LCC (Lack	Coding theory/Co	Encryption	[79]; [22]; [125]; [97]; [83]**;** [40]**;** [166]; [92]; [36]; [75]; [191]; [122];	19 Papers

of Code conversion) in Big machine deep learning systems BiMDLs) in terms of the difficulty of encryption)	mputational complexity theory	complexity	[102]; [37]; [35]; [39]; [34]; [108]; [49]	

High computational complexity: This category identifies the deep learning algorithm complexity. While Deep neural networks (DNNs) can achieve state-of-the-art accuracy on many AI task levels of accuracy, it comes at the expense of a high computational complexity level [46].

However, flaws in deep neural networks' long training algorithm phase make them vulnerable to adversarial samples: adversaries' crafted inputs with the intent of causing deep neural networks to misclassify; this leads to makes it difficult to control the building of the code with the necessary flexibility [141].

Thus, this process becomes more complicated when converting the code to a different environment, especially when using a framework other than the one used to generate the original code, Table 30 provides further details in terms of algorithm complexity whether for ML or DL algorithm.

TABLE 30: LCG (Lack of code Generation) &LCC (Lack of Code conversion) in terms ofML and DLalgorithm complexity

Factors ID	Theories	Open issues Dimensions	References	Total of Ref.
LCG (Lack of code Generation) & LCC (Lack of Code conversion) in Big machine deep learning systems BiMDLs) in terms of ML &DL Algorithm Complexity)	Coding theory/Computational complexity theory	Type of deep learning algorithm. **And** Type of machine learning algorithm	[79]; [22]; [125]; [97]; [83]**;** [144]; [40]**;** [212]; [75]; [127]; [25]; [37]; [35]; [76]; [81]; [103]; [59]; [53]; [120]; [48]; [38]; [43]; [110]; [58]; [39]; [42]; [61]; [34]; [70]; [71]; [60]; [54]; [5]; [56]; [49]; [44]; [52]; [46]; [50]; [63]; [99]; [144]; [67]; [191]; [122]; [102];[47];[94]; [90]; [96]; [62]; [57]; [41]; [68]; [51].	55 Papers

Third issue:The Lack of benchmarks

In the software systems world, benchmarks [11] are known techniques that are usually used to link relationships between different frameworks and to tackle this problem, a series of benchmarks were developed, each of these benchmarks cantered on a common group of large data frameworks, and multiple benchmarking initiatives focused on analysing and contrasting multiple frameworks. Lack of benchmarks among BiMDLs is simply because of the lack of benchmarks among the frameworks themselves [86].The benchmark helps the researcher to compare performance within the framework as well as between-frameworks. [265].

The table 31illustrates various results of a comprehensive review in several factors and dimensions supported by the related theories. BiMDLs models have recently arisen and become increasingly common, and several research efforts have begun to suggest benchmarking structures in this area. However, deep Learning frameworks and their related systems are in high demand for benchmarking [86] ; [11], [71], [168]; [219]. That being said, it is substantially more complex and impossible to successfully benchmark machine / deep learning tech applications and systems than conventional performance-driven benchmarks and this is since deep learning systems are driven by large data are fundamental [219]; [86];[305]. That being said, it is substantially more complex and impossible to successfully benchmark machine / deep learning tech applications and systems than conventional performance-driven benchmarks and this is since deep learning systems are driven by large data are fundamentally both computation-intensive and data-intensive, needing intelligent convergence of massive data parallelism and major computation parallelism at all levels of deep learning architecture.

Therefore, this paper rethinks the problems of BiMDLs benchmarking software frameworks where many machine learning and several deep learning benchmarks have been raised [133], [168], [306], [307].From their in-depth studies, they outlined three observations: (First) the machine deep learning systems are designed for their default configuration settings, and for other datasets, the default configuration optimized on one particular dataset may not work effectively. (Second) When another DL system is used to practice on the same dataset, the default setup configured for one frame to practice on a data set may not fit well. (Third) Different DL frameworks display different levels of robustness against opponent examples and so it is [141].

TABLE 31: ***LoB***(Lack of benchmarks within Big machine deep learning systems BiMDLs)

In terms of input data, dataset type, Data parallelism,data capacity, data points and dimensions

Factors ID	Theory	Open issues Dimensions	References	Total of Ref.
LoB (Lack of benchmarks within Big machine deep learning systems BiMDLs) In terms of input data and dataset type	Complexity theory	Dataset's type Big data type	[217]; [26]; [60]; [11]; [133]; [54]; [5]; [141]; [134]; [148]; [86]; [149]; [142]; [143]; [101]; [40]; [117]; [36]; [169]; [139]; [67]; [92]; [39]; [75]; [105]; [61]; [102]; [25]; [113]; [47]; [106]; [37]; [66]; [132]; [107]; [94]; [69]; [87]; [35]; [76]; [98]; [103]; [53]; [120]; [38]; [43]; [58]; [45]; [73]; [51]; [34]; [72]; [108]; [77]; [78]; [70]; [97]; [71]; [2]; [56]; [49]; [44]; [52]; [46]; [50]; [63].	66 Papers
Factors ID	Theory	Open issues Dimensions	References	Total of Ref.
LoB (Lack of benchmarks within Big machine deep learning systems BiMDLs) In terms of Data parallelism.	Complexity theory	Data parallelism vs major computation parallelism intelligence convergence needed at all levels of ML & DL architecture "***Processing time*** and ***convergence rate*** are two main factors that users concern when training a deep learning model'.	[26]; [60]; [106]; [5]; [11]; [133]; [54]; [141]; [134]; [148]; [86]; [149]; [142]; [143]; [101]; [40]; [126]; [84]; [93]; [36]; [168]; [176]; [8]; [169]; [67]; [92]; [99]; [39]; [75]; [82]; [178]; [114]; [118]; [105]; [64]; [111]; [25]; [47]; [37]; [87]; [35]; [76]; [81]; [59]; [53]; [38]; [42]; [51]; [61]; [34]; [72]; [77]; [2]; [56]; [49]; [44]; [52]; [50]; [63]	59 Papers
Factors ID	Theory	Open issues Dimensions	References	Total of Ref.
LoB (Lack of benchmark within Big machine deep learning systems BiMDLs) In terms of data points and dimensions	Complexity theory	Data points and dimensions that used to evaluate systems.	[26]; [60]; [11]; [133]; [54]; [5]; [141]; [134]; [148]; [86]; [149]; [142]; [143]; [101]; [40]; [93]; [70]; [36]; [145]; [92]; [99]; [39]; [178]; [100]; [114]; [121]; [89]; [25]; [126]; [47]; [37]; [66]; [85]; [55]; [107]; [94]; [35]; [76]; [98]; [59]; [53]; [120]; [38]; [43]; [110]; [58]; [51]; [34]; [77]; [97]; [71]; [56]; [49]; [44]; [52]; [46]; [50]; [63].	58 Papers
Factors ID	Theories	Open issues Dimensions	References	Total of Ref.
LoB (Lack of benchmark within Big machine deep	Complexity theory/	Dataset's size	[217]; [26]; [106]; [11]; [133]; [5]; [141]; [134]; [148]; [86]; [149]; [142]; [143]; [40]; [36]; [169]; [139]; [67]; [92]; [39]; [64]; [61]; [89]; [47]; [146];	55 Papers

learning systems) (Lack of BiMDLs benchmarking in terms of the data capacity.			[37]; [66]; [85]; [107]; [94]; [69]; [87]; [35]; [76]; [53]; [38]; [43]; [58]; [39]; [45]; [73]; [51]; [34]; [77]; [70]; [60]; [54]; [2]; [56]; [49]; [44]; [52]; [46]; [50]; [63]	

As presented in table 32, In the benchmark process, Precision is about how close measure values are to each other, basically how many decimal places are at the end of a given measurement, so Precision does matter. Accuracy is how close a measured value is to the actual value. So, accuracy does matter too, but it is best when measurements are both precise and accurate.

Therefore, due to the lack of compatibility among BiMDLs interfaces and libraries, there is, in many cases, a lack of precision and accuracy measurements, and the results are divergent in most benchmark tests. However, failure to understand the tension between Precision and accuracy can have profound adverse effects on how one processes data, and the final outcome of parallel computing for big data Figure (5) shows the relationship between Precision and accuracy.

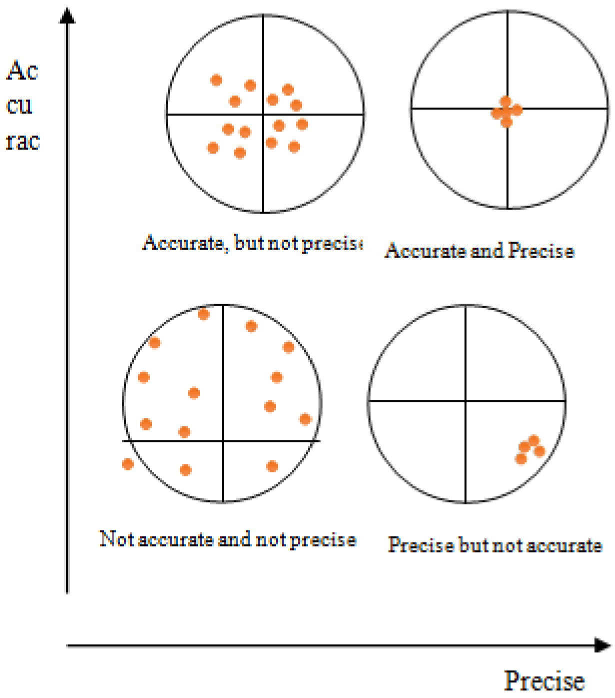


Figure 5: BiMDLs Benchmarking Accuracy and Precision

TABLE 32:***LoB*** (Lack of benchmarks within Big machine deep learning systems BiMDLs) in terms ofAccuracy and Precision

Factors ID	Theory	Open issues Dimensions	References	Total of Ref.
LoB (Lack of benchmarks within Big machine deep learning systems	Complexity theory	Precision vs. Accuracy	[22]; [239]; [11]; [54]; [266]; [218]; [213]; [26]; [60]; [133]; [5]; [141]; [134]; [148]; [86]; [149]; [142]; [143]; [101]**;** [40]**;** [192]; [187]; [70]; [36]; [177]; [92]; [82]; [61]; [102]; [25]; [47]; [106]; [37]; [66]; [91]; [85]; [107]; [94]; [96]; [35]; [76]; [59]; [53]; [48]; [43]; [58]; [39]; [42]; [73]; [51]; [61];	60 Papers

BiMDLs) In terms of outcome precision			[34]; [77]; [56]; [49]; [44]; [52]; [46]; [50]; [63].	

Moreover, the lack of benchmarks among the frameworks/techniques themselves: Refer to table 33, the seven vital characteristics of BiMDLs benchmarks are: 1) Relevance: Benchmarks should measure essential features. 2) Representativeness: Benchmark performance metrics should be broadly accepted by industry and academia. 3) Equity: All systems should be pretty compared. 4) Repeatability: Benchmark results should be verifiable. 5) Cost-effectiveness: Benchmark tests should be economical (e.g., Energy cost). 6) Scalability: Benchmark tests should measure from a single server to multiple servers. 7) Transparency: Benchmark metrics should be readily understandable.

TABLE 33: ***LoB*** (Lack of benchmark within Big machine deep learning systems)

Factors ID	Theories	Open issues Dimensions	References	Total of Ref.
LoB (Lack of benchmark within Big machine deep learning systems) (Lack of BiMDLs benchmarking) in terms of benchmarkingvital characteristics	Complexity theory/ Transaction cost economics (TCE)	1) Irrelevance. 2) Unrepresentative 3) Inequity/Imbalance 4) Unrepeatability 5) Uneconomical 6) lack of scalability 7) lacking in transparency/Opacity	[86]; [11]; [265]; [213]; [71]; [217]; [26]; [60]; [133]; [54]; [5]; [141]; [134]; [148]; [137]; [144]; [149]; [142]; [143]; [83]; [101]; [40]; [92]; [166]; [175]; [112]; [84]; [70]; [36]; [22]; [39]; [82]; [190]; [118]; [64]; [25]; [106]; [37]; [87]; [35]; [81]; [59]; [41]; [42]; [45]; [61]; [34]; [77]; [71]; [56]; [49]; [44]; [52]; [46]; [50]; [63].	56 Papers

While there are some current BiMDLs benchmarking activities, most of them concentrate on researching various CPU-GPU configurations and their effect on common datasets on specific DL frameworks. [86], [106], [225]. And also a few efforts and studies have addressed the performance of DNNs training holistically yet provides an in-depth look at layer-wise performance for different DNNs [225]. On the other hand, [281] conducted a review with distributed machine learning algorithms in which Apache Flink and Apache Spark implemented mathematically equivalent versions of these and it is derived from the literature review with up to 4 trillion data points with 100 million dimensions were used to evaluate systems.

The study contrasted the effects of both systems with single-node implementation success to place the scalability at the core of the analysis. While the results indicated that the systems are configurable, they are inefficient in the case of high dimensional data in question and so it is. In addition, there are a few other studies conducted in this regards, for instance; the DeepBench project [86] Focuses on simple benchmarking operations in neural networks such as dense matrix multiplications, convolutions, and coordination utilizing several tools and different neural network libraries; DAWNBench [106] is also a standard for end-to-end deep learning model training to reach goal precision utilizing deep learning tools like TensorFlow and PyTorch utilizing 3 CIFAR10, ImageNet and SQuAD datasets and all of them are important for DL method. [149]. Table 34 provides various aspects of lack of benchmarks within BiMDLs.

TABLE 34: Lack of benchmarks within BiMDLs in terms of DL algorithm, model, and ANN libraries

Factors ID	Theories	Open issues Dimensions	References	Total of Ref.
LoB (Lack of benchmark within	Complexity theory/	Benchmarking in ANN (e.g., in DeepBench project)	[26]; [60]; [106]; [11]; [133]; [54]; [5]; [141]; [134];	52 Papers

Big machine deep learning systems) (Lack of BiMDLs benchmarking) ANN and Deep learning algorithm.		And RNN, R-CNN, used, dense matrix multiplications, convolutions, and coordination).	[148]; [86]; [149]; [142]; [143]; [101]**;** [40]**;** [175]; [93]; [70]; [36]; [139]; [92]; [39]; [75]; [82]; [118]; [64]; [131]; [127]; [111]; [102]; [25]; [55]; [107]; [90]; [62]; [35]; [76]; [81]; [103]; [57]; [58]; [42]; [34]; [77]; [74]; [49]; [44]; [52]; [46]; [50]; [63].	
Factors ID	Theory	Open issues Dimensions	References	Total of Ref.
LoB (Lack of benchmark within Big machine deep learning systems) (Lack of BiMDLs benchmarking) in terms of End-to-end deep learning model and ANN libraries	Complexity theory/	End-to-end deep learning model and ANN libraries (e.g., DAWNBench) have demonstrated that there is a lack of standard evaluation criteria). As a result of the lack of standard evaluation criteria, a set of poorly understood trade-offs exists.	[26]; [60]; [106]; [11]; [133]; [54]; [5]; [141]; [134]; [148]; [86]; [149]; [142]; [143]; [101]**;** [40]**;** [70]; [36]; [169]; [92]; [99]; [100]; [94]; [90]; [35]; [76]; [120]; [43]; [110]; [58]; [39]; [42]; [34]; [77]; [56]; [49]; [44]; [52]; [46]; [50]; [63]	41 Papers

Table 35 provides several benchmarking issues aspects. Here few concerns of the performance of deep Learning systems and other is libraries under the assumption of the same datasets, methods, hardware: model performance (mean model accuracy) and run-time performance (mean training/inference speed) [22]. Considerable effort has been made by the DL community to benchmark and compares the runtime performance of various frameworks and libraries of interest and recent years have seen the introduction of a non-exhaustive set of benchmarks. The contrast of the most famous deep learning systems such as , TensorFlow, CNTK, PyTorch, Caffe2, Chainer, Theano, MXNet), with and without wrapper libraries such as Keras or Gluon and so those are [239]; [218], [266]; [240]; [219]; [305]. Also, Keras back-ends monitoring [22], [167], [217], [241]. Here, the comparison which has Java API related is still missing [137], [138], among others.DL frameworks with various methods and benchmarking are one of them [242] or LSTMs [18]".

Furthermore, variety of deep Learning frameworks such as configuration, multi-node code migration, accessibility, GitHub popularity, efficiency, memory usage, and scalability [176], [239]. Moreover comparing Caffe2 against PyTorch) and the result showed that it is tough in comparison with PyTorch [308].The consequences of the above benchmarks show comparable exactness for about all structures, while the exhibition of the runtime can here and there shift and then again, as usual, it is regularly hard to survey the amount of this distinction is because of the structures themselves and what amount is expected specifically to the best possible usage of the model [22]. In a specific circumstance, there shouldn't be a major contrast in runtime execution on a basic level since most structures utilize the equivalent basic cuDNN natives and it is critical to recall that there is no reasonable champ for all use cases because of the affectability of systems and issue settings to various decisions [219]. It is found very difficult to compare different methods in the current separated status. [309].

As matter of fact, big machine/deep Learning systems have nothing in common except certain parallels at a broad level and sole proposes its API, execution engine representation, and optimizations, in other meaning , there is a lack of compatibility among these systems [303]; [26]. This complexity also derives from the desire to address a particular range of issues and then attempt to generalize as an afterthought, hampering interoperability, implementation of strategies, and reusability [26]. These technologies are recent and still evolving. Besides, we may assume that benchmarking is known as an important method for testing database structures for improved comprehension [11], [310], respectively. Thus far, few work attempts have tried to address the benchmark machine or deep learning systems problem.

Furthermore, The absence of benchmarks for the machine or deep learning architectures, despite it is in high demand, brings us to another similar problem, and the selection of the proper M / D learning framework for the suitable applications is widely recognized as a daunting task for many researchers, developers and domain scientists [71], [311]. Moreover, since the ML and DL frameworks are increasingly present, space information isn't sufficient to manage complex issues and it presents a major test for information mining adventures in figuring out which assets to pick from the plenty of stages, databases, programming and arrangements from unique AI and profound learning client bunches in various pertinent fields [22].

TABLE 35: ***LoB***(Lack of benchmark within BiMDLs) in terms of configurations and technical process, APIs with AI, Runtime Performance comparison, algorithm complexity, structures and reality

Factors ID	Theories	Open issues Dimensions	References	Total of Ref.
LoB (Lack of benchmark within Big machine deep learning systems BiMDLs) in terms of configurations and technical process	Complexity theory/	Difficulty of benchmark in terms of configuration, multi-node code migration, accessibility, GitHub popularity, efficiency, memory usage, and scalability	[241]; [242]; [18]; [239]; [209]; [213]; [26]; [60]; [106]; [11]; [133]; [54]; [5]; [141]; [134]; [148]; [86]; [149]; [142]; [143]; [101]; [40]; [112]; [93]; [70]; [36]; [158]; [177]; [157]; [92]; [39]; [82]; [118]; [121]; [111]; [25]; [102]; [37]; [35]; [76]; [120]; [42]; [51]; [34]; [77]; [71]; [56]; [49]; [44]; [52]; [50]; [63]	52 Papers
Factors ID	Theory	Open issues Dimensions	References	Total of Ref.
LoB (Lack of benchmark within Big machine deep learning systems) (Lack of BiMDLs benchmarking in terms of APIs with Artificial intelligence AI systems	Complexity theory/	The comparison which has Java APIs related (AI frameworks with their various methods and libraries one of them) is still missing among others.	[217]; [167]; [267]; [241]; [242]; [18]; [60]; [106]; [11]; [133]; [54]; [5]; [141]; [134]; [148]; [86]; [149]; [142]; [143]; [101]; [40]; [84]; [36]; [137]; [145]; [92]; [75]; [105]; [64]; [122]; [89]; [102]; [25]; [47]; [37]; [85]; [87]; [35]; [59]; [38]; [110]; [41]; [42]; [45]; [51]; [34]; [77]; [71]; [56]; [44]; [52]; [46].	52 Papers
Factors ID	Theory	Open issues Dimensions	References	Total of Ref.
LoB (Lack of benchmark within Big machine deep learning systems) (Lack of BiMDLs benchmarking) in terms of Runtime Performance comparison.	Complexity theory/	Runtime performance comparison of the various AI Frameworks, Libraries, and systems	[26]; [11]; [133]; [54]; [5]; [141]; [134]; [148]; [86]; [149]; [142]; [143]; [101]; [40]; [36]; [168]; [158]; [147]; [92]; [39]; [75]; [100]; [105]; [64]; [89]; [102]; [25]; [106]; [37]; [94]; [87]; [35]; [38]; [110]; [41]; [42]; [61]; [34]; [77]; [70]; [74]; [97]; [71]; [60]; [2]; [49]; [44]; [52]; [46]; [50]; [63]	51 Papers
Factors ID	Theory	Open issues Dimensions	References	Total of Ref.
LoB (Lack of benchmark within Big machine deep learning systems BiMDLs) in terms of algorithm complexity	Complexity theory/	Type of ML and DL algorithm and the issue of complexity	[26]; [60]; [106]; [11]; [133]; [54]; [5]; [141]; [134]; [148]; [86]; [149]; [142]; [143]; [101]; [40]; [93]; [70]; [36]; [22]; [92]; [100]; [25]; [47]; [37]; [62]; [35]; [38]; [43]; [58]; [34]; [77]; [74]; [71]; [56]; [49]; [44]; [52]; [46]; [50]; [63]	41 Papers
Factors ID	Theory	Open issues Dimensions	References	Total of Ref.
LoB(Lack of benchmark within Big machine deep learning systems BiMDLs) in terms of Benchmarking structures	Complexity theory/	Benchmarking structures and Benchmarking procedures	[240]; [219]; [22]; [213]; [60]; [220]; [26]; [106]; [11]; [133]; [54]; [5]; [141]; [134]; [148]; [86]; [149]; [142]; [143]; [101]; [40]; [175]; [93]; [70]; [36]; [92]; [99]; [25]; [37]; [59]; [39]; [34]; [77]; [56]; [44]; [52]; [46]; [50]; [63]	39 Papers
Factors ID	Theory	Open issues Dimensions	References	Total

				of Ref.
LoB (Lack of benchmark within Big machine deep learning systems BiMDLs) in terms of reality	Complexity theory/	Absence in reality and practice, despite ML/DL benchmark is in High demand.	[26]; [60]; [11]; [133]; [54]; [5]; [141]; [134]; [148]; [86]; [149]; [142]; [143]; [101]; [40]; [36]; [22]; [39]; [37]; [35]; [42]; [51]; [34]; [77]; [56]; [63]	26 Papers

Fourth issue: Difficulty of selecting the proper framework

This issue is the vital problem in this study, and it comes as a logical consequence of the three open issues that have been discussed above. However, there is a need for more efforts to aid in select the appropriate BiMDLs framework todiscover the following deep learning hot open issues from evolution of the frameworks. This sub-section included several aspects shown in table 36:

TABLE 36: Difficulty of choosing the proper framework among BiMDLs Systems & libraries

Factor ID	**Related theory**	**Dimensions & open issues**	**References**	**Total of Ref.**
(DoS) Difficulties of selecting the right framework among BiMDLs in terms of complexity	Complexity theory	Complexity of existing BiMDLs algorithms	[112]; [22]; [74]; [26]; [170]; [160]; [124]; [161]; [162]; [163]; [61]; [159]; [86]; [164]; [172]; [64]; [138]; [173]; [101]; [40]; [175]; [126]; [84]; [93]; [80]; [36]; [29]; [147]; [145]; [67]; [99]; [79]; [75]; [178]; [100]; [114]; [118]; [64]; [211]; [111]; [89]; [25]; [47]; [37]; [85]; [107]; [90]; [96]; [35]; [103]; [59]; [53]; [57]; [38]; [43]; [58]; [39]; [41]; [42]; [45]; [88]; [51]; [34]; [72]; [77]; [97]; [71]; [5]; [49]; [44]; [52]; [46]; [50]; [63]; [210]	75 P
(DoS) Difficulties of selecting the right framework among BiMDLs in terms of concept and structure understanding	Complexity theory	The difficulty of understanding data structure and algorithm: Since most algorithms and some data can be challenging to comprehend for those who do not have a solid background in BiMDLs or the distributed systems, finding the correct parameters can be a big challenge.	[112]; [22]; [74]; [61]; [26]; [170]; [171]; [160]; [124]; [161]; [162]; [163]; [159]; [86]; [164]; [172]; [64]; [138]; [173]; [101]; [40]; [175]; [126]; [169[80]; [36]; [115]; [29]; [139]; [92]; [99]; [75]; [11]; [25]; [47]; [119]; [37]; [85]; [55]; [90]; [35]; [76]; [59]; [57]; [38]; [43]; [110];]; [58]; [39]; [41]; [42]; [68]; [51]; [34]; [77]; [78]; [5]; [56]; [49]; [44]; [52]; [46]; [50]	63 P
(DoS) Difficulties of selecting the right framework among BiMDLsin terms of training model training	Complexity theory	Mismatching in terms of training model algorithms and components/ Model interoperability	[112]; [22]; [74]; [161]; [26]; [163]; [170]; [160]; [124]; [171]; [162]; [61]; [159]; [86]; [164]; [172]; [64]; [138]; [173]; [101]; [40]; [36]; [186]; [29]; [67]; [99]; [79]; [75]; [178]; [114]; [118]; [64]; [127]; [122]; [102]; [25]; [47]; [37]; [62]; [87]; [35]; [81]; [103]; [57]; [38]; [43]; [110]; [42]; [45]; [34]; [77]; [70]; [71]; [60]; [54]; [5]; [56]; [44]; [52]; [46]; [50]; [63].	62 P
(DoS) Difficulties of selecting the right framework among BiMDLs	Complexity theory	Mismatching in terms of features and attributes	[112]; [22]; [74]; [26]; [170]; [161]; [160]; [124]; [163]; [171]; [162]; [61]; [159]; [86]; [164]; [172]; [64]; [138]; [173]; [101]; [40]; [126]; [80]; [36]; [186]; [29]; [39]; [178]; [61]; [47]; [37];	60 P

			[132]; [55]; [107]; [62]; [35]; [81]; [103]; [53]; [48]; [38]; [43]; [42]; [45]; [88]; [68]; [34]; [72]; [70]; [71]; [60]; [5]; [2]; [56]; [49]; [44]; [52]; [46]; [50]; [63].	
(DoS) Difficulties of selecting the right framework among BiMDLs in terms of features	Complexity theory	Existing BiMDLs systems often offer little or no help on how to set and reduce the parameters. Furthermore, the lack of a standardized benchmarking methodology exacerbates the problem.	[112]; [22]; [161]; [163]; [74]; [26]; [171]; [170]; [61]; [160]; [124]; [162]; [159]; [86]; [164]; [172]; [64]; [138]; [173]; [101]; [40]; [126]; [80]; [36]; [79]; [25]; [126]; [47]; [37]; [66]; [55]; [90]; [35]; [98]; [53]; [48]; [57]; [38]; [43]; [110]; [58]; [42]; [45]; [68]; [34]; [72]; [71]; [60]; [54]; [5]; [56]; [49]; [44]; [46]; [50]; [63]	56 P
(DoS) Difficulties of selecting the right framework among BiMDLs in terms of Interfaces, tools and libraries	Complexity theory	Mismatching among BiMDLs Interfaces, tools and libraries	[112]; [22]; [159]; [26]; [74]; [170]; [160]; [124]; [171]; [161]; [162]; [163]; [61]; [86]; [164]; [172]; [64]; [138]; [173]; [101]; [40]; [166]; [175]; [77]; [167]; [80]; [36]; [186]; [151]; [158]; [157]; [190]; [114]; [211]; [25]; [37]; [62]; [35]; [38]; [43]; [41]; [42]; [45]; [61]; [34]; [70]; [71]; [5]; [56]; [44]; [52]; [46]; [50]; [63]; [209]	55 P
(DoS) Difficulties of selecting the right framework among BiMDLs Framework in terms of goal	Complexity theory	Differences/Similarities in terms of goal and nature of task	[112]; [22]; [74]; [26]; [170]; [161]; [160]; [124]; [171]; [163]; [162]; [61]; [159]; [86]; [164]; [172]; [64]; [138]; [173]; [101]; [40]; [84]; [80]; [67]; [178]; [64]; [25]; [37]; [85]; [55]; [94]; [62]; [35]; [76]; [81]; [59]; [38]; [43]; [110]; [58]; [41]; [42]; [88]; [34]; [97]; [71]; [54]; [5]; [56]; [44]; [52]; [46]; [50]; [63].	54 P
(DoS) Difficulties of selecting the right framework among BiMDLs	Complexity theory	Differences/Similarities in terms of function	[112]; [124]; [22]; [74]; [26]; [170]; [163]; [160]; [161]; [171]; [162]; [61]; [159]; [86]; [164]; [172]; [64]; [138]; [173]; [101]; [40]; [80]; [36]; [23]; [67]; [39]; [25]; [113]; [47]; [55]; [35]; [59]; [38]; [43]; [110]; [41]; [42]; [45]; [34]; [70]; [97]; [71]; [54]; [5]; [56]; [44]; [52]; [46]; [50]; [63].	50 P
(DoS) Difficulties of selecting the right framework among BiMDLs in terms of function	Complexity theory	Difference in terms of design (Model/Framework/algorithm) design	[112]; [171]; [22]; [74]; [26]; [170]; [160]; [163]; [124]; [161]; [162]; [61]; [159]; [86]; [164]; [172]; [64]; [138]; [173]; [101]; [40]; [175]; [93]; [23]; [151]; [158]; [89]; [25]; [37]; [85]; [62]; [35]; [110]; [41]; [42]; [34]; [54]; [5]; [56]; [44]; [52]; [46]; [50]; [63].	44 P
(DoS) Difficulties of selecting the right framework among BiMDLs in terms of purpose	Complexity theory	Difference in terms of purpose (e.g., TensorFlow, Keras, and Theano support almost the same tasks in ANLP or deep learning)	[112]; [124]; [22]; [161];[171]; [74]; [26]; [170]; [160]; [162]; [163]; [61]; [159]; [86]; [164]; [172]; [64]; [138]; [173]; [101]; [84]; [80]; [158]; [64]; [25]; [106]; [85]; [35]; [81]; [38]; [43]; [110]; [41]; [42]; [34]; [70]; [40]; [54]; [5]; [56]; [44]; [52]; [46]; [50].	44 P

RQ4: What are the empirical approaches that overcome the current challenges of BiMDls and reduce the high-cost, time-consuming parallel computing process?

A combination of various machine learning models can boost the parallel computing process's accuracy, model training efficiency, and high-quality computing outcomes. This could be achieved through a unified model where It will undoubtedly lead to improving the performance of the industry's machine deep learning techniques. However, this is important to note that to achieve the best accuracy, decrease time-consuming and reduce the computing cost, a combination of two or more of these methods is required [22]; [28]; [25]. We believe that this research's proposed model MDL-enabled is a promising technology to enhance BiMDLs frameworks' and their related libraries' compatibility Figure. One of our promising model functions is to address the above challenges and to make the conversion process easier, faster, more flexible, and more efficient. According to [22], It is crucial to stress that yet there is no particular method appropriate for each issue and that it is always necessary to mix it to achieve. We provide adequate support for CPU and GPU-based code generation. In practice, this allows the user to quickly examine their BiMDLs and their network design and verify results on a machine that lacks the Nvidia GPU.

Our proposed model will provide its structure for encoding a BiMDLs network. The majority of existing ML and DL libraries rely on external standard description files to define network structure and node connections. For example, the PROTOTXT-based configuration file format in Caffe/Caffe2 has been extensively used in some benchmarks. On the other hand, we prepare to add support for automatically converting/generating these files to/from the (our prototype Framework) constructs as input to the framework compiler. This should help significantly reduce the user's time for authoring the "Framework code" and comparing the "Framework code" with other "Frameworks codes". This will also improve the interoperability and compatibility of BiMDLs Frameworks, allowing a working network that runs with other frameworks to be quickly verified and executed in our Framework, and vice versa. Therefore, we summarize the most crucial contribution points based on our research objectives as following:

1) A list of SLR results and BiDMLs investigation leads to identifying the missing issues, limitations, standard features, advantages, disadvantages and the model formulation.
2) BiMDLs expert's feedback for the proposed empirical Model from the interviews leads to prototype designed.
3) List of AI, ML, DL, BD, and DS practitioner responses and beneficial findings gathered from literature, interviews, survey, Multi-criteria Analysis MCA, and Analytic Hierarchy Process (AHP) lead to final model design and framework formulation.
4) Effectiveness and Usefulness of the Evaluation model from the selected AI, ML, DL, BD, and DS environment lead to Evaluated, validated, verified, and useful proposed Model.
5) Our proposed Model will facilitate choosing the proper framework based on the type of data inputs.
6) Our proposed framework will aid in generating and convert a highly efficient code that is both users friendly and easily debuggable.

5. Conclusions

A comprehensive review of big machine deep learning systems was presented via a systematic literature review (SLR) in this paper. The review considered the critical challenges, open issues, and critical factors and dimensions that influence existing big deep learning systems and their related libraries. Furthermore, it guides researchers and developers to achieve successful BiMDLs by evaluating the factors and related dimensions that influence the parallel computing process. The contributions of this SLR, in response to the research questions raised, are described in detail. The research questions (RQ) asked were as follows;

RQ1: What are the most common characteristics, similarities, differences, attributes, advantages, and disadvantages among big machine deep learning systems (BiMDLs) in terms of their goals, tasks, and functions?

RQ2: What are the main open issues and challenges of current big machine deep learning systems (BiMDLs)?

RQ3: What critical factors and dimensions affect existing big machine deep learning systems (BiMDLs)?

RQ4: What are the empirical approaches that overcome the current challenges of BiMDLs and reduce the high-cost, time-consuming parallel computing process?

284 papers, from different research sources, were analysed to provided answers for each question, and map questions to each developed research objective. The systematic literature review's results demonstrated that researchers face significant challenges in terms of incompatibility among big machine deep learning system's software systems. We also presented and discussed other relevant open issues, such as the difficulty of code conversion, the lack of benchmarks, and the difficulty of selecting the proper framework among big machine deep learning frameworks. The literature of this study reveals that these issues affect parallel computing efficiency and effectiveness in terms of increased computing and development time, difficulty in organizing computing tasks, increased computing process costs, and decreased computing accuracy due to goal mismatches, which makes the process of computing, training, and performance extremely complicated. This study's obtained results contribute more knowledge to the existing literature about big machine deep learning systems. Furthermore, these results will be significant for artificial intelligence (AI) systems, machine learning (ML) methods, deep learning

(DL)frameworks, big data (BDA) analytics, and data scientists' (DS) researchers. The results will benefit entrepreneurs, information technology advisors, developers, industry, and national governments by acting as a reference for developing strategies and programs to improve the parallel computing process by enhancing the compatibility of the machine and deep learning libraries, interfaces, and big data open-source techniques.

6. Future Work

We are working on a practical solution model named MDL-enabled. We believe that this promising model will efficiently improve the compatibility between the BiMDLs libraries and systems. Furthermore, we believe that future work must include the following directions:

a) Data Pre-processing challenge from BiMDLs perspective**:** This part involves data Redundancy, data Heterogeneity, data Noise, data discretization, data Labelling, imbalanced data, and the representation and selection of features.

b) Learning phase challenges fromBiMDLs perspective: It will include data parallelism, non-parallelism, models/parameter parallelism, hybrid approaches.

c) Data veracity from BiMDLs perspective: Managing the data veracity is a significant direction in which algorithms able to access the dependability or credibility of data or data sources are being developed. Moreover, Unreliable data can be disclosed during data pre-processing, and another direction is to develop new ML models that can make inferences with unreliable or even contradictory data. On the other hand, the data Veracity's challenges: In terms of Reliability, Confidentiality, Quality, Interpretation, Uncertainty, Imprecision (lack of exactness or accuracy), and Relevance.

d) Evaluation challenges: 1) Themetrics focus on BiMDLs Prediction Accuracy in terms of Precision, error rate, recovery, squared error, Probability, posterior Probability, information gain, K-L divergence, Cost, optimization error, Profit margin, approximation,estimation error, mean and worst result. 2) Metrics focus on BiMDLs Scalability: In terms of Data, I / O, Performance, Fault tolerance, Real-time processing, Memory usage, Supported data size, Iterative task support, Performance. 3) Metrics focus on BiMDLs Usability: Interpretability, stability, Efficiency, Precision, and Robustness.

e) Ease of use: Complexity in establishing objective functions, Strength, Mean error, Diversity" of models.

7. Acknowledgement

The Authors confirm that this work contains have no conflict of interest. We are grateful to the university of technology Malaysia for the technical support, and to the Deputy Vice Chancellor" (Hal Ehwal Pelajar)" Prof. Dr: Shamsul Bin Sahibuddin, who has been our main inspiration and the source of encouragement and motivation during the whole of this work. We are also thankful to Dr. Eissa M. Alshari and Dr. Mohammed Kaity for the valuable discussions and suggestions.

References

1. Y. Duan, J. S. Edwards, and Y. K. Dwivedi, "Artificial intelligence for decision making in the era of Big Data – evolution, challenges and research agenda," Int. J. Inf. Manage., vol. 48, no. February, pp. 63–71, 2019, doi: 10.1016/j.ijinfomgt.2019.01.021.
2. B. Jan et al., "Deep learning in big data Analytics: A comparative study," J. Comput. Electr. Eng. Sci. Direct, vol. 75, no. September 2018, pp. 275–287, 2019, doi: 10.1016/j.compeleceng.2017.12.009.
3. Shatnawi, G. Al-Bdour, R. Al-Qurran, and M. Al-Ayyoub, "A comparative study of open source deep learning frameworks," 2018 9th Int. Conf. Inf. Commun. Syst. ICICS 2018, vol. 2018-Janua, no. April, pp. 72–77, 2018, doi: 10.1109/IACS.2018.8355444.
4. M. P. Reyes et al., "A Survey on Deep Learning," ACM Comput. Surv., vol. 51, no. 5, pp. 1–36, 2018, doi: 10.1145/3234150.
5. Y. Wu et al., "A Comparative Measurement Study of Deep Learning as a Service Framework," IEEE Trans. Serv. Comput., no. Dl, p. 15, 2019, doi: 10.1109/TSC.2019.2928551.
6. M. Hilbert, "Big Data for Development: A Review of Promises and Challenges," Dev. Policy Rev. peer-reviewed Acad. J. Impact factor1.093 2019 SJR, vol. 34, no. 1, pp. 135–174, 2016, doi: 10.1111/dpr.12142.
7. L. Ruthotto, S. J. Osher, W. Li, L. Nurbekyan, and S. Wu, "A machine learning framework for solving high-dimensional mean field game and mean field control problems," Proc. Natl. Acad. Sci. PNAS April 28, 2020 117 9183-9193; first Publ. April 9, 2020, vol. 117, no. 17, p. 11, 2020, doi: 10.1073/pnas.1922204117.

8. Y. E. Wang, C.-J. Wu, X. Wang, K. Hazelwood, and D. Brooks, "Exploiting Parallelism Opportunities with Deep Learning Frameworks," ACM Digtal Libr. ACM Trans. Arch. Code Optim., Vol. 18, No. 1, Artic. 9, Publ. date December 2020., vol. abs/1908.0, 2020, [Online]. Available: https://www.semanticscholar.org/paper/6abbc3d834a4341aec82228aac9fc0cf3fc0931c.
9. J. Li et al., "Feature selection: A data perspective," ACM Comput. Surv. J. ISIScience Cit. Index Expand., vol. 50, no. 6, 2017, doi: 10.1145/3136625.
10. P. Skryjomski, B. Krawczyk, and A. Cano, "Speeding up k-Nearest Neighbors classifier for large-scale multi-label learning on GPUs," Neurocomputing J. Publ. ELSEVIER , RADARWEG 29, AMSTERDAM, NETHERLANDS, 1043 NX; ISI Sci. Cit. Index Expand., vol. 354, 2019, doi: 10.1016/j.neucom.2018.06.095.
11. F. Bajaber, S. Sakr, O. Batarfi, A. Altalhi, and A. Barnawi, "Benchmarking big data systems: A survey," Comput. Commun., vol. 149, pp. 241–251, 2020, doi: 10.1016/j.comcom.2019.10.002.
12. Holzinger, "Introduction to MAchine Learning & Knowledge Extraction (MAKE)," J. Mach. Learn. Knowl. Extr. 2019, 1(1), 1-20; https//doi.org/10.3390/make1010001. Peer-Reviewed journal. MDPI, pp. 1–20, 2019, doi: 10.3390/make1010001.
13. X. Wei, B. Cao, and P. S. Yu, "Unsupervised feature selection on networks: A generative view," 30th AAAI Conf. Artif. Intell. AAAI 2016, no. 2016, Association for the Advancement of Artificial Intelligence, pp. 2215–2221, 2016.
14. X. Da and J. Grizzle, "Combining trajectory optimization , supervised machine learning , and model structure for mitigating the curse of dimensionality in the control of bipedal robots," SAJE Journals, Int. J. Robot. Res. Scopus Google Sch., vol. 9, p. 35, 2019, doi: 10.1177/0278364919859425.
15. Jason . Brownlee, "A Gentle Introduction to Tensors for Machine Learning with NumPy," pp. 1–17, 2018, [Online]. Available: https://machinelearningmastery.com/introduction-to-tensors-for-machine-learning/.
16. Facebook Open Source, "Caffe2 Open Source Brings Cross Platform Machine Learning Tools to Developers," Https://Caffe2.Ai/, pp. 3–6, 2017, [Online]. Available: https://caffe2.ai/blog/2017/04/18/caffe2-open-source-announcement.html.
17. Y. You, J. Demmel, K. Keutzer, Z. Zhang, and C. Hsieh, "Fast Deep Neural Network Training on Distributed Systems and Cloud TPUs," IEEE Trans. Parallel Distrib. Syst., vol. 30, no. 11, 2019, doi: 10.1109/TPDS.2019.2913833.
18. S. Braun, "LSTM Benchmarks for Deep Learning Frameworks," arXiv Prepr. arXiv1806.01818. 2018 Jun 5. Cornell Univ., vol. 1, p. 9, 2018, [Online]. Available: http://arxiv.org/abs/1806.01818.
19. M. Al-ayyoub and A. Shatnawi, "A DETAILED COMPARATIVE STUDY OF OPEN SOURCE DEEP LEARNING FRAMEWORKS," arXiv Prepr. arXiv1903.00102 Conell Univ. Google Sch. Semant. Sch., vol. 2, p. 25, 2020.
20. Shatnawi, G. Al-Bdour, R. Al-Qurran, and M. Al-Ayyoub, "A comparative study of open source deep learning frameworks," 2018 9th Int. Conf. Inf. Commun. Syst. ICICS 2018, vol. 2018-Janua, pp. 72–77, 2018, doi: 10.1109/IACS.2018.8355444.
21. Spark, "TensorFlowOnSpark," pp. 3–5, 2020.
22. G. Nguyen et al., "Machine Learning and Deep Learning frameworks and libraries for large-scale data mining: a survey," Artif. Intell. Rev. 5277–124 Springer, vol. 52, no. 1, pp. 77–124, 2019, doi: 10.1007/s10462-018-09679-z.
23. X. B. Huang, "DESIGN AND IMPLEMENTATION OF A DOMAIN SPECIFIC LANGUAGE FOR DEEP LEARNING," ProQuest 10792864 Publ. by ProQuest LLC (2018), no. May, p. 136, 2018.
24. J. D. Dignam, P. L. Martin, B. S. Shastry, and R. G. Roeder, "Eukaryotic gene transcription with purified components," Methods Enzymol. i In12th {USENIX} Symp. Oper. Syst. Des. Implement. ({OSDI} 16) 2016 (pp. 265-283)., vol. 101, no. C, pp. 582–598, 2016, doi: 10.1016/0076-6879(83)01039-3.
25. Z. Wang, K. Liu, J. Li, Y. Zhu, and Y. Zhang, "Various Frameworks and Libraries of Machine Learning and Deep Learning: A Survey," Arch. Comput. Methods Eng., no. 0123456789, p. 24, 2019, doi: 10.1007/s11831-018-09312-w.
26. L. Nguyen, P. Yu, and M. Chowdhury, "No!: Not Another Deep Learning Framework," Proc. Work. Hot Top. Oper. Syst. - HOTOS '17, Whistler, BC, Canada, May 08-10, 2017, https//doi.org/10.1145/3102980.3102995. ACM, vol. Part F1293, no. May 08-10, 2017, pp. 88-93 (6 Pages), 2017, doi: 10.1145/3102980.3102995.
27. Darwish, A. E. Hassanien, and S. Das, "A survey of swarm and evolutionary computing approaches for deep learning," Artif. Intell. Rev., vol. 53, no. 3, 2020, doi: 10.1007/s10462-019-09719-2.
28. R. . Shyamchari, "Image Recognition And Study Of Hyperparameter Optimization Of Convolutional Neural Networks Using Tensorflow And Keras Frameworks Deep," Phd Diss., 2018 Lyles Coll. Eng. Calif. State Univ. Fresno May 2018, No. May 2018, P. 400, 2018.

29. G. A. Lewis, S. Bellomo, and A. Galyardt, “Component Mismatches Are a Critical Bottleneck to Fielding AI-Enabled Systems in the Public Sector,” Carnegie Mellon Softw. Eng. Inst. Pittsburgh, PA USA arXiv1910.06136v1.AAAI, vol. 1, pp. 2–5, 2019.
30. J. R. Ragini, P. M. R. Anand, and V. Bhaskar, “Big data analytics for disaster response and recovery through sentiment analysis,” Int. J. Inf. Manage., vol. 42, no. May, pp. 13–24, 2018, doi: 10.1016/j.ijinfomgt.2018.05.004.
31. Kitchenham, “Guidelines for performing Systematic Literature Reviews in Software Engineering,” p. 65, 2007, doi: 10.1145/1134285.1134500.
32. S. Sanchez-Gordon and S. Luján-Mora, “Technological Innovations in Large-Scale Teaching: Five Roots of Massive Open Online Courses,” J. Educ. Comput. Res., vol. 56, no. 5, pp. 623–644, 2018, doi: 10.1177/0735633117727597.
33. F. Hujainah, R. B. A. Bakar, M. A. Abdulgabber, and K. Z. Zamli, “Software requirements prioritisation: a systematic literature review on significance, stakeholders, techniques and challenges,” IEEE Access, vol. 6, pp. 71497–71523, 2018.
34. S. Mittal and S. Vaishay, “A survey of techniques for optimizing deep learning on GPUs,” J. Syst. Archit. Publ. Elsevier , Radarweg 29, Amsterdam, Netherlands, 1043 NX, vol. 99, p. 45, 2019, doi: 10.1016/j.sysarc.2019.101635.
35. M. A. Ebrahimighahnavieh, S. Luo, and R. Chiong, “Deep learning to detect Alzheimer’s disease from neuroimaging: A systematic literature review,” Comput. Methods Programs Biomed., vol. 187, p. 105242, 2020, doi: 10.1016/j.cmpb.2019.105242.
36. Darwish, A. Ella, and H. Swagatam, “A survey of swarm and evolutionary computing approaches for deep learning,” Artif. Intell. Rev., 2019, doi: 10.1007/s10462-019-09719-2.
37. S. Fatima, K. C. Desouza, and G. S. Dawson, “National strategic artificial intelligence plans: A multi-dimensional analysis,” J. Econ. Anal. Policy; Publ. Elsevier , Radarweg 29, Amsterdam, Netherlands, 1043 NX, vol. 67, pp. 178–194, 2020, doi: 10.1016/j.eap.2020.07.008.
38. Q. Mao, F. Hu, and Q. Hao, “Deep learning for intelligent wireless networks: A comprehensive survey,” IEEE Commun. Surv. Tutorials, vol. 20, no. 4, pp. 2595–2621, 2018, doi: 10.1109/COMST.2018.2846401.
39. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez, “A survey on deep learning techniques for image and video semantic segmentation,” Appl. Soft Comput. Journal; Publ. Elsevier , Radarweg 29, Amsterdam, Netherlands, 1043 NX, vol. 70, pp. 41–65, 2018, doi: 10.1016/j.asoc.2018.05.018.
40. Lavecchia, “Deep learning in drug discovery: opportunities, challenges and future prospects,” Drug Discov. Today J. 1359-6446/ã 2019 Elsevier Ltd. All rights Reserv. SCI, vol. 24, no. 10, p. 16, 2019, doi: 10.1016/j.drudis.2019.07.006.
41. T. R. Rao, P. Mitra, R. Bhatt, and A. Goswami, “The big data system, components, tools, and technologies: a survey,” Knowl. Inf. Syst. ISI Sci. Cit. Index Expand., vol. 60, no. 3, pp. 1165–1245, 2019, doi: 10.1007/s10115-018-1248-0.
42. Y. Zahidi, Y. El Younoussi, and Y. Al-amrani, “A powerful comparison of deep learning frameworks for Arabic sentiment analysis,” Int. J. Electr. Comput. Eng., vol. 11, no. 1, pp. 745–752, 2021, doi: 10.11591/ijece.v11i1.pp745-752.
43. M. Salvi, U. R. Acharya, F. Molinari, and K. M. Meiburger, “The impact of pre- and post-image processing techniques on deep learning frameworks: A comprehensive review for digital pathology image analysis,” Comput. Biol. Med., vol. 128, p. 104129, 2021, doi: 10.1016/j.compbiomed.2020.104129.
44. Shatnawi, G. Al-Bdour, R. Al-Qurran, and M. Al-Ayyoub, “A comparative study of open source deep learning frameworks,” 2018 9th Int. Conf. Inf. Commun. Syst. ICICS 2018; IEEE, vol. 2018-Janua, no. April, pp. 72–77, 2018, doi: 10.1109/IACS.2018.8355444.
45. Oussous, F. Z. Benjelloun, A. Ait Lahcen, and S. Belfkih, “Big Data technologies: A survey,” J. King Saud Univ. - Comput. Inf. Sci. ISI Sci. Cit. Index Expand., vol. 30, no. 4, pp. 431–448, 2018, doi: 10.1016/j.jksuci.2017.06.001.
46. V. Sze, Y. Chen, T. Yang, and J. Emer, “Efficient Processing of Deep Neural Networks: A Tutorial and Survey,” Proc. IEEE, vol. 105, no. 12, 2017, doi: 10.1109/JPROC.2017.2761740.
47. N. Mahendran, P. M. D. R. Vincent, and K. Srinivasan, “Machine Learning Based Computational Gene Selection Models : A Survey , Performance Evaluation , Open Issues , and Future Research Directions,” Front. Genet. J. SCIE (web Sci., vol. 11, no. December, pp. 1–25, 2020, doi: 10.3389/fgene.2020.603808.
48. Q. Zhang, L. T. Yang, Z. Chen, and P. Li, “A survey on deep learning for big data,” Inf. Fusion, vol. 42, no. August 2017, pp. 146–157, 2018, doi: 10.1016/j.inffus.2017.10.006.

49. R. Vinayakumar, K. Soman, and P. Poornachandran, "Evaluation of recurrent neural network and its variants for intrusion detection system (IDS)," Int. J. Inf. Syst. Model. Des., vol. 8, no. 3, 2017, doi: 10.4018/IJISMD.2017070103.
50. Awan, K. Manian, C. Chu, H. Subramoni, and D. Panda, "Optimized large-message broadcast for deep learning workloads: MPI, MPI+NCCL, or NCCL2?," Parallel Comput., vol. 85, 2019, doi: 10.1016/j.parco.2019.03.005.
51. D. W. Bates, A. Auerbach, P. Schulam, A. Wright, and S. Saria, "Reporting and Implementing Interventions Involving Machine Learning and Artificial Intelligence," Ann. Intern. Med. Jour, vol. 172, no. 11, 2020, doi: 10.7326/M19-0872.
52. W. Dai and D. Berleant, "Benchmarking contemporary deep learning hardware and frameworks: A survey of qualitative metrics," Proc. - 2019 IEEE 1st Int. Conf. Cogn. Mach. Intell. CogMI 2019, pp. 148–155, 2019, doi: 10.1109/CogMI48466.2019.00029.
53. Y. Chen, Y. Tian, and M. He, "Monocular human pose estimation: A survey of deep learning-based methods," Comput. Vis. Image Underst., vol. 192, no. December 2019, p. 102897, 2020, doi: 10.1016/j.cviu.2019.102897.
54. X. Lu, H. Shi, R. Biswas, M. Javed, and D. Panda, "DLoBD: A Comprehensive Study of Deep Learning over Big Data Stacks on HPC Clusters," IEEE Trans. Multi-Scale Comput. Syst., vol. 4, no. 4, 2018, doi: 10.1109/TMSCS.2018.2845886.
55. U. Naseem, I. Razzak, and P. W. Eklund, "A survey of pre-processing techniques to improve short-text quality: a case study on hate speech detection on twitter," Multimed. Tools Appl. Journal; Publ. SPRINGER , VAN GODEWIJCKSTRAAT 30, DORDRECHT, NETHERLANDS, 3311 GZ, p. 28, 2020, doi: 10.1007/s11042-020-10082-6.
56. Jain, A. A. Awan, H. Subramoni, and D. K. Panda, "Scaling TensorFlow, PyTorch, and MXNet using MVAPICH2 for high-performance deep learning on Frontera," Proc. DLS 2019 Deep Learn. Supercomput. - Held conjunction with SC 2019 Int. Conf. High Perform. Comput. Networking, Storage Anal., 2019, doi: 10.1109/DLS49591.2019.00015.
57. T. Meng, X. Jing, Z. Yan, and W. Pedrycz, "A survey on machine learning for data fusion," Inf. Fusion, vol. 57, no. 2, pp. 115–129, 2020, doi: 10.1016/j.inffus.2019.12.001.
58. F. Zaki, A. E. Mohamed, and S. G. Sayed, "CtuNet : A Deep Learning-based Framework for Fast CTU Partitioning of H265 / HEVC Intra- coding," Ain Shams Eng. J., no. xxxx, pp. 1–8, 2021, doi: 10.1016/j.asej.2021.01.001.
59. M. Li et al., "The Deep Learning Compiler: A Comprehensive Survey," IEEE Trans. Parallel Distrib. Syst. Publ. IEEE Comput. SOC , 10662 LOS VAQUEROS CIRCLE, PO BOX 3014, LOS ALAMITOS, USA, CA, 90720-1314, vol. 32, no. 3, pp. 708–727, 2021, doi: 10.1109/TPDS.2020.3030548.
60. Y. Oyama, S. Matsuoka, T. Ben-Nun, and T. Hoefler, "Accelerating Deep Learning Frameworks with Micro-Batches," Proc. - IEEE Int. Conf. Clust. Comput. ICCC, vol. 2018-, no. 11, 2018, doi: 10.1109/CLUSTER.2018.00058.
61. T. Doleck, D. J. Lemay, R. B. Basnet, and P. Bazelais, "Predictive analytics in education: a comparison of deep learning frameworks," Educ. Inf. Technol., vol. 25, no. 3, pp. 1951–1963, 2020, doi: 10.1007/s10639-019-10068-4.
62. E. M. Dogo, N. I. Nwulu, B. Twala, and C. Aigbavboa, "A survey of machine learning methods applied to anomaly detection on drinking-water quality data," Urban Water J., vol. 16, no. 3, pp. 235–248, 2019, doi: 10.1080/1573062X.2019.1637002.
63. J. H. Tao et al., "BenchIP: Benchmarking Intelligence Processors," J. Comput. Sci. Technol., vol. 33, no. 1, 2018, doi: 10.1007/s11390-018-1805-8.
64. L. Hung, Y. Y. Lin, C. Y. Tang, C. Wang, and M. C. Chen, "Performance of convolution neural network based on multiple GPUs with different data communication models," Proc. - 2018 IEEE/ACIS 19th Int. Conf. Softw. Eng. Artif. Intell. Netw. Parallel/Distributed Comput. SNPD 2018, pp. 87–92, 2018, doi: 10.1109/SNPD.2018.8441056.
65. L. Zhou, S. Pan, J. Wang, and A. V Vasilakos, "Neurocomputing Machine learning on big data : Opportunities and challenges," ELSEVIER J. Sci. Direct, vol. 237, no. September 2016, pp. 350–361, 2017, doi: 10.1016/j.neucom.2017.01.026.
66. S. Mao, B. Wang, Y. Tang, and F. Qian, "Opportunities and Challenges of Artificial Intelligence for Green Manufacturing in the Process Industry," Eng. Journal; Publ. ELSEVIER , RADARWEG 29, AMSTERDAM, NETHERLANDS, 1043 NX, vol. 5, no. 6, pp. 995–1002, 2019, doi: 10.1016/j.eng.2019.08.013.
67. L. E. Lwakatare, A. Raj, I. Crnkovic, J. Bosch, and H. H. Olsson, "Large-scale machine learning systems in real-world industrial settings: A review of challenges and solutions," Inf. Softw. Technol. 127 106368 Contents ELSEVIER, vol. 127, no. June, p. 17, 2020, doi: 10.1016/j.infsof.2020.106368.

68. E. Breck, N. Polyzotis, S. Roy, S. E. Whang, and M. Zinkevich, "DATA VALIDATION FOR MACHINE LEARNING," Proc. 2 nd SysML Conf. Palo Alto, CA, USA, 2019, pp. 1--14, 2019.
69. Y. Roh, G. Heo, and S. E. Whang, "A survey on data collection for machine learning: A big data - AI integration perspective," IEEE Trans. Knowl. Data Eng. Publ. IEEE Comput. SOC , 10662 LOS VAQUEROS CIRCLE, PO BOX 3014, LOS ALAMITOS, USA, CA, 90720-1314, vol. 2, pp. 1–20, 2018, doi: 10.1109/tkde.2019.2946162.
70. G. Al-bdour, R. Al-qurran, M. Al-ayyoub, A. Shatnawi, and M. Al-ayyoub, "Benchmarking open source deep learning frameworks," Int. J. Electr. Comput. Eng., vol. 10, no. 5, pp. 5479–5486, 2020, doi: 10.11591/ijece.v10i5.pp5479-5486.
71. P. Xu, S. Shi, and X. Chu, "Performance Evaluation of Deep Learning Tools in Docker Containers," Proc. - 2017 3rd Int. Conf. Big Data Comput. Commun. BigCom 2017, pp. 395–403, 2017, doi: 10.1109/BIGCOM.2017.32.
72. M. Štufi and B. Baˇ, "Big Data Analytics and Processing Platform in Czech Republic Healthcare," Appl. Sci. J. . Appl. Sci. 2020, 10, 1705; doi10.3390/app10051705 www.mdpi.com/journal/applsci10, 1705. MDPI, no. Appl. Sci. 2020, 10, 1705; doi:10.3390/app10051705, p. 23, 2020.
73. F. Cappa, R. Oriani, E. Peruffo, and I. McCarthy, "Big Data for Creating and Capturing Value in the Digitalized Environment: Unpacking the Effects of Volume, Variety, and Veracity on Firm Performance*," J. Prod. Innov. Manag. Publ. WILEY , 111 RIVER ST, HOBOKEN, USA, NJ, 07030-5774 ISI Sci. Cit. Index Expand. | Soc. Sci. Cit. Index, vol. 0, no. 0, pp. 1–19, 2020, doi: 10.1111/jpim.12545.
74. Sheth, M. Gaur, U. Kursuncu, and R. Wickramarachchi, "Shades of Knowledge-Infused Learning for Enhancing Deep Learning," IEEE Internet Comput., vol. 23, no. 6, pp. 54–63, 2019, doi: 10.1109/MIC.2019.2960071.
75. E. Kusmenko, S. Nickels, S. Pavlitskaya, B. Rumpe, and T. Timmermanns, "Modeling and Training of Neural Processing Systems," Conf. Model Driven Eng. Lang. Syst. (MODELS'19), pp. 283--293, IEEE, Munich, Sep. 2019. www.se-rwth.de/publications/, p. 11, 2019.
76. L. F. Sánchez-Peralta, L. Bote-Curiel, A. Picón, F. M. Sánchez-Margallo, and J. B. Pagador, "Deep learning to find colorectal polyps in colonoscopy: A systematic literature review," Artif. Intell. Med. Publ. ELSEVIER , RADARWEG 29, AMSTERDAM, NETHERLANDS, 1043 NX, vol. 108, no. March, p. 101923, 2020, doi: 10.1016/j.artmed.2020.101923.
77. S. Bianco, R. Cadène, L. Celona, and P. Napoletano, "Benchmark Analysis of Representative Deep Neural Network Architectures," IEEE Access IEEE Trans. JOURNALS, vol. 6, pp. 64270–64277, 2018, doi: 10.1109/ACCESS.2018.2877890.
78. Vinothini and S. B. Priya, "Survey of machine learning methods for big data applications," ICCIDS 2017 - Int. Conf. Comput. Intell. Data Sci. Proc., vol. 2018-Janua, pp. 1–5, 2018, doi: 10.1109/ICCIDS.2017.8272638.
79. Y. Liu et al., "Enhancing the interoperability between deep learning frameworks by model conversion," Proc. 28th ACM Jt. Meet. Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng., p. 10, 2020, doi: 10.1145/3368089.3417051.
80. W. G. Hatcher and W. Yu, "A Survey of Deep Learning: Platforms, Applications and Emerging Research Trends," IEEE Access 6 24411-24432, vol. 6, pp. 24411–24432, Apr. 2018, doi: 10.1109/ACCESS.2018.2830661.
81. S. Sengupta et al., "A review of deep learning with special emphasis on architectures, applications and recent trends," Knowledge-Based Syst., vol. 194, p. 105596, 2020, doi: 10.1016/j.knosys.2020.105596.
82. W. Xiang, "Applications Of Deep Learning In Large-Scale Object Detection And Semantic Segmentation," Proquest Number:13855813, No. March, P. 142, 2018.
83. R. M. Al-shamasneh and U. H. B. Obaidellah, "Artificial Intelligence Techniques for Cancer Detection and Classification: Review Study," Eur. Sci. J. January 2017 Ed. vol.13, No.3 ISSN 1857 – 7881 e - ISSN 1857- 7431, vol. 13, no. 3, pp. 342–370, 2017, doi: 10.19044/esj.2016.v13n3p342.
84. S. Amiri, "A Survey of Scalable Deep Learning Frameworks," J. IEEE Comput. Soc. IEEE Digit. Libr., pp. 2019–2020, 2019, doi: 10.1109/eScience.2019.00102.
85. P. Casares, "The Brain Of The Future And The Viability Of Democratic Governance: The Role Of Artificial Intelligence, Cognitive Machines, And Viable Systems," J. Futur. Publ. Elsevier Sci Ltd , Boulevard, Langford Lane, Kidlington, Oxford, England, Oxon, Ox5 1gb, Vol. 103, No. April, Pp. 5–16, 2018, Doi: 10.1016/J.Futures.2018.05.002.
86. S. Shams, R. Platania, K. Lee, and S. J. Park, "Evaluation of Deep Learning Frameworks over Different HPC Architectures," Proc. - Int. Conf. Distrib. Comput. Syst., pp. 1389–1396, 2017, doi: 10.1109/ICDCS.2017.259.

87. R. Sharma, S. S. Kamble, A. Gunasekaran, V. Kumar, and A. Kumar, "A systematic literature review on machine learning applications for sustainable agriculture supply chain performance," Comput. Oper. Res. J., vol. 119, p. 104926, 2020, doi: 10.1016/j.cor.2020.104926.
88. N. Elgendy and A. Elragal, "Big Data Analytics in Support of the Decision Making Process," Procedia Comput. Sci. Publ. ELSEVIER Sci., vol. 100, pp. 1071–1084, 2016, doi: 10.1016/j.procs.2016.09.251.
89. S. Ghemawat et al., "TensorFlow: A system for large-scale machine learning," Proc. 12th USENIX Symp. Oper. Syst. Des. Implementation, OSDI 2016, p. 21, 2016.
90. K. Kourou, T. P. Exarchos, K. P. Exarchos, M. V. Karamouzis, and D. I. Fotiadis, "Machine learning applications in cancer prognosis and prediction," Comput. Struct. Biotechnol. J., vol. 13, pp. 8–17, 2015, doi: 10.1016/j.csbj.2014.11.005.
91. Kaplan and M. Haenlein, "Rulers of the world, unite! The challenges and opportunities of artificial intelligence," J. Bus. Horizons; Publ. ELSEVIER , RADARWEG 29, AMSTERDAM, NETHERLANDS, 1043 NX, vol. 63, no. 1, pp. 37–50, 2019, doi: 10.1016/j.bushor.2019.09.003.
92. P. Mattson et al., "MLPerf: An Industry Standard Benchmark Suite for Machine Learning Performance," IEEE MICRO, Publ. by IEEE Comput. Soc., no. 0272-1732 _ 2020 IEEE, p. 9, 2020.
93. L. Zheng et al., "Ansor : Generating High-Performance Tensor Programs for Deep Learning," USENIX Assoc. 14th USENIX Symp. Oper. Syst. Des. Implement. Oper. Syst. Des. Implement. ({OSDI} 20) (pp. 863-879), vol. 4, p. 19, 2020.
94. J. P. Li, N. Mirza, B. Rahat, and D. Xiong, "Machine learning and credit ratings prediction in the age of fourth industrial revolution," Technol. Forecast. Soc. Chang. J., vol. 161, no. September, p. 120309, 2020, doi: 10.1016/j.techfore.2020.120309.

    A. Di Vaio, R. Palladino, R. Hassan, and O. Escobar, "Artificial intelligence and business models in the sustainable development goals perspective: A systematic literature review," J. Bus. Res., vol. 121, no. September, pp. 283–314, 2020, doi: 10.1016/j.jbusres.2020.08.019.
95. Z. Elamrani Abou Elassad, H. Mousannif, H. Al Moatassime, and A. Karkouch, "The application of machine learning techniques for driving behavior analysis: A conceptual framework and a systematic literature review," Eng. Appl. Artif. Intell., vol. 87, no. October 2019, p. 103312, 2020, doi: 10.1016/j.engappai.2019.103312.
96. V. H. Kumar, "Python Libraries , Development Frameworks and Algorithms for Machine Learning Applications," Int. J. Eng. Res. Technol., vol. 7, no. 04, pp. 210–217, 2018, [Online]. Available: http://www.ijert.org.
97. Abrol, M. Bhattarai, A. Fedorov, Y. Du, S. Plis, and V. Calhoun, "Deep residual learning for neuroimaging: An application to predict progression to Alzheimer's disease," J. Neurosci. Methods, vol. 339, no. September 2019, p. 108701, 2020, doi: 10.1016/j.jneumeth.2020.108701.
98. F. Mohr, M. Wever, and E. Hüllermeier, "ML-Plan : Automated machine learning via hierarchical planning," Mach. Learn. J. Springer, no. Published online : 03 July 2018, p. 21, 2018, doi: 10.1007/s10994-018-5735-z.
99. M. Goli, L. Iwanski, and A. Richards, "Accelerated machine learning using tensorflow and SYCL on OpenCL devices," ACM Int. Conf. Proceeding Ser. ACM Digit. Libr., vol. Part F1277, p. 4, 2017, doi: 10.1145/3078155.3078160.
100. H. Mirzaei, M. Fathollahi, and T. Givargis, "OpEB: Open physical environment benchmark for artificial intelligence," 2017 IEEE 3rd Int. Forum Res. Technol. Soc. Ind. (pp. 1-6). IEEE Xplore, 2017.
101. M. Allamanis, "The Adverse Effects of Code Duplication in Machine Learning Models of Code," Proc. ofthe 2019ACM SIGPLANInternational Symp. New Ideas, New Paradig. Reflections Program. Softw. (Onward! '19), Oct. 23–24, 2019, Athens, Greece. ACM, New York, NY, USA, 11 pages, no. October 23–24, 2019, pp. 143–153, 2019, [Online]. Available: https://doi.org/10.1145/3359591.3359735.
102. K. Chen, X. Song, X. Zhai, B. Zhang, B. Hou, and Y. Wang, "An Integrated Deep Learning Framework for Occluded Pedestrian Tracking," IEEE Access, vol. 7, pp. 26060–26072, 2019, doi: 10.1109/ACCESS.2019.2900296.
103. M. G. Kibria, K. Nguyen, G. P. Villardi, O. Zhao, K. Ishizu, and F. Kojima, "Big Data Analytics, Machine Learning, and Artificial Intelligence in Next-Generation Wireless Networks," IEEE Access, Open Access J., vol. 6, pp. 32328–32338, 2018, doi: 10.1109/ACCESS.2018.2837692.
104. Tecnol, "Performance Comparison of Deep Learning Frameworks in Image Classification Problems using Convolutional and Recurrent Networks," In2017 IEEE Colomb. Conf. Commun. Comput. 2017 Aug 16 (pp. 1-6). IEEE. 978-1-5386-1060-2/17/$31.00 ©2017 IEEE, no. In2017 IEEE Colombian Conference on Communications and Computing (COLCOM) 2017 Aug 16 (pp. 1-6). IEEE., p. 6, 2017.
105. Coleman et al., "DAWNBench: An End-to-End Deep Learning Benchmark and Competition," Thirty-first Annu. Conf. Neural Inf. Process. Syst., no. Nips, 2017, [Online]. Available: http://dawn.cs.stanford.edu/benchmark.

106.M. Y. Kamil, “A deep learning framework to detect Covid-19 disease via chest X-ray and CT scan images,” Int. J. Electr. Comput. Eng., vol. 11, no. 1, pp. 844–850, 2021, doi: 10.11591/ijece.v11i1.pp844-850.
107.Y. Zhao, Y. Yu, Y. Li, G. Han, and X. Du, “Machine learning based privacy-preserving fair data trading in big data market,” Inf. Sci. Journal; Publ. ELSEVIER Sci. INC , STE 800, 230 Park AVE, NEW YORK, USA, NY, 10169, vol. 478, pp. 449–460, 2019, doi: 10.1016/j.ins.2018.11.028.
108.N. Khan, A. Naim, M. R. Hussain, Q. N. Naveed, N. Ahmad, and S. Qamar, “The 51 V’s of big data: Survey, technologies, characteristics, opportunities, issues and challenges,” Proc. ACM Omni-layer Intell. Syst. Conf. (COINS’19). ACM, Heraklion, Crete, Greece 6 pages. DOI 10.1145/3312614.3312623, vol. Part F1481, pp. 19–24, 2019, doi: 10.1145/3312614.3312623.
109.Krishna, N. Rekulapelli, and B. P. Kauda, “Comparison of different deep learning frameworks,” Mater. Today Proc., no. xxxx, p. 5, 2020, doi: 10.1016/j.matpr.2020.09.608.
110.L. Li, J. Fang, H. Fu, J. Jiang, W. Zhao, and C. He, “swCaffe : a Parallel Framework for Accelerating Deep Learning Applications on Sunway TaihuLight,” 2018 IEEE Int. Conf. Clust. Comput., no. March 2019, pp. 413–422, 2018, doi: 10.1109/CLUSTER.2018.00087.
111.Ulker, S. Stuijk, H. Corporaal, and R. Wijnhoven, “Reviewing inference performance of state-of-the-art deep learning frameworks,” Proc. 23rd Int. Work. Softw. Compil. Embed. Syst. SCOPES 2020. ACM, pp. 48–53, 2020, doi: 10.1145/3378678.3391882.
112.and S. S. Lu, Huimin, Yujie Li, Min Chen, Hyoungseop Kim, “Brain Intelligence: Go Beyond Artificial Intelligence,” Met. Mater. Int. IEEE xplore, vol. 24, no. 2, pp. 371–379, 2018, doi: 10.1007/s12540-018-0053-3.
113.Ning and F. You, “Optimization under Uncertainty in the Era of Big Data and Deep Learning : When Machine Learning Meets Mathematical Programming,” Comput. Chem. Eng. ; Publ. PERGAMON-ELSEVIER Sci. LTD , BOULEVARD, LANGFORD LANE, KIDLINGTON, OXFORD, ENGLAND, OX5 1GB, no. 607, pp. 1–42, 2020.
114.N. Polyzotis, S. Roy, S. E. Whang, and M. Zinkevich, “Data Lifecycle Challenges in Production Machine Learning : A Survey,” Conf. SIGMOD Rec. June 2018 (Vol. 47, No. 2), vol. 47, no. 2, p. 12, 2018.
115.M. Al-Salim, A. Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmirghani, “Energy Efficient Big Data Networks: Impact of Volume and Variety,” IEEE Trans. Netw. Serv. Manag. Publ. IEEE-INST Electr. Electron. Eng. INC , 445 HOES LANE, PISCATAWAY, USA, NJ, 08855-4141 ISI Sci. Cit. Index Expand., vol. 15, no. 1, pp. 458–474, 2018, doi: 10.1109/TNSM.2017.2787624.
116.J. Nilsson, J. Delsing, and F. Sandin, “Autoencoder Alignment Approach to Run-Time Interoperability for System of Systems Engineering,” INES 2020 • 24th Int. Conf. Intell. Eng. Syst. • July 8-10, 2020 • Reykjavík, Iceland; IEEE Xplore, no. 978-1-7281-1059–2/20/$31.00 ©2020 IEEE, pp. 139–144, 2020.
117.P. Gibson, “Orpheus : A New Deep Learning Framework for Easy Deployment and Evaluation of Edge Inference,” In2020 IEEE Int. Symp. Perform. Anal. Syst. Softw. Perry Gibson, Jos´e Cano Sch. Comput. Sci. Univ. Glas. UK, no. 2020 Aug 23 (pp. 229–230). IEEE, p. 2, 2020.
118.M. Mustak, J. Salminen, L. Plé, and J. Wirtz, “Artificial intelligence in marketing: Topic modeling, scientometric analysis, and research agenda,” J. Bus. Res. Publ. ELSEVIER Sci. INC , STE 800, 230 Park AVE, NEW YORK, USA, NY, 10169, no. October, p. 16, 2020, doi: 10.1016/j.jbusres.2020.10.044.
119.J. Yang, X. L. Zhang, and P. Su, “Deep-Learning-Based Agile Teaching Framework of Software Development Courses in Computer Science Education,” Procedia Comput. Sci., vol. 154, pp. 137–145, 2018, doi: 10.1016/j.procs.2019.06.021.
120.Paszke et al., “PyTorch: An imperative style, high-performance deep learning library,” 33rd Conf. Neural Inf. Process. Syst. (NeurIPS 2019), Vancouver, Canada, vol. V1, no. NeurIPS, 2019.
121.J. Nilsson and J. Nilsson, System of Systems Interoperability Machine Learning Model System of Systems Interoperability Machine Learning Model. Dept. of Computer Science and Electrical Engineering Lule˚a University of Technology Lule˚a, Sweden: Printed by Luleå University of Technology, Graphic Production 2019 ISSN 1402-1757 ISBN 978-91-7790-458-8 (print) ISBN 978-91-7790-459-5 (pdf) Luleå 2019 www.ltu.se, 2019.
122.R. H. Hariri, E. M. Fredericks, and K. M. Bowers, “Uncertainty in big data analytics: survey, opportunities, and challenges,” J. Big Data ISI Emerg. Sources Cit. Index, vol. 6, no. 1, p. 16, 2019, doi: 10.1186/s40537-019-0206-3.
123.Luo, X. He, J. Zhan, L. Wang, W. Gao, and J. Dai, “Comparison and Benchmarking of AI Models and Frameworks on Mobile Devices,” arXiv Prepr. arXiv2005.05085. Cornell Univ. J., vol. abs/2005.0, 2020, [Online]. Available: https://www.semanticscholar.org/paper/3858ccebd64b63e27a8cbc2d4d4498e063c05688.
124.W. Zhang et al., “An Online-Offline Combined Big Data Mining Platform,” Proc. - 2017 IEEE 15th Int. Conf. Dependable, Auton. Secur. Comput. 2017 IEEE 15th Int. Conf. Pervasive Intell. Comput. 2017

IEEE 3rd Int. Conf. Big Data Intell. Compu, vol. 2018-Janua, pp. 1220–1225, 2018, doi: 10.1109/DASC-PICom-DataCom-CyberSciTec.2017.195.

125. M. C. H. Chiang, E. Lughofer, and E. Egrioglu, "Deep learning : emerging trends , applications and research challenges," Soft Comput. J., vol. 24, no. 11, pp. 7835–7838, 2020, doi: 10.1007/s00500-020-04939-z.
126. M. Maher et al., "SmartML : A Meta Learning-Based Framework for Automated Selection and Hyperparameter Tuning for Machine Learning Algorithms Mohamed Maher , Sherif Sakr To cite this version : HAL Id : hal-02087414 SmartML : A Meta Learning-Based Framework for Automated Se," EDBT 22nd Int. Conf. Extending Database Technol. Mar 2019, Lisbon, Port. 10.5441/002/edbt.2019.54 . hal- 02087414 HAL, 2019.
127. Valle-Cruz, J. I. Criado, R. Sandoval-Almazán, and E. A. Ruvalcaba-Gomez, "Assessing the public policy-cycle framework in the age of artificial intelligence: From agenda-setting to policy evaluation," J. Gov. Inf. Quarterly; Publ. ELSEVIER INC , 525 B STREET, STE 1900, SAN DIEGO, USA, CA, 92101-4495, vol. 37, no. 4, p. 101509, 2020, doi: 10.1016/j.giq.2020.101509.
128. S. Gupta, A. K. Kar, A. Baabdullah, and W. A. A. Al-Khowaiter, "Big data with cognitive computing: A review for the future," Int. J. Inf. Manag. ISI Soc. Sci. Cit. Index, vol. 42, no. June, pp. 78–89, 2018, doi: 10.1016/j.ijinfomgt.2018.06.005.
129. Anagnostopoulos, S. Zeadally, and E. Exposito, "Handling big data: research challenges and future directions," J. Supercomput. ISI Sci. Cit. Index Expand., vol. 72, no. 4, pp. 1494–1516, 2016, doi: 10.1007/s11227-016-1677-z.
130. R. Harris, "PrototypeML : A Neural Network Integrated Design and Development Environment," arXiv Prepr. arXiv2007.01097. 2020 Jul 1. Harvard Univ. Prepr. Under Rev., vol. V1, p. 10, 2020.
131. R. Perez-Vega, V. Kaartemo, C. R. Lages, N. Borghei Razavi, and J. Männistö, "Reshaping the contexts of online customer engagement behavior via artificial intelligence: A conceptual framework," J. Bus. Res., no. November, p. 9, 2020, doi: 10.1016/j.jbusres.2020.11.002.
132. V. Kovalev, A. Kalinovsky, and S. Kovalev, "Deep Learning with Theano, Torch, Caffe, TensorFlow, and Deeplearning4J: Which One Is the Best in Speed and Accuracy? Recognition of multi-drug resistant tubercolusis based on CT and X-ray image analysis View project UAV: back to base problem View project," XIII Int. Conf. Pattern Recognit. Inf. Process. SPRINGER, pp. 99–103, 2016, [Online]. Available: http://imlab.grid.by/.
133. Jain, A. Awan, Q. Anthony, H. Subramoni, and D. Panda, "Performance Characterization of DNN Training using TensorFlow and PyTorch on Modern Clusters," 2019 IEEE Int. Conf. Clust. Comput. (pp. 1-11). IEEE., vol. 2019-, p. 35, 2019, doi: 10.1109/CLUSTER.2019.8891042.
134. L'Heureux, K. Grolinger, H. F. Elyamany, and M. A. M. Capretz, "Machine Learning with Big Data: Challenges and Approaches," IEEE ACCESS J. Publ. IEEE-INST Electr. Electron. Eng. INC , 445 HOES LANE, PISCATAWAY, USA, NJ, 08855-4141 IEEE Access Index. byISI Sci. Cit. Index Expand., vol. 5, pp. 7776–7797, 2017, doi: 10.1109/ACCESS.2017.2696365.
135. M. Al-Salim, T. E. H. El-Gorashi, A. Q. Lawey, and J. M. H. Elmirghani, "Greening Big Data Networks: The Impact of Veracity," Mater. Sci. Eng. 3rd Int. Conf. Eng. Sci. Publ. IOP Publ. LTD , TEMPLE CIRCUS, TEMPLE WAY, BRISTOL, ENGLAND, BS1 6BE, pp. 1–14, 2018.
136. T. Zhao and X. Huang, "Design and implementation of DeepDSL: A DSL for deep learning," Comput. Lang. Syst. Struct., vol. 54, pp. 39–70, 2018, doi: 10.1016/j.cl.2018.04.004.
137. T. Zhao, X. Bing Huang, and Y. Cao, "DeepDSL: A compilation-based domain-specific language for deep learning," ICLR 2017 Int. Conf. Learn. Represent. arXiv 1701.02284 V1 [cs.PL], vol. 1, pp. 1–11, 2017.
138. L. Liu et al., "Deep Learning for Generic Object Detection : A Survey," Int. J. Comput. Vis. 128261–318 Springer, vol. 128, no. 2, pp. 261–318, 2020, doi: 10.1007/s11263-019-01247-4.
139. N. Soni, E. K. Sharma, N. Singh, and A. Kapoor, "Artificial Intelligence in Business: From Research and Innovation to Market Deployment," Procedia Comput. Sci. J. / Int. Conf. Comput. Intell. Data Sci. (ICCIDS 2019)., vol. 167, no. 2019, pp. 2200–2210, 2020, doi: 10.1016/j.procs.2020.03.272.
140. N. Papernot, P. Mcdaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," Proc. - 2016 IEEE Eur. Symp. Secur. Privacy, EURO S P 2016, pp. 372–387, 2016, doi: 10.1109/EuroSP.2016.36.
141. Y. Wang et al., "Benchmarking the Performance and Energy Efficiency of AI Accelerators for AI Training," Proc. - 20th IEEE/ACM Int. Symp. Clust. Cloud Internet Comput. CCGRID 2020, pp. 744–751, 2020, doi: 10.1109/CCGrid49817.2020.00-15.
142. Y. Wang, G. Y. Wei, and D. Brooks, "Benchmarking TPU, GPU, and CPU platforms for deep learning," arXiv Prepr. arXiv1907.10701. Coenell Univ., vol. 4, 2019.
143. J. Murphy, "Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges," Expert Syst. Appl., pp. 1–7, 2018, [Online].

Available: file:///F:/My desktop/My Theses/New papers from WOS and Scopuse/Deep learning algorithms for human activity recognition using mobile wearable sensor networks%3B State of the art and research.htm.

144. M. Zaharia et al., “Accelerating the Machine Learning Lifecycle with MLflow,” IEEE Comput. Soc. Tech. Comm. Data Eng., no. IEEE Data Eng. Bull. 41, 4 (2018): 39-45, pp. 39–45, 2018.
145. D. Panda, A. Awan, and H. Subramoni, “High performance distributed deep learning: A Beginner’s guide,” Proc. ACM SIGPLAN Symp. Princ. Pract. Parallel Program. PPOPP, p. 3, 2019, doi: 10.1145/3293883.3302260.
146. Rithika Shyam Chari, “Image Recognition And Study Of Hyperparameter Optimization Of Convolutional Neural Networks Using Tensorflow And Keras Frameworks Deep,” No. May 2018, 2018, [Online]. Available: Http://Library1.Nida.Ac.Th/Termpaper6/Sd/2554/19755.Pdf.
147. Osborne et al., “MLPerf Inference Benchmark,” 2020 ACM/IEEE 47th Annu. Int. Symp. Comput. Archit. (pp. 446-459). IEEE., vol. 2020-, 2020, doi: 10.1109/ISCA45697.2020.00045.
148. P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, “SQuad: 100,000+ questions for machine comprehension of text,” EMNLP 2016 - Conf. Empir. Methods Nat. Lang. Process. Proc., no. ii, pp. 2383–2392, 2016.
149. M. A. Goralski and T. K. Tan, “Artificial intelligence and sustainable development,” Int. J. Manag. Educ., vol. 18, no. 1, p. 9, 2020, doi: 10.1016/j.ijme.2019.100330.
150. M. Milutinovic, R. Zinkov, D. Song, W. Harvey, F. Wood, and W. Shen, “End-to-end Training of Differentiable Pipelines Across Machine Learning Frameworks,” 31st Conf. Neural Inf. Process. Syst. (NIPS 2017), Long Beach, CA, USA, no. Nips, 2017.
151. Shah, J. Wang, and Q. P. He, “Feature engineering in big data analytics for IoT-enabled smart manufacturing – Comparison between deep learning and statistical learning,” Comput. Chem. Eng. ISI Sci. Cit. Index Expand., vol. 141, p. 106970, 2020, doi: 10.1016/j.compchemeng.2020.106970.
152. B. Hansen and S. Bøgh, “Artificial intelligence and internet of things in small and medium-sized enterprises: A survey,” J. Manuf. Syst., no. October 2019, p. 11, 2020, doi: 10.1016/j.jmsy.2020.08.009.
153. J. Zacharias, M. Barz, and D. Sonntag, “A Survey on Deep Learning Toolkits and Libraries for Intelligent User Interfaces,” ACM Journal, arXiv Prepr. arXiv1803.04818. ACM, vol. 2, p. 10, 2018.
154. N. Khan, M. Alsaqer, H. Shah, G. Badsha, A. A. Abbasi, and S. Salehian, “The 10 Vs, issues and challenges of big data,” ACM Int. Conf. Proceeding Ser., no. September, pp. 52–56, 2018, doi: 10.1145/3206157.3206166.
155. Authors, F. Wu, T. Li, F. Luo, S. Wu, and C. Xiao, “A Survey on Deep Transfer Learning and Edge Computing for Mitigating the COVID-19 Pandemic,” J. Syst. Archit., p. 32, 2020, doi: https://doi.org/10.1016/j.sysarc.2020.101830.
156. J. Khan et al., “MIOpen: An Open Source Library For Deep Learning Primitives,” Proc. 30th Int. Conf. Comput. Graph. Mach. Vis., vol. 2744, no. int Petersburg, Russia, September 22-25, 2020, p. 11, 2020, [Online]. Available: https://www.semanticscholar.org/paper/9ea5d67ce98c52690c34b1de51be92f182cc872a.
157. J. Github, “Toolkits and Libraries for Deep Learning,” J. Digit. IMAGING Publ. SPRINGER , ONE NEW YORK PLAZA, SUITE 4600 , NEW YORK, United States, NY, 10004, p. 6, 2017, [Online]. Available: https://github.com/zer0n/deepframeworks.
158. Pester, C. Madritsch, T. Klinger, and X. L. de Gereña, “Deep Learning Frameworks for Convolutional Neural Networks—A Benchmark Test,” in International Conference on Remote Engineering and Virtual Instrumentation, 2019, pp. 817–831.
159. Q. Zhang et al., “A Survey on Deep Learning Benchmarks: Do We Still Need New Ones?,” in In International Conference on Remote Engineering and Virtual Instrumentation (pp. 817-831). Springer, Cham., 2018, pp. 36–49, doi: 10.1007/978-3-030-32813-9_5.
160. M. Modasshir, A. Quattrini Li, and I. Rekleitis, “Deep neural networks: A comparison on different computing platforms,” Proc. - 2018 15th Conf. Comput. Robot Vision, CRV 2018 978-1-5386-6481-0/18/$31.00 ©2018 IEEE, pp. 383–389, 2018, doi: 10.1109/CRV.2018.00060.
161. N. D. Cilia, C. De Stefano, F. Fontanella, C. Marrocco, M. Molinara, and A. S. Di, “An Experimental Comparison between Deep Learning and Classical Machine Learning Approaches for Writer Identification in Medieval Documents,” J. Imaging ISI Emerg. Sources Cit. Index, vol. 6, no. 9, pp. 1–15, 2020, doi: 10.3390/JIMAGING6090089.
162. K. Anding, L. Haar, G. Polte, J. Walz, and G. Notni, “Comparison of the performance of innovative deep learning and classical methods of machine learning to solve industrial recognition tasks,” Proc. SPIE; Proc. SPIE Vol. 11144 111440R-8; Jt. TC1 - TC2 Int. Conf. Photonics Educ. Meas. Sci. 2019, 2019, Jena, Ger., vol. 11144, no. September 2019, p. 26, 2019, doi: 10.1117/12.2530899.
163. X. Du et al., “Comparative study of distributed deep learning tools on supercomputers,” in International Conference on Algorithms and Architectures for Parallel Processing, 2018, pp. 122–137.

164. S. M. C. Loureiro, J. Guerreiro, and I. Tussyadiah, “Artificial intelligence in business: State of the art and future research agenda,” J. Bus. Res. Publ. ELSEVIER Sci. INC , STE 800, 230 Park AVE, NEW YORK, USA, NY, 10169, no. January 2019, p. 16, 2020, doi: 10.1016/j.jbusres.2020.11.001.
165. J. Y. Liou, X. Wang, S. Forrest, and C. J. Wu, “GEVO: GPU code optimization using evolutionary computation,” arXiv Prepr. arXiv2004.08140 (2020). under Rev. ACM TACO, vol. 2, pp. 1–25, 2020, doi: 10.1145/3418055.
166. H. V. Pham, T. Lutellier, W. Qi, and L. Tan, “CRADLE: Cross-Backend Validation to Detect and Localize Bugs in Deep Learning Libraries,” Proc. - Int. Conf. Softw. Eng. 2019 IEEE/ACM, vol. 2019-May, pp. 1027–1038, 2019, doi: 10.1109/ICSE.2019.00107.
167. S. Shi, Q. Wang, P. Xu, and X. Chu, “Benchmarking state-of-the-art deep learning software tools,” Proc. - 2016 7th Int. Conf. Cloud Comput. Big Data, CCBD 2016; IEEE; IEEE Comput. Sociiety, pp. 99–104, 2017, doi: 10.1109/CCBD.2016.029.
168. J. Nilsson, F. Sandin, and J. Delsing, “Interoperability and machine-to-machine translation model with mappings to machine learning tasks,” INES 2020 • 24th Int. Conf. Intell. Eng. Syst. • July 8-10, 2020 • Reykjavík, Iceland; EEE xplore, p. 7, 2020.
169. G. Lewis, “Component Mismatches Are a Critical Bottleneck to Fielding AI-Enabled Systems in the Public Sector,” Carnegie Mellon Univ. Softw. Eng. Inst., vol. 1, pp. 2–5, 2020.
170. Y. Sun, Y. Zhang, and H. Wang, “Select the model who knows the image best: a multi-model method,” in In Optoelectronic Imaging and Multimedia Technology VII (Vol. 11550, p. 115500O). International Society for Optics and Photonics. SPIE Digital Library, 2020, vol. 11550, p. 115500O.
171. R. Xu, S. Ma, and Y. Guo, “Performance analysis of different convolution algorithms in GPU environment,” 2018 IEEE Int. Conf. Networking, Archit. Storage, NAS 2018 - Proc., no. 61672526, pp. 1–10, 2018, doi: 10.1109/NAS.2018.8515695.
172. J. Wyrobek, “Predicting Bankruptcy at Polish Companies: A Comparison of Selected Machine Learning and Deep Learning Algorithms,” Zesz. Nauk. Uniw. Ekon. w Krakowie CEEOL online Libr., vol. 6, no. 6(978), pp. 41–60, 2018, doi: 10.15678/znuek.2018.0978.0603.
173. X. Liu et al., “Enhancing Veracity of IoT Generated Big Data in Decision Making,” 2018 IEEE Int. Conf. Pervasive Comput. Commun. Work. PerCom Work. 2018, pp. 149–154, 2018, doi: 10.1109/PERCOMW.2018.8480371.
174. Shridhar, P. Tomson, and M. Innes, “Interoperating Deep Learning models with ONNX.jl,” JuliaCon Proc. J. peer-review J., vol. 1, no. 1, p. 59, 2020, doi: 10.21105/jcon.00059.
175. J. Liu et al., “Usability Study of Distributed Deep Learning Frameworks For Convolutional Neural Networks,” ACM ISBN 978-x-xxxx-xxxx-x/YY/MM. Proc. Deep Learn. Day SIGKDD Conf. Knowl. Discov. Data Min. (KDD’18)., no. 1, p. 9, 2018, [Online]. Available: https://caffe2.ai/.
176. Holmes, D. Mawhirter, and B. Wu, “GRNN : Low-Latency and Scalable RNN Inference on GPUs,” ACM ISBN; Fourteenth EuroSys Conf. 2019 (EuroSys ’19), March 25–28, 2019, Dresden, Ger. 978-1-4503-6281-8/19/03. . . $15.00 https//doi.org/10.1145/3302424.3303949, no. ACM, New York, NY, USA, 16 pages, p. 16, 2019, [Online]. Available: https://doi.org/10.1145/3302424.3303949%0A1.
177. Yang, Y. Akimoto, D. W. Kim, and M. Udell, “Oboe : Collaborative Filtering for AutoML Model Selection,” Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, pp. 1173-1183. 2019. arXiv1808.03233v2, vol. 2, p. 11, 2019.
178. B. Arrieta et al., “Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI,” Inf. Fusion. 2020 Jun Publ. Elsevier, 1;5882-115 arXiv1910.10045v2, p. 72, 2020.
179. M. Gheisari, G. Wang, and Z. A. Bhuiyan, “A Survey on Deep Learning in Big Data,” IEEE 20th Int. Conf. Comput. Support. Coop. Work Des. (CSCWD 2016), no. July, 2017, doi: 10.1109/CSE-EUC.2017.215.
180. T. L. D. Huynh, E. Hille, and M. A. Nasir, “Diversification in the age of the 4th industrial revolution: The role of artificial intelligence, green bonds and cryptocurrencies,” Technol. Forecast. Soc. Chang. Publ. ELSEVIER Sci. INC , STE 800, 230 Park AVE, NEW YORK, USA, NY, 10169, vol. 159, no. June, p. 9, 2020, doi: 10.1016/j.techfore.2020.120188.
181. Gandomi and M. Haider, “Beyond the hype: Big data concepts, methods, and analytics,” Int. J. Inf. Manag. ISI Soc. Sci. Cit. Index Sci., vol. 35, no. 2, pp. 137–144, 2015, doi: 10.1016/j.ijinfomgt.2014.10.007.
182. S. Erevelles, N. Fukawa, and L. Swayne, “Big Data consumer analytics and the transformation of marketing,” J. Bus. Res. Sci. Direct ELSEVIER, vol. 69, no. 2, pp. 897–904, 2016, doi: 10.1016/j.jbusres.2015.07.001.
183. H. M. Mohammad Ali, M. S. Kod, A. M. A. Al-Salim, and C. Al-Amiri, “Algorithm for Greening Big Data Networks in Iraq under Veracity Dimension,” Mater. Sci. Eng. 3rd Int. Conf. Eng. Sci. Publ. IOP

Publ. LTD , TEMPLE CIRCUS, TEMPLE WAY, BRISTOL, ENGLAND, BS1 6BE, vol. 671, no. 1, p. 16, 2020, doi: 10.1088/1757-899X/671/1/012052.

184. M. S. Hadi, A. Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "Big data analytics for wireless and wired network design: A survey," Comput. Networks 132 180-199 Publ. ELSEVIER Sci. ISI Sci. Cit. Index Expand., pp. 1–23, 2018.
185. G. Lewis, "Software Engineering Institute Detecting Mismatches in Machine-Learning Systems," Softw. Eng. Inst. SEI, pp. 1–7, 2020.
186. Y. Zhou, S. Chen, Y. Wang, and W. Huan, "Review of research on lightweight convolutional neural networks," 2020 IEEE 5th Inf. Technol. Mechatronics Eng. Conf., pp. 1713–1720, 2020, doi: 10.1109/ITOEC49072.2020.9141847.
187. N. Singh, D. P. Singh, and B. Pant, "A comprehensive study of big data machine learning approaches and challenges," Proc. - 2017 Int. Conf. Next Gener. Comput. Inf. Syst. ICNGCIS 2017, pp. 58–63, 2018, doi: 10.1109/ICNGCIS.2017.14.
188. N. A. Ghani, S. Hamid, I. A. Targio Hashem, and E. Ahmed, "Social media big data analytics: A survey," Comput. Hum. Behav. UM Univ. Malaysia ELSEVIER, vol. 101, no. Computers in Human Behavior, 101, 417-428., pp. 417–428, 2019, doi: 10.1016/j.chb.2018.08.039.
189. T. L. Chen, S. H. Chien, and J. K. Lee, "ViennaCL++: Enable TensorFlow/Eigen via ViennaCL with OpenCL C++ flow," ACM Int. Conf. Proceeding Ser. ACM, pp. 1–2, 2018, doi: 10.1145/3204919.3207894.
190. M. Goli, L. Iwanski, J. Lawson, U. Dolinsky, and A. Richards, "OpenCL Acceleration for TensorFlow," arXiv:1605.02688, pp. 1–3, 2018.
191. V. Shankar, R. Roelofs, H. Mania, A. Fang, B. Recht, and L. Schmidt, "Evaluating Machine Accuracy on ImageNet," Proc. 37th Int. Conf. Mach. Learn. Vienna, Austria, PMLR 119, 2020. Copyr. 2020 by author(s), vol. 119, p. 11, 2020, [Online]. Available: https://www.semanticscholar.org/paper/7bcb4fdb1e7a5d0a30daa4938d37dbf06874dec2.
192. N. Shehab, M. Badawy, and H. Arafat, "Big Data Analytics Concepts, Technologies Challenges, and Opportunities," in In International Conference on Advanced Intelligent Systems and Informatics (pp. 92-101). Springer, Cham., 2019, pp. 92–101, doi: https://doi.org/10.1007/978-3-030-31129-2_9.
193. J. Qiu, Q. Wu, G. Ding, Y. Xu, and S. Feng, "A survey of machine learning for big data processing," Eurasip J. Adv. Signal Process. ISI Sci. Cit. Index Expand., vol. 2016, no. 1, 2016, doi: 10.1186/s13634-016-0355-x.
194. R. Rohit, B. Gupta, and K. Gola, "Analysis of Machine Learning for Processing Big Data in High Performance Computing: A Review," EAI Endorsed Trans. Cloud Syst. J. peer-reviewed EAI Index. by Comput. Database (ProQuest), p. 166353, 2018, doi: 10.4108/eai.7-9-2020.166353.
195. Prakash, N. Navya, and J. Natarajan, Big Data Preprocessing for Modern World: Opportunities and Challenges, vol. 26. Banglore: Springer International Publishing, 2019.
196. K. S. Divya, P. Bhargavi, and S. Jyothi, "Machine Learning Algorithms in Big data Analytics," Int. J. Comput. Sci. Eng. IJCSE) ISI Emerg. Sources Cit. Index (in Process., vol. 6, no. 1, pp. 63–70, 2018, doi: 10.26438/ijcse/v6i1.6370.
197. R. Bhatnagar, "Machine Learning and Big Data Processing: A Technological Perspective and Review," Adv. Intell. Syst. Comput. Publ. Springer, vol. 723, pp. 468–478, 2018, doi: 10.1007/978-3-319-74690-6_46.
198. R. Bhatnagar, Unleashing machine learning onto big data: Issues, challenges and trends, vol. 801. Springer International Publishing, 2019.
199. Kumar Padhi, S. S. Nayak, and B. N. Biswal, "Machine Learning for Big Data Processing: A Literature Review," Int. J. Innov. Res. Technol. IJIRT 140001 December 2018 | IJIRT | Vol. 5 Issue 7 | ISSN 2349-6002 samples, vol. 5, no. 7, p. 10, 2018, [Online]. Available: https://www.researchgate.net/publication/331497335.
200. W. Tsai, C. F. Lai, H. C. Chao, and A. V. Vasilakos, "Big data analytics: a survey," J. Big Data Publ. SPRINGERNATURE , CAMPUS, 4 CRINAN ST, LONDON, ENGLAND, N1 9XW. ISI Indexed., vol. 2, no. 1, pp. 1–32, 2015, doi: 10.1186/s40537-015-0030-3.
201. T. S. Demidova and A. A. Sobolev, "Review and comparative analysis of machine learning libraries for machine learning," http://journals.rudn.ru/miph, vol. 27, no. 4, pp. 305–315, 2019, doi: 10.22363/2658-4670-2019-27-4-305-315.
202. R. O. B. Ashmore, R. Calinescu, and C. Paterson, "Assuring the Machine Learning Lifecycle: Desiderata, Methods, and Challenges," arXiv 1905 . 04223v1 [cs . LG] 10 May 2019 Methods , Challenges, 2019.
203. Y. Yang, S. Liu, and N. Xie, "Uncertainty and grey data analytics," J. Mar. Econ. Manag. Publ. Index. by Emerald Publ. Limited., vol. 2, no. 2, pp. 73–86, 2019, doi: 10.1108/maem-08-2019-0006.

204. S. X. Zou et al., “Distributed training large-scale deep architectures,” Int. Conf. Adv. Data Min. Appl. pp. 18-32. Springer, Cham, 2017, vol. 10604 LNAI, pp. 18–32, 2017, doi: 10.1007/978-3-319-69179-4_2.
205. T. Guo, C. Xu, S. He, B. Shi, C. Xu, and D. Tao, “Robust Student Network Learning,” IEEE Trans. Neural Networks Learn. Syst. Publ. IEEE-INST Electr. Electron. Eng. INC , 445 HOES LANE, PISCATAWAY, USA, NJ, 08855-4141, vol. 31, pp. 2455–2468, 2020, doi: 10.1109/TNNLS.2019.2929114.
206. Siddiqa, A. Karim, and A. Gani, “Big data storage technologies: a survey,” Front. Inf. Technol. Electron. Eng. Publ. ZHEJIANG UNIV , Editor. BOARD, 20 YUGU RD, HANGZHOU, PEOPLES R CHINA, 310027, vol. 18, no. 8, pp. 1040–1070, 2017, doi: 10.1631/FITEE.1500441.
207. R. Weiss, “Social Set Analysis: A Set Theoretical Approach to Big Data Analytics,” IEEE Access Journal; Publ. IEEE-INST Electr. Electron. Eng. INC , 445 HOES LANE, PISCATAWAY, USA, NJ, 08855-4141. ISI Index., vol. 4, p. 30, 2016.
208. Li et al., “A Survey on optimized implementation of deep learning models on the NVIDIA Jetson platform,” J. Syst. Archit., p. 15, 2019, doi: https://doi.org/10.1016/j.sysarc.2019.01.011.
209. Park, J. H. Lee, Y. Oh, S. Ha, and S. Lee, “Privacy attacks against deep learning models and their countermeasures,” J. Syst. Archit. J., vol. 9, no. 114 (2021) 101940, 2021, [Online]. Available: www.elsevier.com/locate/sysarc%0APrivacy.
210. J. Nilsson, “Semantic Interoperability in Industry 4 . 0 : Survey of Recent Developments and Outlook,” 2018 IEEE 16th Int. Conf. Ind. Informatics (INDIN), pp. 127-132. IEEE, 2018, pp. 127–132, 2018.
211. R. Wei and L. Schwartz, “DLVM : A modern compiler framework for neural network DSLs,” 31st Conf. Neural Inf. Process. Syst. (NIPS 2017), Long Beach, CA, USA. ACM, no. Nips, p. 8, 2017.
212. L. Liu, Y. Wu, W. Wei, W. Cao, S. Sahin, and Q. Zhang, “Benchmarking deep learning frameworks: Design considerations, metrics and beyond,” Proc. - Int. Conf. Distrib. Comput. Syst., vol. 2018-July, pp. 1258–1269, 2018, doi: 10.1109/ICDCS.2018.00125.
O. Valery, P. Liu, and J. Wu, “A collaborative CPU-GPU approach for deep learning on mobile devices,” Concurr. Comput. Pr. Exper journal. 2019;e5225; Publ. WILEY , 111 RIVER ST, HOBOKEN, USA, NJ, 07030-5774; Index. by ISI SCIE, no. February, pp. 1–21, 2019, doi: 10.1002/cpe.5225.
213. S. Alghunaim and H. H. Al-Baity, “On the Scalability of Machine-Learning Algorithms for Breast Cancer Prediction in Big Data Context,” IEEE Access J. Publ. IEEE-INST Electr. Electron. Eng. INC , 445 HOES LANE, PISCATAWAY, USA, NJ, 08855-4141 Index. by ISI Sci. Cit. Index Expand., vol. 7, pp. 91535–91546, 2019, doi: 10.1109/ACCESS.2019.2927080.
214. J. Rauber, R. Zimmermann, M. Bethge, and W. Brendel, “Foolbox Native : Fast adversarial attacks to benchmark the robustness of machine learning models in PyTorch , Statement of need,” J. open source Softw. JOSS, vol. 5, pp. 3–5, 2020, doi: 10.21105/joss.02607.
215. J. Bhatia, “A Survey of Deep-learning Frameworks,” Int. Conf. Inven. Syst. Control, p. 7, 2020, [Online]. Available: https://sci-hub.se/https://ieeexplore.ieee.org/abstract/document/8068684.
216. M. S. S. B. N. R. fixed-term.Lukas.Schott, “COMPARATIVE STUDY OF CAFFE, NEON, THEANO, AND TORCH FOR DEEP LEARNING,” Neurosurg. J., vol. 62, no. 2, pp. 294–310, 2016, doi: 10.1227/01.NEU.0000297044.82035.57.
217. L. Liu, Y. Wu, W. Wei, W. Cao, S. Sahin, and Q. Zhang, “Benchmarking deep learning frameworks: Design considerations, metrics and beyond,” Proc. - Int. Conf. Distrib. Comput. Syst. IEEE Comput. Soc., vol. 2018-July, pp. 1258–1269, 2018, doi: 10.1109/ICDCS.2018.00125.
218. T. Anuprathibha and C. S. K. Selvib, “A survey of twitter sentiment analysis,” IIOAB J., vol. 7, no. 9Special Issue, pp. 374–378, 2016, doi: 10.26438/ijcse/v6i11.644648.
219. R. Deo and S. Panigrahi, “Performance Assessment of Machine Learning Based Models for Diabetes Prediction,” 2019 IEEE Healthc. Innov. Point Care Technol. HI-POCT 2019, no. 11, pp. 147–150, 2019, doi: 10.1109/HI-POCT45284.2019.8962811.
220. Expo, “Distributed Machine Learning with a Serverless Architecture,” 2019, IEEE INFOCOM 2019 - IEEE Conf. Comput. Commun., p. 47, 2019.
221. E. Papalexakis, C. Faloutsos, and N. D. Sidiropoulos, “Tensors for Data Mining and Data Fusion : Models , Applications , and Scalable Algorithms r r r,” ACM Trans. Intell. Syst. Technol. 8, 2, Artic. 16 (October 2016), 44 pages, vol. 8, no. 2, p. 44, 2016, doi: DOI: http://dx.doi.org/10.1145/2915921.
222. Y. Zhang, “Fast Convolutional Neural Networks with Fine-Grained FFTs,” Proc. ofthe 2020 Int. Conf. Parallel Archit. Compil. Tech. (PACT ’20), Oct. ACM, no. 2, pp. 255–265, 2020.
223. Awan, H. Subramoni, and D. K. Panda, “An In-depth Performance Characterization of CPU- and GPU-based DNN Training on Modern Architectures,” Proc. MLHPC 2017 Mach. Learn. HPC Environ. - Held conjunction with SC 2017 Int. Conf. High Perform. Comput. Networking, Storage Anal. ACM, New York, NY, USA, p. 8, 2017, doi: 10.1145/3146347.3146356.

224. Z. Woldaregay et al., “Data-driven modeling and prediction of blood glucose dynamics: Machine learning applications in type 1 diabetes,” J. Artif. Intell. Med., vol. 98, no. August 2018, pp. 109–134, 2019, doi: 10.1016/j.artmed.2019.07.007.
225. Fanaee-t and J. Gama, “Tensor-based anomaly detection: An interdisciplinary survey Hadi,” Knowledge-Based Syst. Publ. ELSEVIER Sci., vol. 98, pp. 130–147, 2016, doi: 10.1016/j.knosys.2016.01.027.
226. S. Jin, L. Li, and Y. Zhu, “A consensus-based global optimization method for high dimensional machine learning problems,” Comput. Sci. Math. arXiv Optim. Control, vol. 2, no. Corpus ID: 202712742, published 2019, pp. 1–30, 2019, doi: DOI:10.1051/cocv/2020046.
227. W. Wang et al., “SINGA: Putting Deep Learning in the Hands of Multimedia Users,” MM 2015 - Proc. 2015 ACM Multimed. Conf., pp. 25–34, 2015, doi: 10.1145/2733373.2806232.
228. Koutsoukas, K. Monaghan, X. Li, and J. Huan, “Deep-learning: Investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data,” J. Cheminformatics Springer, vol. 9, no. 1, 2017, doi: 10.1186/s13321-017-0226-y.
229. Fronzetti Colladon and P. A. Gloor, “Measuring the impact of spammers on e-mail and Twitter networks,” Int. J. Inf. Manag. Sci. Direct, vol. 48, pp. 254–262, 2019, doi: 10.1016/j.ijinfomgt.2018.09.009.
230. Chu, K. Hamidouche, A. Venkatesh, A. Awan, and D. Panda, “CUDA Kernel Based Collective Reduction Operations on Large-scale GPU Clusters,” 2016 16th IEEE/ACM Int. Symp. Clust. Cloud Grid Comput. (pp. 726-735). IEEE, 2016, doi: 10.1109/CCGrid.2016.111.
231. P. Singhal and S. Pareek, “Artificial neural network for prediction of breast cancer,” Proc. Int. Conf. I-SMAC (IoT Soc. Mobile, Anal. Cloud), I-SMAC 2018 IEEE Xplore, pp. 464–468, 2019, doi: 10.1109/I-SMAC.2018.8653700.
232. Suvarnamukhi and M. Seshashayee, “Big data processing system for diabetes prediction using machine learning technique,” Int. J. Innov. Technol. Explor. Eng. scopus, vol. 8, no. 12, pp. 4478–4483, 2019, doi: 10.35940/ijitee.L3515.1081219.
233. E.-D. and K. Maalmi, “Performance evaluation of machine learning based Big data processing framework for prediction of heart disease,” 2019 Int. Conf. Intell. Syst. Adv. Comput. Sci. (ISACS), pp. 1-5. IEEE, 2019, IEEE Xplore, p. 5, 2019.
234. K. Grolinger, M. A. M. Capretz, and L. Seewald, “Energy consumption prediction with big data: Balancing prediction accuracy and computational resources,” Proc. - 2016 IEEE Int. Congr. Big Data, BigData Congr. 2016, pp. 157–164, 2016, doi: 10.1109/BigDataCongress.2016.27.
235. K. Jakhar and N. Hooda, “Big data deep learning framework using keras: A case study of pneumonia prediction,” 2018 4th Int. Conf. Comput. Commun. Autom. ICCCA 2018, no. December, p. 6, 2018, doi: 10.1109/CCAA.2018.8777571.
236. Viégas et al., “Visualizing Dataflow Graphs of Deep Learning Models in TensorFlow,” J. EEE Trans. Vis. Comput. Graph. Publ. IEEE Comput. SOC , 10662 LOS VAQUEROS CIRCLE, PO BOX 3014, LOS ALAMITOS, USA, CA, 90720-1314, vol. 24, no. 1, 2018, doi: 10.1109/TVCG.2017.2744878.
237. T. Akiba, K. Fukuda, and S. Suzuki, “ChainerMN: Scalable Distributed Deep Learning Framework,” 31st Conf. Neural Inf. Process. Syst. (NIPS 2017), Long Beach, CA, USA TSUBAME e-Science J. (ESJ). Tokyo Inst. Technol., vol. 1, no. arXiv preprint arXiv:1710.11351., p. 6, 2017, [Online]. Available: http://arxiv.org/abs/1710.11351.
238. P. Migdal, “Predictive analytics in education: a comparison of deep learning frameworks,” Educ. Inf. Technol., vol. 19, pp. 1–19, 2019.
239. M. Woolf, “Landscape Similarity Analysis Using Texture Encoded Deep-Learning Features on Unclassified Remote Sensing Imagery,” Remote SENSING. Web Sci. Publ. MDPI , ST ALBAN-ANLAGE 66, BASEL, SWITZERLAND, CH-4052, p. 25, 2021, [Online]. Available: https://www.mdpi.com/2072-4292/13/3/492.
240. M. Soumith, “DeepLink: Recovering issue-commit links based on deep learning,” J. Syst. Softw., p. 13, 2019, [Online]. Available: https://doi.org/10.1016/j.jss.2019.110406 0164-1212/© 2019 Elsevier Inc. All rights reserved.
241. R. E. Almamlook, K. M. Kwayu, M. R. Alkasisbeh, and A. A. Frefer, “Comparison of machine learning algorithms for predicting traffic accident severity,” 2019 IEEE Jordan Int. Jt. Conf. Electr. Eng. Inf. Technol. JEEIT 2019 - Proc., no. June, pp. 272–276, 2019, doi: 10.1109/JEEIT.2019.8717393.
242. S. Serapião, G. S. Corrêa, F. B. Gonçalves, and V. O. Carvalho, “Combining K-Means and K-Harmonic with Fish School Search Algorithm for data clustering task on graphics processing units,” Appl. Soft Comput. J. Sci. Direct ISI SCIE Publ. ELSEVIER , RADARWEG 29, AMSTERDAM, NETHERLANDS, 1043 NX, vol. 41, p. 15, 2016, doi: 10.1016/j.asoc.2015.12.032.
243. M. Z. Alom et al., “A state-of-the-art survey on deep learning theory and architectures,” Electron. J. Publ. MDPI , ST ALBAN-ANLAGE 66, BASEL, SWITZERLAND, CH-4052, vol. 8, no. 3, pp. 1–67, 2019, doi: 10.3390/electronics8030292.

244. Kim, H. Nam, W. Jung, and J. Lee, “Performance Analysis of CNN Frameworks for GPUs Convolutional Neural Network Deep Learning Framework GPU Library,” 2017 IEEE Int. Symp. Perform. Anal. Syst. Softw. (ISPASS), pp. 55-64. IEEE, 2017. Multicore Comput. Reseaerch IEEE Xplore Lab. Cent. Manycore Program. SEOUL Natl. Univ., p. 40, 2017.
245. Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun, “DetNet: A Backbone network for Object Detection,” Cornell Univ. J. arXiv 1804 . 06215v2 [cs . CV] 19 Apr 2018 DetNet A Backbone Netw. Object, vol. 2, pp. 1–17, 2018, [Online]. Available: https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=DetNet%3A+A+Backbone+network+for+Object+Detection&btnG=.
246. Y. E. Wang, C. J. Wu, X. Wang, K. Hazelwood, and D. Brooks, “Exploiting parallelism opportunities with deep learning frameworks,” ACM Trans. Archit. Code Optim. (TACO), 18(1), 1-23 Publ. ASSOC Comput. Mach. , 2 PENN PLAZA, STE 701, NEW YORK, USA, NY, 10121-070 ISIScience Cit. Index Expand., pp. 1–21, 2020.
247. Kotras, “Precision Medicine Informatics: Principles, Prospects, and Challenges,” IEEE Access, p. 20, 2020, [Online]. Available: https://ieeexplore-ieee-org.ezproxy.utm.my/ielx7/6287639/8948470/08957137.pdf.
248. Adie, I. Pradana, and Pranowo, “Parallel computing accelerated image inpainting using GPU CUDA, Theano, and Tensorflow,” 2018 10th Int. Conf. Inf. Technol. Electr. Eng. (pp. 621-625). IEEE, no. 978-1-5386-4738–7, p. 6, 2018, doi: 10.1109/ICITEED.2018.8534858.
249. K. Grolinger, A. L’Heureux, M. A. M. Capretz, and L. Seewald, “Energy forecasting for event venues: Big data and prediction accuracy,” Energy Build. Publ. ELSEVIER Sci. SA , PO BOX 564, LAUSANNE, SWITZERLAND, 1001, ISIScience Cit. Index Expand., vol. 112, pp. 222–233, 2016, doi: 10.1016/j.enbuild.2015.12.010.
250. S. Saif, D. Datta, A. Saha, S. Biswas, and C. Chowdhury, “Data Science and AI in IoT Based Smart Healthcare: Issues, Challenges and Case Study,” in In Enabling AI Applications in Data Science, pp. 415-439. Springer, Cham, 2020., Springer, 2020, pp. 415–439.
251. metode penelitian Nursalam, 2016 and A. . Fallis, “Robotics, artificial intelligence, and the evolving nature of work,” J. Chem. Inf. Model. Publ. AMER Chem. SOC , 1155 16TH ST, NW, WASHINGTON, USA, DC, 20036, vol. 53, no. 9, pp. 1689–1699, 2020.
252. R. Mayer and H. A. Jacobsen, “Scalable deep learning on distributed infrastructures: Challenges, techniques, and tools,” ACM Comput. Surv. Publ. ASSOC Comput. Mach. , 2 PENN PLAZA, STE 701, NEW YORK, USA, NY, 10121-0701, vol. 53, no. 1, p. 37, 2020, doi: 10.1145/3363554.
253. Akoka, I. Comyn-Wattiau, and N. Laoufi, “Research on Big Data – A systematic mapping study,” Comput. Stand. Interfaces, Publ. ELSEVIER , RADARWEG 29, AMSTERDAM, NETHERLANDS, 1043 NX, vol. 54, pp. 105–115, 2017, doi: 10.1016/j.csi.2017.01.004.
254. S. Liu, D. Huang, and Y. Wang, “Receptive Field Block Net for Accurate and Fast Object Detection,” InProceedings Eur. Conf. Comput. Vis. 2018 (pp. 385-400 Springer, vol. abs/1711.0, 2018, doi: 10.1007/978-3-030-01252-6_24.
255. Oztekin, D. Delen, A. Turkyilmaz, and S. Zaim, “A machine learning-based usability evaluation method for eLearning systems,” Decis. Support Syst. Journal, Publ. ELSEVIER; Sci., p. 11, 2015.
256. Mendelson, “Security and Privacy in the Age of Big Data and Machine Learning,” Comput. Journal; Publ. IEEE Comput. SOC , 10662 LOS VAQUEROS CIRCLE, PO BOX 3014, LOS ALAMITOS, USA, CA, 90720-1314 IEEE, vol. 52, pp. 65–70, 2019, doi: 10.1109/MC.2019.2943137.
257. Papageorgiou, K. Hadjigeorgiou, A. N. Ness, and SCOPUS, “A big data approach to developing a smart pedestrian network (SPN) system,” J. WSEAS Trans. Environ. Dev., vol. 15, no. E-ISSN: 2224-3496, p. 7, 2019.
258. Berrar, P. Lopes, and W. Dubitzky, “Incorporating domain knowledge in machine learning for soccer outcome prediction,” Mach. Learn. Publ. SPRINGER , VAN GODEWIJCKSTRAAT 30, DORDRECHT, NETHERLANDS, 3311 GZ, vol. 108, no. 1, pp. 97–126, 2019, doi: 10.1007/s10994-018-5747-8.
259. M. Childs and N. R. Washburn, “Embedding domain knowledge for machine learning of complex material systems,” MRS Commun. J. Publ. online by Cambridge Univ. Press 10 July 2019, vol. 9, no. MRS Communications, Volume 9 , Issue 3 , September 2019 , pp. 806–820, pp. 1–15, 2019, doi: 10.1557/mrc.2019.90.
260. Kotenko, I. Saenko, A. Kushnerevich, and A. Branitskiy, “Attack detection in IoT critical infrastructures : a machine learning and big data processing approach,” 2019 27th Euromicro Int. Conf. Parallel, Distrib. Network-Based Process. (PDP); IEEE, pp. 340–347, 2019, doi: 10.1109/PDP.2019.00057.

261. Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun, "DetNet: A Backbone network for Object Detection," ArXiv, vol. abs/1804.0, 2018, [Online]. Available: https://www.semanticscholar.org/paper/af77faec0f71d934013c1f17b368edc9e845235f.
262. Rauber, M. Bethge, and W. Brendel, "EagerPy: Writing code that works natively with PyTorch, tensorflow, JAX, and NumPy," arXiv Prepr. arXiv2008.04175 NASA ADS Google Sch. Semant. Sch., vol. 1, pp. 1–9, 2020.
263. Kim, "Benchmarking deep learning techniques for face recognition," J. Vis. Commun. Image R. Sci., p. 14, 2019, doi: https://doi.org/10.1016/j.jvcir.2019.102663.
264. S. Bahrampour, N. Ramakrishnan, L. Schott, and M. Shah, "Comparative Study of Deep Learning Software Frameworks," arXiv Prepr. arXiv1511.06435, vol. 3, p. 9, 2016, [Online]. Available: http://arxiv.org/abs/1511.06435.
265. Zhang, "Seesaw-Net: Convolution Neural Network With Uneven Group Convolution," arXiv Prepr. arXiv1905.03672 https//api.semanticscholar.org/arXiv1905.03672, vol. abs/1905.0, 2019, [Online]. Available: https://www.semanticscholar.org/paper/afb1cacdf860d1eff7e3e5eea240b2f730a68665.
266. Qiu, C. Chen, S. Liu, and B. Zeng, "SlimConv: Reducing Channel Redundancy in Convolutional Neural Networks by Weights Flipping," arXiv Prepr. arXiv2003.07469 Univ. Electron. Sci. Technol. China, vol. 1, p. 20, 2020.
267. S. Ren, X. Lu, and T. Wang, "Application of Ontology in Medical Heterogeneous Data Integration," 2018 IEEE 3rd Int. Conf. Big Data Anal. Appl. IEEE, pp. 150–155, 2018.
268. Eric and M. Issa, "Machine Learning Techniques for Gait Biometric Recognition," Libr. Congr. Control Number 2015960234; ISBN 978-3-319-29086-7 DOI 10.1007/978-3-319-29088-1 ISBN 978-3-319-29088-1 (eBook). SPRINGER 2016, p. 29, 2016, doi: 10.1007/978-3-319-29088-1.
269. T. Ben-Nun and T. Hoefler, "Demystifying parallel and distributed deep learning: An in-depth concurrency analysis," ACM Comput. Surv., vol. 52, no. 4, 2019, doi: 10.1145/3320060.
270. N. Parmar et al., "Mesh-tensorflow: Deep learning for supercomputers," 32nd Conf. Neural Inf. Process. Syst. (NeurIPS 2018), Montréal, Canada., vol. pp. 10414-, 2018, [Online]. Available: https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=Mesh-tensorflow%3A+Deep+learning+for+supercomputers&btnG=.
271. Viebke, S. Memeti, S. Pllana, and A. Abraham, "CHAOS: a parallelization scheme for training convolutional neural networks on Intel Xeon Phi," J. Supercomput., vol. 75, no. 1, 2019, doi: 10.1007/s11227-017-1994-x.
272. Chen And Q. Huo, "Scalable Training Of Deep Learning Machines By Incremental Block Training With Intra-Block Parallel Optimization And Blockwise Model-Update Filtering," 2016 Ieee Int. Conf. Acoust. Speech Signal Process. (Pp. 5880-5884). Ieee., Pp. 5880–5884, 2016, [Online].
273. Nurvitadhi, D. Sheffield, J. Sim, A. Mishra, G. Venkatesh, and D. Marr, "Accelerating binarized neural networks: Comparison of FPGA, CPU, GPU, and ASIC," in In proceeding International Conference on Field-Programmable Technology (FPT), pp. 77-84. IEEE, 2016., May 2017, pp. 77–84, doi: 10.1109/FPT.2016.7929192.
274. Awan, C. Chu, H. Subramoni, D. Panda, and J. Bedorf, "Scalable distributed DNN training using TensorFlow and CUDA-Aware MPI: Characterization, designs, and performance evaluation," Submitt. to IEEE IPDPS 2019 (Main Track) Peer Rev. Ohio State Univ. Leiden Univ. Leiden, Netherlands. IEEE ACCESS, vol. 1, p. 10, 2019, doi: 10.1109/CCGRID.2019.00064.
275. W. N. van Wieringen, "On the mean squared error of the ridge estimator of the covariance and precision matrix," Stat. Probab. Lett. Publ. ELSEVIER , RADARWEG 29, AMSTERDAM, NETHERLANDS, 1043 NX, ISI Sci. Cit. Index Expand., vol. 123, pp. 88–92, 2017, doi: 10.1016/j.spl.2016.12.002.
276. Kerkeni et al., "Overcoming Challenges in Predictive Modeling of Laser-Plasma Interaction Scenarios. The Sinuous Route from Advanced Machine Learning to Deep Learning," IntechOpen, world's Lead. Publ. We are IntechOpen, Built by Sci. Sci. Open Access books, no. tourism, p. 13, 2018, doi: http://dx.doi.org/10.5772/intechopen.72844 85.
277. Asaadi and B. Chapman, "Comparative study of deep learning framework in HPC environments," 2017 New York Sci. Data Summit (NYSDS), pp. 1-7. IEEE, 2017. IEEE Xplore, p. 7, 2017, doi: 10.1109/NYSDS.2017.8085040.
278. U. Chavan and D. Kulkarni, "Performance issues of parallel, scalable convolutional neural networks in deep learning," in Advances in Intelligent Systems and Computing, vol. 810, NEW YORK: Springer Verlag, 2019, pp. 333–343.
279. Boden, A. Spina, T. Rabl, and V. Markl, "Benchmarking data flow systems for scalable machine learning," Proc. 4th ACM SIGMOD Work. Algorithms Syst. MapReduce Beyond, BeyondMR 2017, no. ACM ISBN 978-1-4503-5019-8/17/05., p. 10, 2017, doi: 10.1145/3070607.3070612.

280. Kim, H. Nam, W. Jung, and J. Lee, "Performance analysis of CNN frameworks for GPUs," in ISPASS 2017 - IEEE International Symposium on Performance Analysis of Systems and Software, Jul. 2017, pp. 55–64, doi: 10.1109/ISPASS.2017.7975270.
281. R. MAYER and H. A. JACOBSEN, "Scalable deep learning on distributed infrastructures: Challenges, techniques and tools," ACM Comput. Surv. 1, 1, Artic. 1 (September, vol. 1, no. 1, p. 35, 2019.
282. T. Haryanto, H. Suhartanto, and X. Lie, "Past, present, and future trend of GPU computing in deep learning on medical images," in In 2017 International Conference on Advanced Computer Science and Information Systems (ICACSIS), pp. 21-28. IEEE, 2017, May 2018, vol. 2018-Janua, pp. 21–27, doi: 10.1109/ICACSIS.2017.8355007.
283. J. Pfeiffer, J. Neville, and P. N. Bennett, "Overcoming relational learning biases to accurately predict preferences in large scale networks," WWW 2015 - Proc. 24th Int. Conf. World Wide Web; ACM 978-1-4503-3469-3/15/05, pp. 853–863, 2015, doi: 10.1145/2736277.2741668.
284. Alonso, "The Practice of Labeling : Everything You Always Wanted to Know About Labeling (But Were Afraid to Ask)," Companion Proc. 2019 World Wide Web Conf. (WWW '19 Companion), May 13–17, 2019, San Fr. CA, USA. ACM, New York, NY, USA, 1 page, p. 1293, 2019, doi: https://doi.org/10.1145/3308560.3320083.
285. Corrieri and P. P. Id, "Challenges of machine learning model validation using correlated behaviour data : Evaluation of cross-validation strategies and accuracy measures," PLoS ONE J. 15(7) e0236092. Publ. PUBLIC Libr. Sci. , 1160 Batter. STREET, STE 100, SAN Fr. USA, CA, 9411 https//doi.org/10.1371/journal.pone.0236092, pp. 1–14, 2020, doi: 10.1371/journal.pone.0236092.
286. S. Tamy, H. Belhadaoui, M. A. Rabbah, N. Rabbah, and M. Rifi, "AN EVALUATION OF MACHINE LEARNING ALGORITHMS TO DETECT ATTACKS IN SCADA NETWORK," 2019 7th Mediterr. Congr. Telecommun. ;IEEE, pp. 1–5, 2019.
287. Wu, Q. Li, Z. Fu, W. Zhu, Y. Zhang, and L. Zhang, "Enabling Flexible Resource Allocation in Mobile Deep Learning Systems," IEEE Trans. Parallel Distrib. Syst. IEEE XPLORE, vol. 30, no. 2, 2019, doi: 10.1109/TPDS.2018.2865359.
288. H. Mei, M. Bansal, and M. R. Walter, "What to talk about and how? Selective generation using LSTMs with coarse-to-fine alignment," 2016 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. NAACL HLT; 2016 - Proc. Conf. Publ. Assoc. Comput. Linguist., vol. 2, pp. 720–730, 2016, doi: 10.18653/v1/n16-1086.
289. Eberhard, S. Walk, L. Posch, and D. Helic, "Proceedings of IUI 2019," IUI '19 24th Int. Conf. Intell. User Interfaces Mar. del Ray Calif. March, 2019 ACM, p. 10, 2019, [Online]. Available: https://doi.org/10.1145/3301275.3308446.
290. A. Myers, "IUI4EUD : intelligent user interfaces for end-user development," IUI '20 Proc. 25th Int. Conf. Intell. User InterfacesMarch 2020; ACM Digit. Libr., p. 4, 2020, doi: https://doi.org/10.1145/3377325.3380622.
291. Boehm et al., "SystemDS : A Declarative Machine Learning System for the End-to-End Data Science Lifecycle," 10th Annu. Conf. Innov. Data Syst. Res. (CIDR '20) January 12-15, 2020, Amsterdam, Netherlands, vol. V2, p. 8, 2020, [Online]. Available: http://creativecommons.org/licenses/by/3.0/.
292. E. Schüle, M. Bungeroth, A. Kemper, S. Günnemann, and T. Neumann, "MLearn : A Declarative Machine Learning Language for Database Systems," DEEM'19 Proc. 3rd Int. Work. Data Manag. End-to-End Mach. Learn. 2019 Artic. No. 7 Pages 1–4https//doi.org/10.1145/3329486.3329494 ACM Digit. Libr., pp. 3–6, 2019, [Online]. Available: https://doi.org/10.1145/3329486.3329494.
293. Choi, "Novel data mining paradigms based on soft computing and machine learning in the current and upcoming information society revolution," J. Concurr. Comput. Exp. 2020;e5937. https//doi.org/10.1002/cpe.5937 wileyonlinelibrary.com/journal/cpe ©2020, no. June, pp. 8–10, 2020, doi: 10.1002/cpe.5937.
294. Muwawa, J. Nestor, and K. A. Ogudo, "Practical Implementation of Machine Learning And Predictive Analytics In Cellular Network Transactions In Real Time," 2018 Int. Conf. Adv. Big Data, Comput. Data Commun. Syst., pp. 1–10, 2018.
295. Krishnan, "Machine Learning Based Intrusion Detection for Virtualized Infrastructures," 2018 Int. CET Conf. Control. Commun. Comput. (IC4); IEEE, pp. 366–371, 2018.
296. Y. C. Answers and I. Gloo, "Caffe2 vs pytorch," pp. 2020–2023, 2020.
297. Florencio, T. Valença, E. Moreno, and M. ColaçoJunior, "Performance analysis of deep learning libraries: Tensor flow and PyTorch," J. Comput. Sci., vol. 15, no. 6, 2019, doi: 10.3844/jcssp.2019.785.799.
298. Seide, "CNTK: Microsoft's Open-Source Deep-Learning Toolkit SPEAKER BIOGRAPHIES," p. 2945397, 2016, [Online]. Available: http://dx.doi.org/10.1145/2939672.2945397.
299. Semikron, "Technical Explanation SEMiX®5," pp. 1–50, 2016.

300. Z. Blumenfeld, “Scalable Distributed DNN Training Using Commodity GPU Cloud Computing,” J. Vitr. Fertil. Embryo Transf., vol. 8, no. 3, pp. 127–136, 2016, doi: 10.1007/BF01131701.
301. Y. Liu, C. Chen, R. Zhang, M. Research, H. Lin, and M. Yang, “Enhancing the Interoperability between Deep Learning Frameworks by Model Conversion Tingting Qin,” 2020, [Online]. Available: https://en.wikipedia.org/wiki/Interoperability.
302. B. WALLACH, “Developing,” A World Made Money, pp. 241–294, 2017, doi: 10.2307/j.ctt1d98bxx.10.
303. S. Shi, Q. Wang, P. Xu, and X. Chu, “Benchmarking state-of-the-art deep learning software tools,” Proc. - 7th Int. Conf. Cloud Comput. Big Data, CCBD 2017; IEEE IEEE Comput. Soc., vol. 7, pp. 99–104, 2017, doi: 10.1109/CCBD.2016.029.
304. Y. Wu, W. Cao, S. Sahin, and L. Liu, “Experimental Characterizations and Analysis of Deep Learning Frameworks,” Proc. - 2018 IEEE Int. Conf. Big Data, Big Data 2018, pp. 372–377, 2019, doi: 10.1109/BigData.2018.8621930.
305. [307] S. Shi, Q. Wang, and X. Chu, “Performance modeling and evaluation of distributed deep learning frameworks on GPUs,” Proc. - IEEE 16th Int. Conf. Dependable, Auton. Secur. Comput. IEEE 16th Int. Conf. Pervasive Intell. Comput. IEEE 4th Int. Conf. Big Data Intell. Comput. IEEE 3, pp. 943–948, 2018, doi: 10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.000-4.
306. Y. C. Answers and I. Gloo, “Caffe2 vs pytorch,” pp. 1–5, 2017.
307. T. Anuprathibha and C. S. K. Selvib, “Like It or Not: A Survey of Twitter Sentiment Analysis Methods,” IIOAB J., vol. 7, no. 9Special Issue, pp. 374–378, 2016.
308. S. Sakr, A. Shafaat, F. Bajaber, A. Barnawi, O. Batarfi, and A. Altalhi, “Liquid benchmarking: A platform for democratizing the performance evaluation process,” EDBT 2015 - 18th Int. Conf. Extending Database Technol. Proc., pp. 537–540, 2015, doi: 10.5441/002/edbt.2015.52.
309. Y. Wu et al., “A Comparative Measurement Study of Deep Learning as a Service Framework,” IEEE Trans. Serv. Comput., no. Dl, 2019, doi: 10.1109/TSC.2019.2928551.