Indonesian Journal of Computer Science
ISSN 2549-7286 (*online*)
Jln. Khatib Sulaiman Dalam No. 1, Padang, Indonesia
Website: ijcs.stmikindonesia.ac.id | E-mail: ijcs@stmikindonesia.ac.id

Distributed Systems for Machine Learning in Cloud Computing: A Review of Scalable and Efficient Training and Inference

Sheren Sadiq Hasan[1], Subhi R. M. Zeebaree[2]
sheren.hasan@dpu.edu.krd, subhi.rafeeq@dpu.edu.krd
[1]ITM Dept., Duhok Technical College, Duhok Polytechnic University, Duhok, Iraq
[2]Energy Eng. Dept., Technical College of Engineering, Duhok Polytechnic University, Duhok, Iraq

Article Information

Submitted : 7 Mar 2024
Reviewed: 17 Mar 2024
Accepted : 1 Apr 2024

Keywords

Distributed Systems, Machine Learning in Cloud Computing, Scalable Training, Efficient Training, Inference

Abstract

Traditional computer systems have been pushed to their limits as a result of the exponential rise of data and the rising complexity of machine learning (ML) models. As a result of its on-demand scalability and resource agility, cloud computing has emerged as the platform of choice for training and deploying large-scale machine learning models. However, in order to make good use of cloud resources for machine learning, it is necessary to make use of distributed systems. These systems are responsible for coordinating computations over several nodes in order to manage the demanding workloads. The purpose of this paper is to investigate the realm of distributed systems for machine learning in cloud computing, with a particular emphasis on training and inference that is both scalable and efficient. During the discussion on the need of distributed systems in machine learning, it was made clear why conventional single-machine techniques are not enough for the requirements of current machine learning and how distributed systems might help solve these difficulties. Scalability and Efficiency Considerations were reviewed in relation to the primary elements that contribute to the effectiveness of a distributed system for machine learning. These elements include task partitioning, communication overhead, fault tolerance, and resource optimization that were discussed. In the context of cloud computing, the purpose of this review research is to provide a complete overview of the fascinating topic of distributed systems for machine learning. In order to successfully traverse the intricate and ever-changing world of cloud-based machine learning, it provides vital insights and information.

A. Introduction

These days, data centers host a large number of machine learning workloads. For many of them, operating on a single node would probably take weeks or months [1]. As a result, they usually operate on a distributed, parallel platform [2]. Applying popular machine learning algorithms to large amounts of data raised new challenges for the ML practitioners. MapReduce paradigm is the most popular method for addressing scalability [3]. A practitioner can benefit from distributed machine learning frameworks like Spark ML, TensorFlow, and others. To speed up the creation of large models, distributed training across heterogeneous computer systems should develop exponentially. Mixture-of-Experts (MoE) models have been suggested as a practical approach to cut down on the expenses associated with training while taking into account the total amount of data and models [4] [5]. Within the context of a divide-and-conquer approach, this is accomplished via the use of gating and parallelism. A substantial amount of work has been put forward by Deep Speed in order to carry out thorough machine learning training on a variety of infrastructures. Nevertheless, there are still a number of system characteristics that have the potential to enhance the effectiveness of training and inference strategies. A few examples of these variables include the constraints placed on the amount of memory that may be used, the need for load balancing, and the optimization of the processes of communication and computing [6]. However, scaling a machine learning task is still quite trial and error. This occurs because parallelization of algorithms that were previously thought to be purely sequential is necessary for the scalability of machine learning algorithms, and practitioners are unsure of where to search for bottlenecks [7] [8].

Machine learning (ML) and deep learning (DL) systems require scalable and effective training and inference, particularly as models continue to increase in size and complexity[9] [10]. One example of scalable training is distributed training, which involves updating the model weights concurrently while distributing the training data among several devices or nodes. Model parallelism, on the other hand, describes the splitting of the model among devices or nodes and the execution of parallel computing for various model components [11] [12].

The three-dimensional training scale is one of the main reasons behind DL's effectiveness [13]. It is the size and complexity of the models themselves that serve as the major components, and they are the fundamental level of scale. Starting with neural networks that were simple and superficial, considerable progress was achieved in the direction of improving the accuracy of models via the use of model designs that were more elaborate and complicated [14]. This was accomplished by commencing with neural networks that were used. The use of neural networks was necessary in order to achieve this objective. Another aspect that plays a role in determining the scope of the experiment is the quantity of data that is included into the training process [15]. The accuracy of the model may be significantly improved by increasing the quantity of training data that it is exposed to. This can be accomplished by increasing the amount of data that the model goes through. With the help of a considerable quantity of training data, which may range anywhere from tens to hundreds of terabytes (TB), an artificial intelligence (DL) model is often taught in applications that are used in the real world. This kind of instruction is referred to as "deep learning." A third component that is taken into

consideration is the scale of the infrastructure [16]. This dimension is taken into account. Having access to technology that is both programmable and highly parallel is extremely necessary in order to train big models in an effective manner while making use of a significant amount of training data. This is an absolutely essential need. Graphics processing units, or GPUs, are the name given to this category of technology [11] [17].

Distributed systems play a crucial role in enabling scalable and efficient machine learning (ML) workflows in cloud computing environments [18]. This review will cover key aspects of distributed systems for machine learning, focusing on both training and inference phases.

B. Background Theory

1. Training in Distributed Systems

The problem of building big deep learning models with a substantial quantity of training data is a difficult one [19]. It is often carried out in a distributed architecture that is made up of a number of computer nodes, each of which may be equipped with several graphics processing units (GPUs). It is important to note that this presents a number of difficulties [20]. It is of the utmost importance to maximise the utilisation of processing resources, more especially to prevent the interruption of expensive GPU resources to the extent that communication limits are present. One of the defining characteristics of the cloud computing paradigm is the fact that the processing, storage, and network resources are often shared across several users or training procedures [20]. Consequently, this enables a higher level of efficiency and effectiveness. To guarantee that expenditures are decreased while still providing flexibility, this is done in order to give flexibility [21]. Whenever it comes to cloud computing, distributed systems for machine learning make use of a network of linked devices or nodes in order to carry out machine learning operations in a collaborative manner [13]. When it comes to efficiently managing enormous datasets, sophisticated models, and tasks that need a substantial amount of processing power, it is very necessary to make use of this method. The field of deep learning has a variety of various parallelization opportunity opportunities. This article will describe the three fundamental methods that may be used to parallelize deep learning. These methods are known as data parallelism, model parallelism, and pipeline parallelism. In addition, we will talk about both traditional and hybrid forms of parallel programming [11] [22]. one of the common considerations in the context of distributed systems for machine learning in the cloud is data parallelism which means divide the dataset across multiple nodes, and each node trains on a subset.it is Efficient use of resources, scales well with large datasets, but there are Communication overhead, synchronization issues [23].

2. Data Parallelism

Concurrent analysis of the data being collected. Through the use of data parallelism, a number of workers, including personal computers and graphics processing units, load replicates of the deep learning model that are similar to one

another (for a more in-depth explanation, please refer to Figure 1 for more reference). In order to make the process of training the worker model copy more manageable, the information that is utilized for the purpose of training the worker model copy is separated into portions that do not overlap with one another [24] [25]. Changes are made to the parameters of the model as a result of the process of each worker training independently on the subset of training data that has been assigned to them. In light of this, it is of the utmost importance to establish a synchronization between the parameters of the model. Among the members of the work force assemblage [11] [26].

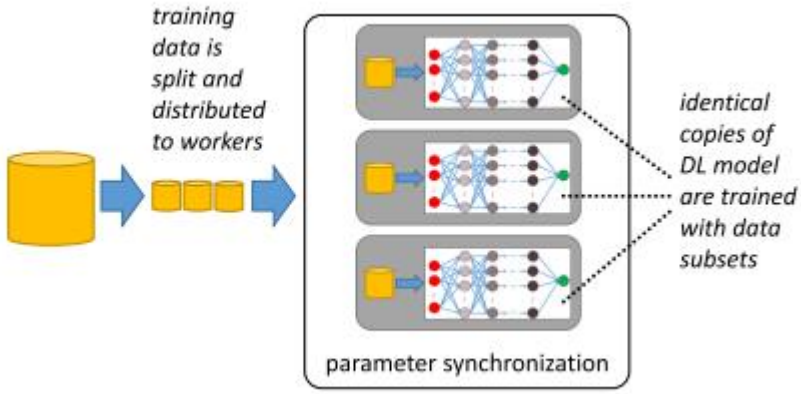

Figure 1. Data parallelism.

The primary benefit of data parallelism is that it can be used to any DL model architecture without the need for additional model domain expertise [27]. For computation-intensive tasks with few parameters, like CNNs, it scales well [28].

3. Model parallelism

If you wish to employ parallelism in models, you will need to partition the deep learning model into a large number of workers, each of whom will be responsible for training a distinct component of the model (for more information, refer to Figure 2 for more information) [29]. Personal training data is given to workers who are liable for the input layer of the DL model. These employees are the ones who get the training data that has been supplied to them. The computation of the output signal takes place during the forward pass, and after that, it is sent to the workers who are responsible for the subsequent layer of the deep learning model [30]. This occurs after the forward pass has successfully completed. During the backpropagation phase, gradients are computed, starting with the workers who are responsible for the output layer of the deep learning model and propagating towards the workers who are responsible for the input layers. This process is repeated until all of the workers have been calculated. This method will continue until it is completed, which will be till the gradients are computed [31]. One of the most important benefits of model parallelism is that it takes less memory than other methodological methods. This is one of the most significant advantages. As a result of the use of the split model, the amount of memory that is required by each worker is decreased to the fullest degree that is feasible. Because it takes into consideration both the size of the model and the size

of the device, this function is useful in situations when the model is either too huge or too little to fit on a single device. There is a good chance that this is the case, however it is contingent upon whether or not the device is outfitted with specialist hardware, such as a graphics processing unit (GPU) or a thermophoresis unit (TPU) [11] [32].

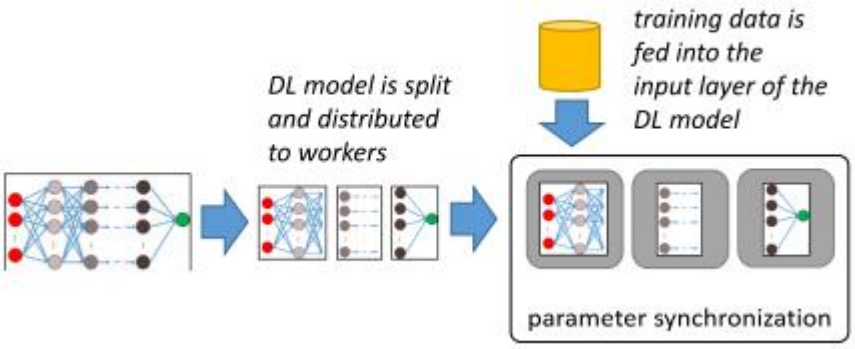

Figure 2. Models' parallelism.

4. Pipelines Parallelism

Data and model parallelism are combined in pipeline parallelism. Pipeline parallelism loads distinct portions of the DL model for training on separate workers, as seen in Figure 3 [33] [34]. Microbatches are used for the goal of making the training data more manageable by breaking it into smaller individual chunks. It is now the responsibility of each worker to compute output signals for a group of microbatches and to promptly send those signals to the workers that come after them in the line of work. At this time, this obligation is operating as intended. For the purpose of calculating the gradients for each partition of the model, the workers make use of a huge number of microbatches during the backpropagation phase of the process. The process is carried out in precisely the same manner as it was in the past. After that, they swiftly communicate these findings to the specialists that came before them in the process of putting together the team [35] [36]. By concurrently streaming a large number of micro batches via the forward and backpropagation pass, it is feasible to dramatically boost the utilisation of workers in comparison to pure model parallelism, which only processes one batch at a time. This is because pure model parallelism only processes one batch at a time. By using this method, it is possible to significantly expand the number of personnel that are used. One of the many major advantages that model parallelism offers is the elimination of the need for a single worker to store the whole model. This is only one of a long list of advantages. There are a great number of benefits that model parallelism offers, and this is only one of them [11] [33] [37].

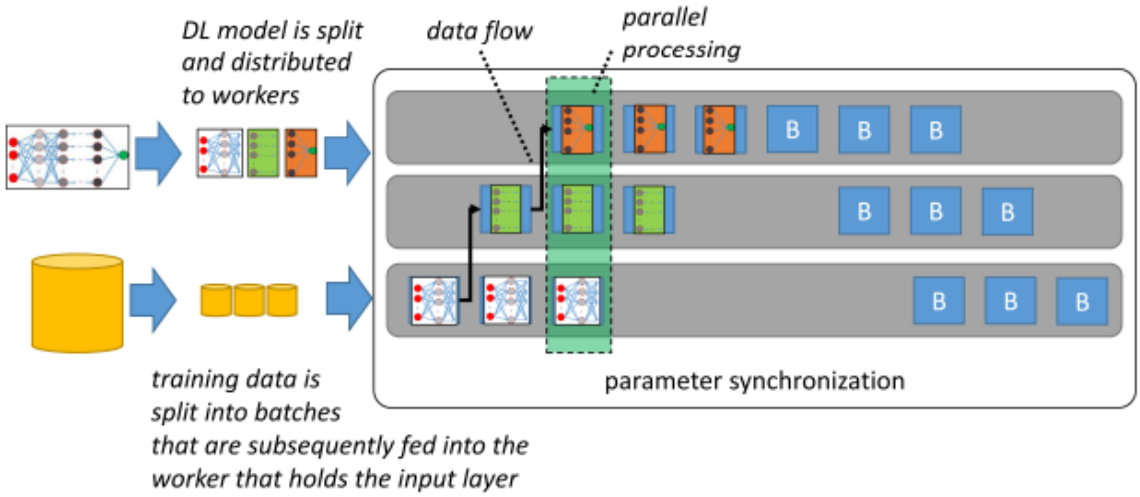

Figure 3. Pipeline parallelism.

5. Hybrid Parallelism

When dealing with deep learning models, it is essential to make use of a variety of parallelization strategies since these models are sometimes rather sophisticated. In most cases, these models are made up of several layers, each of which has a large lot of variation in its design [11].

C. Inference in Distributed Systems

In distributed systems, inference is the process of using a machine learning model on distributed and networked computer resources to make predictions or choices [38]. This might entail executing inference tasks concurrently across several nodes or devices in order to increase overall performance, reduce latency, and improve scalability [39]. The main aspect of inference in distributed systems is distributed inference which means running inference tasks on multiple nodes or devices concurrently [40]. The advantage of distributed inference is improved throughput, reduced latency, and efficient use of resources [41] [42]. the other aspect is Auto-scaling which refers to dynamically adjusting the number of resources based on the inference workload. it help to handle varying workloads [43]. The successful implementation of distributed systems for machine learning in cloud computing demands a Scalable and Efficient Training and Inference[44]. As a result, many researchers studied this field with its developing.

D. Literature Review

Researchers in [45] In order to facilitate the implementation of Human Resource Management (HRM) in Small and Medium Enterprises (SMEs), a cloud computing architecture that is completely unique was proposed. Furthermore, the practical applications of this architecture in the area of human resource management were shown. The use of cloud computing technologies is ultimately responsible for the improvement of the human resource management subsystems. While the researchers were in the process of putting the system into place, they

took into account the numerous challenges that may potentially arise. A number of issues, including the transfer of data, the training of users, and the modification of the system, were among the issues that they resolved. The proposed solutions that they provided were presented in order to solve these challenges. The findings of the study demonstrated that the utilisation of cloud technology enhances human resource management (HRM), which in turn enables small and medium-sized businesses (SMEs) to expand, effectively manage their workforce in a flexible manner, facilitate streamlined decision-making for managers, and ultimately leads to an increase in productivity. The fact that the study was carried out is evidence that this hypothesis is correct. Computing services that are hosted in the cloud are able to handle massive amounts of data, which helps them to improve their adaptability, scalability, cost-effectiveness, and efficiency. The term "cloud computing" may also refer to providers of cloud computing services.

Researchers in [46] An original approach that has been given the name AutoDeep has been introduced for the purpose of being used in deep neural network (DNN) inference tasks of a variety of different types. In order to reach the highest possible level of cost effectiveness, this technology is able to make a dynamic selection of the cloud configuration and device location that are the most suitable for the circumstances at hand. TensorFlow was used by the researchers in order to develop AutoDeep, and a significant amount of testing was performed on Microsoft Azure. Both of these methods were really carried out by the researchers who were accountable for carrying them out. When compared to non-trivial baselines such as Google's RL-based device placement technique and Lowest Cost First cloud architecture, AutoDeep displays significant gains in inference performance, search speed, and inference cost. These improvements are shown by AutoDeep. The baselines shown here are instances of baselines that are not inconsequential. Throughout its operations, AutoDeep displays these significant advancements. It is important to note that the conclusions that were discovered by these investigations are supported by research that was conducted with the aid of two DNN inference models that are frequently used. An effort was made by the researchers to automate the deployment of real-time cloud computing for online deep neural network (DNN) inference. This was done with the goal of achieving their aims of reducing costs and assuring an acceptable degree of latency. In the study, it is shown that the results on the scalability of the system in connection to a range of workloads and resource needs are supplied. This assertion is supported by the findings. These claims encompass a number of different things, one of which is the capability of the system to efficiently manage extended inference needs within the context of a live online service environment. In addition to this, they talk about the latency that is achieved for deep learning inference. In the article, this topic is examined in further detail.

Authors in [47] We investigated the principles of training deep learning models and investigated a variety of alternative approaches to divide these tasks across a cluster in order to simplify the process of training collaborative models. Our goal was to make the process of training collaborative models simpler. This investigation is being conducted with the intention of shedding light on the fundamental concepts that are involved in the process of training deep neural networks on a collection of computers that are independent of one another. The

purpose of this inquiry is to provide information in an effort to accomplish this goal. The dataset serves as an early signal within the context of the situation that has arisen. In order for data-parallel training to be as successful as it can possibly be, the dataset must be of a size that is sufficiently big to allow for the insertion of additional mini-batches that are sampled simultaneously. As a consequence of this, variation enhancement is achieved. As a consequence of their analysis, they came to the conclusion that it would be more advantageous to utilise dispersed training methodologies that would not make the complexity of the issue even more severe. This was the conclusion that they arrived to. Utilising simple model-parallel and synchronous data parallel approaches, which are superior to other methods, is the most effective way to take use of the additional resources that are offered by the cluster in order to increase the effective mini-batch size. This is because these techniques are superior to other similar methods. As a consequence of this, the variations in the gradients become less varied. Because of this, this is the consequence.

Authors in [48] It was suggested that a technique for distributed data aggregations that is capable of handling enormous data sets need to be something that is quick, adaptable, and all-encompassing. This was the recommendation that was made. Utilising a novel data structure that is referred to as the "Aggregation Tree," which enables efficient parallel processing of aggregation queries, is one way that this method may be put into action. Since this is the case, it is now feasible to implement the strategy that has been described. In addition, the study suggests a distributed method for the generation of the Aggregation Tree, as well as a query processing algorithm that makes use of the tree structure in order to reach the highest possible level of efficient performance. Both of these ideas are included in the research. These two approaches are discussed in the research that was conducted. During the course of their debate, the authors brought to light the difficulties that are connected to the management and examination of very large datasets. In order to effectively address these difficulties, it is often necessary to use data aggregation approaches that are not only efficient but also flexible. In terms of scalability and the amount of time it takes to perform queries, the recommended strategy was researched using both simulated and real-world datasets. The results of the investigation showed that it is superior to the methods that are now being used, which are currently being utilised. As a result of this, it is shown that it is not only effective but also efficient in the administration of operations that entail the gathering of extremely large amounts of data.

Authors in [49] Performance models were developed in order to assess a number of different distributed deep learning frameworks. The purpose of these models was to establish how effective the frameworks were utilising distributed deep learning. To be more specific, these models were developed in order to fulfil the need for increased processing capabilities and memory in the hardware. In order to achieve this objective, it was necessary to investigate a variety of various approaches to computing that were parallel and distributed. Data parallelism, model parallelism, pipeline parallelism, and hybrid parallelism were some of the tactics that were used. A significant portion of their work was directed on the challenges that are involved with training big and complicated deep learning models within a time window that is realistic. They give a knowledge that is

unparalleled in terms of the communication limitations that are associated with distributed training architectures, in addition to the significant impact that system design has on the performance of training environments. Additionally, they provide an introduction to the idea of distributed training structures during the course of the course. As a consequence of these findings, which will have significant ramifications, it is envisaged that they will be of tremendous aid in the process of creating and optimizing distributed deep learning systems.

Researchers on [50] For the purpose of delivering instruction via the internet, it has been proposed that a brand new top-k scarification communication library be constructed. This library displays an exceptional degree of efficiency in terms of both the processing and delivery of information. They increased the system's input/output speed by using a multi-level data caching method that was neither difficult nor wasteful. This enabled them to achieve their goal. The scalability of the operating system was improved by taking this measure, which was taken into consideration. Additionally, they built a proprietary parallel tensor operator, which enabled them to improve the process of updating. This was necessary in order to do this. The system outperforms other cutting-edge systems in terms of performance on CNNs and Transformer models, as shown by the results of testing that was carried out on a cluster of sixteen nodes that is part of Tencent Cloud. The margin of improvement is anywhere from 25 to 40 percent, which is an indication that the system is superior to other systems that are considered to be cutting edge. A total of eight graphics processing units (GPUs) from Nvidia Tesla V100 have been deployed in every single node of the cluster. Retraining ResNet-50 allowed the researchers to obtain a top-5 accuracy of 93% on the ImageNet dataset. This was accomplished by the researchers. When compared to the previous record on the DAWN Bench, which created a record for the greatest accuracy, this accuracy was higher than the previous record.

Authors in [51] An innovative variation of the distributed k-means algorithm has been proposed. A low-cost communication system and a paradigm for virtual parallel distributed computing are both included into a single machine via the implementation of this technique. In the context of a micro-services team that collaborates, the k-means algorithm is implemented as a distributed service via the use of an asynchronous communication mechanism that is founded on the AMQP protocol. This is the case when seen from the perspective of the micro-services team. We segment magnetic resonance imaging (MRI) pictures by constructing and executing a high-performance computing (HPC) programme that is both distributed and parallel. This is what we do in order to achieve this goal. The development of this software was done with the intention of making deployment on the cloud easier. The results of the trials demonstrate that the proposed method (DSCM) and the model that it provides in order to attain a high degree of scalability were effective in accomplishing the goals that were meant to be accomplished. In the future, we want to construct high-performance computing systems that are scalable and capable of grouping vast amounts of data. This will be accomplished via the use of our clustering technology.

Researchers in [52] The general public now has access to a cloud-based parallel computing system that employs a single-client multi-hash single-server multi-thread architecture. This system was made accessible to the public. In order

to properly manage parallel processing processes inside a cloud environment, the technology was developed expressly for that purpose throughout its development. Consequently, it was designed such that it could fulfil that particular function. It is now possible to conduct surveillance of expected outcomes in a manner that is far more efficient and effective than it was previously possible. This is due to the fact that both the quantity of data that is available and the amount of time that is required for processing have significantly increased. It was shown that the proposed solution demonstrated improved efficiency in terms of Kernel-burst, User-burst, and Total-execution durations, all while running with little load (single hash-code) across a number of servers and processors. This was demonstrated by the fact that the system was able to maintain its efficiency. One way in which this was proved was by the fact that the remedy was proposed. This is supported by evidence that can be found.

Authors in [53] presented Mixture-of-Experts (MoE) models which make use of divide-and-conquer strategies that make use of gating and parallelism in order to lower the costs associated with training while taking into account the total number of models and data that are involved. This is done in order to achieve that goal. A proposal known as SE-MoE has been put forth by researchers. This proposal promotes Elastic MoE training using 2D prefetch and Fusion communication that makes use of Hierarchical storage. This is done with the purpose of maximising the use of various parallelisms in order to get optimal results. When the size of the models is more than the amount of GPU RAM that is available, SE-MoE employs a collaborative method to load the models. This is done in circumstances when larger models are loaded. In order to simplify the loading process, it organises the memory of the central processing unit (CPU) and graphics processing unit (GPU) into a structure that resembles a ring. Following this, it ensures that the computational workloads are distributed across the memory partitions in a round-robin method in order to ensure that the inference process is carried out effectively. As a consequence of this, it is now feasible to carry out inference operations on a single computing device in an efficient way, while at the same time maintaining the capability to handle larger workloads. In terms of throughput, SE-MoE was able to outperform DeepSpeed thanks to a 33% boost in training and an overall 13% gain in inference. In terms of performance, DeepSpeed was exceptional. This advantage, which was notably obvious in the Ministry of Education Tasks that were not balanced, was particularly noticeable.

Researchers on [54] In order to support the smooth integration and removal of new staff members without creating any disturbance to the training process that is already in place, it was suggested that a dynamic scaling approach be used. This would make it feasible to facilitate the seamless incorporation and removal of team members. In addition to this, they provide a method that is resistant to stragglers and one that incorporates an awareness of the heterogeneity of the data. When the performance of cloud GPUs is unknown, this technique provides an increase in performance by delivering more resources to the system. This is accomplished by ensuring that the system receives additional resources. When compared to the checkpoint-based system that had been in place up until that point in time, the solution that was offered resulted in a considerable increase in throughput. An increase in throughput that was 17.52 times greater than it had been in the past

was achieved as a consequence of the approach that was suggested. The current cluster of five workers was expanded to a total of ten workers in order to accomplish this desired result. In the framework known as Elastic Horovod, which is capable of dynamic scaling, the training process was abruptly paused for a duration of 841 seconds, and then it was resumed at a level of performance that was 95.52% of its maximum level. There was a momentary disruption to the structure, which led to this occurrence. This occurrence took happened after an extra 841 seconds had passed since the last one. When GPUs with varying capacities were combined, it was found that the average difference between the estimated batch size and the optimal batch size was 3.37 percent. This was the case even when GPUs with different capabilities were combined. 3.37 percent was determined to be the optimal batch size, according to the final finding.

Authors in [55] As a unique approach to scaling deep learning models to trillion parameter models, the use of sparsity approaches that are not only easy but also efficient was proposed as a strategy. The Switch Transformer is a specialised Transformer model that was provided by the authors. This model makes use of a mixture of experts (MoE) to choose one-of-a-kind parameters for each input. The authors provided a description of this concept in the presentation that they gave. This results in a model that is only partly active and has a substantial number of parameters. As a consequence of this, the model is generated. For the purpose of demonstrating their empirical results about the pre-training of models that had as many as one trillion parameters, the researchers made use of a large dataset. When compared to prior models, they were able to accomplish a large improvement in pre-training time, with an increase that was up to seven times higher than what was previously achieved. A considerable number of discoveries were made concerning the enhancement of the scalability of deep learning models during the course of the research endeavour that was carried out. In order to make effective use of vast amounts of computer resources for deep learning projects, this is an essential component that must be present.

Authors in [56] In order to accomplish the goal of training deep learning models in a collaborative manner across a number of remote clusters that have a variety of hardware and wide area networks (WANs) with limited bandwidth, the Nebula-I framework was developed. The training of deep learning models is the primary objective of this framework, which is an all-encompassing solution that was developed this purpose. Using parameter-efficient training processes, hybrid parallel computing approaches, and adaptive communication acceleration techniques, the framework is able to strike a balance between the accuracy of its communication and the efficiency with which it communicates. This allows the framework to ensure that its communication is both accurate and efficient. This enables the framework to strike a balance between the two factors that are being considered. The framework for deep learning known as PaddlePaddle is employed in the construction of Nebula-I, which is a collaborative training system that is utilised for the purpose of training. As a consequence of the qualities that it has, it is able to provide training across a wide variety of hardware, including GPU and NPU, among other types of hardware. Through the execution of tests, namely in the domain of natural language processing (NLP) activities, the research demonstrates that Nebula-I is effective in accomplishing the goals that it was designed to achieve.

The creation of multilingual language models and the improvement of machine translation models are two examples of the kinds of endeavours that are included in this category of work.

Authors in [57] MiCS is a solution that was designed primarily with the idea of lowering the amount of communication overhead and offering virtually linear scalability for the purpose of training big models on clusters in public cloud settings. This was done in order to achieve the specific goal of training gigantic models. These are some examples that illustrate it. By making efficient use of a wide range of network resources, MiCS is able to eliminate communication delays, minimise network congestion on slower connections, and lessen the total amount of congestion on the network. Consequently, this results in scalability that is much more efficient than linear in performance and is almost linear in terms of efficiency. It is possible for MiCS to accomplish these goals on account of the decrease in the number of individuals who participate in communication groups. It is shown that the framework is capable of making full use of theoretical computing resources, in addition to demonstrating a considerable increase in system performance. In addition to this, it brings about the possibility of training unique models on public cloud clusters that include billions of parameters. A demonstration of both of these capabilities is provided by the framework. MiCS makes a major addition to the area of distributed machine learning and cloud computing because it offers a solution to the challenges of efficiently expanding the size of model training in large-scale cloud settings with a range of networking situations. This is one of the reasons why MiCS is so important.

Authors in [58] Using the resources that are typically associated with networking, the system known as Varuna was built with the purpose of easing the process of training deep learning models on a wide scale. This was accomplished by utilising the resources that are typically associated with networking. It is possible for Varuna to make use of "low-priority" virtual machines (VMs), which are about five times more affordable than graphics processing units (GPUs) that are specifically designed for graphics processing. This is as a result of the fact that it makes the most effective use of the resources available for networking and automatically adapts the training setting for the user. It is a direct consequence of this that the cost of training large-scale models is significantly reduced, which is a consequence of this. By training large-scale models, such as a model with 200 billion parameters, utilising "spot VMs," which are five times less costly than standard virtual machines (VMs) while still attaining outstanding training speed, the success of the approach may be shown. It is possible to demonstrate the effectiveness of the method by training large-scale models. Since "spot VMs" are capable of achieving high training speed, this is something that can be accomplished. Varuna has the potential to dramatically reduce the amount of time required for training language models such as BERT and GPT-2 by up to 18 times when compared to the model-parallel procedures that were previously used. This is a considerable improvement over the prior methods. On GitHub, which is the website where the source code for Varuna is hosted, it is possible to locate the code in its original form. Within the scope of this essay, one of the most important contributions that Varuna has made to the field of distributed machine learning and cloud computing is described. By concentrating on the efficient training of

large-scale deep learning models on public cloud clusters, it makes an effort to address the challenges that are connected with this objective. This is done within the framework of this ambition.

Authors in [59] The presentation of a novel network interface card (NIC) that makes use of field-programmable gate arrays (FPGAs) was made for the purpose of using it in artificial intelligence training systems that are situated in remote places. Research of the additional expenses that are connected with collective communication activities is carried out by the authors in the context of distributed artificial intelligence training systems. They are responsible for the design and implementation of a technologically advanced smart network interface card (NIC). This card was developed with the intention of removing the responsibility of communication from the computing resources of the system. It is possible for the computing resources of the system to direct their attention to activities that need a larger amount of processing power if this is done. The goal of this inquiry is to provide empirical data that was gathered from a prototype distributed artificial intelligence training system that consisted of six compute nodes. This data was obtained for the purpose of this investigation. In comparison to the baseline system, which makes use of traditional NICs, the data indicate that there is a considerable improvement in the overall training performance that is 1.6 times in magnitude. The basic system, on the other hand, makes use of traditional network interface controllers (NICs). The authors have created a prediction model for the performance of the artificial intelligence smart network interface (NIC) that is based on FPGA technology. Authors are responsible for the creation of this model. Their assessment reveals that there would be a performance improvement that is 2.5 times larger than what was previously noticed with the deployment of 32 nodes. This suggests that the performance boost would be significant.

Researchers in [60] In order to improve the effectiveness and efficiency of deep learning, I have meticulously investigated artificial intelligence frameworks that are distributed and scalable, and that make use of cloud computing. This was done with the intention of boosting the efficacy and efficiency of deep learning. The topics that were discussed included a wide variety of topics, including an introduction to AI frameworks and cloud services that are commonly used, the management and organisation of data in AI systems that are based in the cloud, the cost-effectiveness of AI solutions in the cloud, the deployment and utilisation of AI models in the cloud, numerous methods for distributed and parallel training, and various strategies for optimising AI workloads in the cloud. For the purpose of ensuring that cloud-based artificial intelligence technologies are cost-effective, the study carried out an exhaustive investigation into the costs, developed methods for achieving the highest possible degree of efficiency, and provided examples of successful implementations. The goal of the research was to test whether or not these technologies are capable of performing the tasks for which they were designed. After doing study, the researchers arrived at the opinion that artificial intelligence models might be deployed as distinct services that are only weakly coupled to one another as part of a microservices architecture. This was the conclusion that stemmed from their findings. It is possible to carry out the building, expansion, and upgrading of these services on an individual basis. When it comes to the deployment and maintenance of artificial intelligence models in

production environments, this technique allows greater flexibility, agility, and scalability at the same time. In addition to this, it paves the way for potential that include enhanced scalability.

E. Discussion and Comparison

The integration of distributed deep learning and cloud computing has become a promising area of research. The use of distributed machine learning and deep learning techniques allows for the training of large-scale models over vast volumes of data, which is not feasible on a single machine. This study involve compares among different papers. The papers listed cover a range of topics related to distributed deep learning and cloud computing. The approach involves dividing the learning process across multiple workstations, achieving scalability of learning algorithms. Furthermore, the use of cloud or server-based distributed machine learning allows for the collection of combined models from multiple participants, with each training their own model locally. Recent research has focused on developing scalable, distributed AI frameworks that leverage cloud computing for enhanced deep learning performance and efficiency. Additionally, new techniques and systems have been proposed to enable scalable distributed training of deep learning models on public cloud clusters, addressing challenges related to inter-connection bandwidth and system scalability. These advancements are crucial for handling naturally distributed datasets, which are common in real-world applications. The proposed solutions aim to optimize communication efficiency, I/O, and system scalability, ultimately breaking records in training large-scale models. Overall, the research in this area is driving the development of more intelligent systems by leveraging the capabilities of distributed deep learning and cloud computing. As shown in Table 1. Which illustrate the comparison among different studies related to distributed machine learning and cloud computing resources for training large-scale models.

Table 1. Summary of literature review related to distributed machine learning and cloud computing resources for training large-scale models.

Number /year	Short description	Advantages	Disadvantages
[45] 2020	• A Human Resource Management (HRM) system that is developed specifically to fulfil the requirements of small and medium-sized organisations (SMEs) and is hosted in the cloud.	Improved HRM, Increased flexibility, Reduced costs, Enhanced decision-making	Security concerns
[46] 2020	• The process of automating the deployment of cloud	Automation, Real-	Potential complexity,

Number /year	Short description	Advantages	Disadvantages
	infrastructure for the purpose of carrying out deep learning inference on real-time web services is now under consideration.	time inference	Dependency on cloud infrastructure
[47] 2020	• An analysis of the many methods of distributed training for deep learning models, organised according to their taxonomic classification	Scalability, Comprehensive analysis	Overhead in coordination, Communication costs
[48] 2020	• A method that is capable of acquiring data from a variety of sources, with the ability to swiftly scale up and adapt to changing circumstances.	Speed, Scalability	Data consistency, Network overhead
[49] 2020	• In distributed systems, an evaluation of the effectiveness and scalability of deep learning algorithms is required.	Improved performance, Scalability	Communication overhead, Complexity
[50] 2021	• On a broad scale, enabling the training of deep learning models on public cloud clusters in a manner that is both efficient and effective.	Scalability, Utilization of cloud resources	Network latency, Cost
[51] 2021	• In order to improve the capabilities of high-performance computing, a new distributed k-means algorithm that is scalable is presented along with the utilisation of cloud micro-services.	Scalability, High-performance computing	Dependency on micro-services, Overhead
[52] 2021	• A cloud-based parallel computing system	Parallel processing, Cloud-	Complexity, Resource

Number /year	Short description	Advantages	Disadvantages
	incorporates a single-client multi-hash single-server multi-thread architecture. This architecture is employed in the construction of the system.	based	management
[53] 2022	• The Se-moe system is a distributed training and inference system that is both scalable and economical. It employs a mixture-of-experts technique to improve its performance.	Scalability, Efficiency	Complexity, Coordination overhead
[54] 2022	• A methodology for training deep neural networks that is scalable and geared for use in a GPU cloud environment that is heterogeneous is called Scale-Train.	Scalability, Heterogeneous training	Dependency on GPU, Resource allocation
[55] 2022	• The use of switch transformers enables the utilisation of trillion parameter models via the utilisation of sparsity approaches that are both simple and efficient.	Scalability, Efficiency	Complexity, Model sparsity
[56] 2022	• The Nebula-I framework is an all-encompassing system that was developed with the intention of facilitating the collaborative training of deep learning models on cloud clusters that have restricted bandwidth.	Collaboration, Low-bandwidth support	Coordination overhead, Bandwidth limitation
[57] 2022	• Accomplishing near-linear scalability for the purpose of training extremely large models on public cloud systems is the focus of MiCS.	Scalability, Linear scaling	Cost, Resource allocation

Number /year	Short description	Advantages	Disadvantages
[58] 2022	• The training of large-scale deep learning models may be accomplished in a manner that is both efficient and cost-effective with the help of Varuna.	Scalability, Cost-effectiveness	Resource allocation, Coordination
[59] 2022	• Network interface cards (NICs) that are driven by FPGAs and are meant for artificial intelligence training systems that are dispersed and scalable.	Scalability, FPGA utilization	Hardware dependency, Cost
[60] 2023	• Cloud computing has the potential to be employed in order to enhance the performance and efficiency of deep learning. This may be accomplished via the utilisation of artificial intelligence frameworks that are modular and distributed.	Scalability, Enhanced performance	Resource allocation, Cost

Extracted Statistics

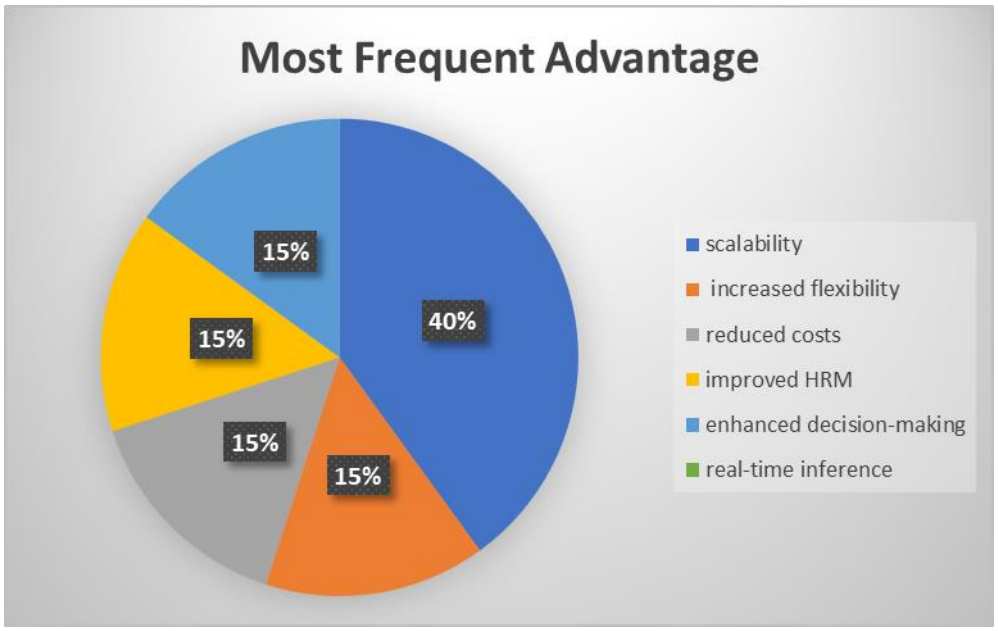

Figure 4. Most frequent advantages depended by the researchers

Based on the provided search results, the advantages can be categorized into several groups, such as scalability, performance enhancement, cost reduction, and flexibility. Results shows that the most frequent advantages are related to scalability (8 times), followed by improved HRM, increased flexibility, reduced costs, and enhanced decision-making (3 times each). The other advantages appear less frequently, with automation, real-time inference, and collaboration appearing twice, and the rest appearing only once data. The plot shows that the majority of the papers focus on scalability, which is a key advantage of cloud computing and distributed systems. Scalability allows for the efficient use of resources and the ability to handle large amounts of data. Additionally, many papers highlight the benefits of parallel processing, which can improve performance and reduce costs. Cloud-based solutions also offer increased flexibility and enhanced decision-making capabilities. Real-time inference and automation are also important advantages, as they allow for faster and more efficient processing of data. Finally, some papers emphasize the utilization of FPGA for improved performance and efficiency.

F. Recommendations

The objective of this section is to provide an overview of a useful cloud computing reference. Additionally, discover the benefits and drawbacks of a review paper. Based on the search results, some of the studies provided insights into device placement optimization and its impact on inference cost. For example, "Device Placement Optimization with Reinforcement Learning" by Pham et al. presented a method that learns to optimize device placement for TensorFlow computational graphs using a sequence-to-sequence model to predict subsets of operations and their execution time as a reward signal. "Efficient Algorithms for

Device Placement of DNN Graph Operators" by Gao et al. addressed the optimization problem at the core of device placement for both inference and training of deep neural networks. These resources provided insights into the implementation and scalability of device placement optimization.

Adding to that, it is recommended to depend on “Cloud computing offers scalability and agility, making it ideal for large-scale Machine Learning (ML)” instead of the traditional systems struggle with the immense data and complex models of (ML). However, the distributed systems are crucial to manage demanding workloads by coordinating computations across multiple machines. This paper explores distributed systems for efficient and scalable ML in cloud environments, focusing on overcoming limitations of single-machine approaches and highlighting factors like task division, communication, fault tolerance, and resource optimization for effective cloud-based ML.

G. Conclusion

The development of applications that are capable of machine learning is becoming more dependent on the use of cloud resources for distributed training, which is becoming an increasingly crucial component. This is because the quantity of data that is utilized for training is growing at an exponential rate, and neural networks are becoming more complex. Both of these factors are contributing to increasing complexity. This is something that has come about as a consequence of the growing amount of data that is being used for the purpose of training activities. The distributed training frameworks that are now being used are unable to scale out their training clusters without causing the training process that is currently being carried out to come to a stop. This is because scaling out their training clusters is not feasible. Despite the fact that cloud training clusters are expected to have elastic scalability, this is the situation that has arisen. For the purpose of adding insult to injury, as a direct consequence of this, it is more difficult for dispersed training frameworks to make effective use of the resources that are now accessible via the cloud. Throughout the course of this inquiry, we have covered a broad variety of topics pertaining to the components of distributed systems for machine learning in cloud computing. Every one of these constituents is discussed in this section. Training and inference that is not just effective but also scalable are two of the features that fall under this category. [19]

H. References

[1] Malallah, H., et al., *A comprehensive study of kernel (issues and concepts) in different operating systems.* Asian Journal of Research in Computer Science, 2021. **8**(3): p. 16-31.

[2] Zeebaree, S. and K. Jacksi, *Effects of processes forcing on CPU and total execution-time using multiprocessor shared memory system.* Int. J. Comput. Eng. Res. Trends, 2015. **2**(4): p. 275-279.

[3] Pop, D., *Machine learning and cloud computing: Survey of distributed and saas solutions.* arXiv preprint arXiv:1603.08767, 2016.

[4] Haji, L.M., et al., *A State of Art Survey for OS Performance Improvement.* Science Journal of University of Zakho, 2018. **6**(3): p. 118-123.

[5] Ibrahem, A.H. and S.R. Zeebaree, *Tackling the Challenges of Distributed Data Management in Cloud Computing-A Review of Approaches and Solutions.* International Journal of Intelligent Systems and Applications in Engineering, 2024. **12**(15s): p. 340-355.

[6] Sadeeq, M.A. and S.R. Zeebaree, *Design and implementation of an energy management system based on distributed IoT.* Computers and Electrical Engineering, 2023. **109**: p. 108775.

[7] Ström, N., *Scalable distributed DNN training using commodity GPU cloud computing.* 2015.

[8] Jghef, Y.S., et al., *Bio-Inspired Dynamic Trust and Congestion-Aware Zone-Based Secured Internet of Drone Things (SIoDT).* Drones, 2022. **6**(11): p. 337.

[9] Osanaiye, B.S., et al., *Network data analyser and support vector machine for network intrusion detection of attack type.* REVISTA AUS, 2019. **26**(1): p. 91-104.

[10] Zeebaree, S.R., et al., *Multicomputer multicore system influence on maximum multi-processes execution time.* TEST Engineering & Management, 2020. **83**(03): p. 14921-14931.

[11] Mayer, R. and H.-A. Jacobsen, *Scalable deep learning on distributed infrastructures: Challenges, techniques, and tools.* ACM Computing Surveys (CSUR), 2020. **53**(1): p. 1-37.

[12] Haji, L.M., et al., *Dynamic resource allocation for distributed systems and cloud computing.* TEST Engineering & Management, 2020. **83**(May/June 2020): p. 22417-22426.

[13] Mohammed, S.M., K. Jacksi, and S. Zeebaree, *A state-of-the-art survey on semantic similarity for document clustering using GloVe and density-based algorithms.* Indonesian Journal of Electrical Engineering and Computer Science, 2021. **22**(1): p. 552-562.

[14] Ageed, Z.S. and S.R. Zeebaree, *Distributed Systems Meet Cloud Computing: A Review of Convergence and Integration.* International Journal of Intelligent Systems and Applications in Engineering, 2024. **12**(11s): p. 469-490.

[15] Khalid, Z.M. and S.R. Zeebaree, *Big data analysis for data visualization: A review.* International Journal of Science and Business, 2021. **5**(2): p. 64-75.

[16] Ibrahim, I.M., et al. *Hybrid Client/Server Peer to Peer Multitier Video Streaming*. in *2021 International Conference on Advanced Computer Applications (ACA)*. 2021. IEEE.

[17] Zebari, I.M., S.R. Zeebaree, and H.M. Yasin. *Real time video streaming from multi-source using client-server for video distribution*. in *2019 4th Scientific International Conference Najaf (SICN)*. 2019. IEEE.

[18] Jghef, Y.S. and S. Zeebaree, *State of art survey for significant relations between cloud computing and distributed computing.* International Journal of Science and Business, 2020. **4**(12): p. 53-61.

[19] Zeebaree, S., N. Cavus, and D. Zebari, *Digital Logic Circuits Reduction: A Binary Decision Diagram Based Approach.* LAP LAMBERT Academic Publishing, 2016.

[20] Fawzia, H., et al., *A REVIEW OF AUTOMATED DECISION SUPPORT TECHNIQUES FOR IMPROVING TILLAGE OPERATIONS.* REVISTA AUS, 2019. **26**: p. 219-240.

[21] Zhou, L., et al., *Machine learning on big data: Opportunities and challenges.* Neurocomputing, 2017. **237**: p. 350-361.

[22] Abdullah, H.S. and S.R. Zeebaree, *Distributed Algorithms for Large-Scale Computing in Cloud Environments: A Review of Parallel and Distributed Processing.* International Journal of Intelligent Systems and Applications in Engineering, 2024. **12**(15s): p. 356-365.

[23] Rashid, Z.N., et al. *Distributed cloud computing and distributed parallel computing: A review*. in *2018 International Conference on Advanced Science and Engineering (ICOASE)*. 2018. IEEE.

[24] Zangana, H.M. and S.R. Zeebaree, *Distributed Systems for Artificial Intelligence in Cloud Computing: A Review of AI-Powered Applications and Services.* International Journal of Informatics, Information System and Computer Engineering (INJIISCOM), 2024. **5**(1): p. 1-20.

[25] Zeebaree, S. and I. Zebari, *Multilevel client/server peer-to-peer video broadcasting system.* International Journal of Scientific & Engineering Research, 2014. **5**(8): p. 260-265.

[26] Shukur, H., et al., *Cloud computing virtualization of resources allocation for distributed systems.* Journal of Applied Science and Technology Trends, 2020. **1**(3): p. 98-105.

[27] Shukur, H., et al., *Cache coherence protocols in distributed systems.* Journal of Applied Science and Technology Trends, 2020. **1**(3): p. 92-97.

[28] Xing, E.P., et al. *Petuum: A new platform for distributed machine learning on big data*. in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2015.

[29] Sami, T.M.G., S.R. Zeebaree, and S.H. Ahmed, *A Comprehensive Review of Hashing Algorithm Optimization for IoT Devices.* International Journal of Intelligent Systems and Applications in Engineering, 2023. **11**(6s): p. 205-231.

[30] Abdulkhaleq, I.S. and S. Zeebaree, *Science and Business.* International Journal. **5**(3): p. 126-136.

I. S. Abdulkhaleq, & S. R. Zeebaree. "State of Art for Distributed Databases: Faster Data Access, processing, Growth Facilitation and Improved Communications", International Journal of Science and Business, vol. 5(3), pp. 126-136, 2021.

[31] Kako, N.A., *DDLS: Distributed Deep Learning Systems: A Review.* Turkish Journal of Computer and Mathematics Education (TURCOMAT), 2021. **12**(10): p. 7395-7407.

[32] Castelló, A., et al. *Analysis of model parallelism for distributed neural networks*. in *Proceedings of the 26th European MPI Users' Group Meeting*. 2019.

[33] Huang, Y., et al., *Gpipe: Efficient training of giant neural networks using pipeline parallelism.* Advances in neural information processing systems, 2019. **32**.

[34] Zeebaree, S.R., et al. *Design and simulation of high-speed parallel/sequential simplified DES code breaking based on FPGA*. in *2019 International Conference on Advanced Science and Engineering (ICOASE)*. 2019. IEEE.

[35] Salih, A., et al. *A survey on the role of artificial intelligence, machine learning and deep learning for cybersecurity attack detection*. in *2021 7th International Engineering Conference "Research & Innovation amid Global Pandemic"(IEC)*. 2021. IEEE.

[36] Zeebaree, S., *DES encryption and decryption algorithm implementation based on FPGA.* Indones. J. Electr. Eng. Comput. Sci, 2020. **18**(2): p. 774-781.

[37] Li, Z., et al. *Terapipe: Token-level pipeline parallelism for training large-scale language models*. in *International Conference on Machine Learning*. 2021. PMLR.

[38] Shukur, H., et al., *A state of art survey for concurrent computation and clustering of parallel computing for distributed systems.* Journal of Applied Science and Technology Trends, 2020. **1**(4): p. 148-154.

[39] Jebali, A., S. Sassi, and A. Jemai. *Inference control in distributed environment: a comparison study*. in *Risks and Security of Internet and Systems: 14th International Conference, CRiSIS 2019, Hammamet, Tunisia, October 29–31, 2019, Proceedings 14*. 2020. Springer.

[40] Abdullah, P.Y., et al., *An hrm system for small and medium enterprises (sme) s based on cloud computing technology.* International Journal of Research-GRANTHAALAYAH, 2020. **8**(8): p. 56-64.

[41] Gao, Y., et al., *A review of distributed statistical inference.* Statistical Theory and Related Fields, 2022. **6**(2): p. 89-99.

[42] Shukur, H.M., et al. *Design and implementation of electronic enterprise university human resource management system*. in *Journal of Physics: Conference Series*. 2021. IOP Publishing.

[43] Mohammed, T., et al. *Distributed inference acceleration with adaptive DNN partitioning and offloading*. in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. 2020. IEEE.

[44] Hasan, D.A., et al., *The impact of test case generation methods on the software performance: A review.* International Journal of Science and Business, 2021. **5**(6): p. 33-44.

[45] Abdullah, P.Y., et al., *HRM system using cloud computing for Small and Medium Enterprises (SMEs).* Technology Reports of Kansai University, 2020. **62**(04): p. 04.

[46] Li, Y., et al. *Automating cloud deployment for deep learning inference of real-time online services*. in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. 2020. IEEE.

[47] Langer, M., et al., *Distributed training of deep learning models: A taxonomic perspective.* IEEE Transactions on Parallel and Distributed Systems, 2020. **31**(12): p. 2802-2818.

[48] Perera, N., et al. *A fast, scalable, universal approach for distributed data aggregations*. in *2020 IEEE International Conference on Big Data (Big Data)*. 2020. IEEE.

[49] Mahon, S., et al. *Performance analysis of distributed and scalable deep learning*. in *2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*. 2020. IEEE.

[50] Shi, S., et al., *Towards scalable distributed training of deep learning on public cloud clusters.* Proceedings of Machine Learning and Systems, 2021. **3**: p. 401-412.
[51] Benchara, F.Z. and M. Youssfi, *A new scalable distributed k-means algorithm based on Cloud micro-services for High-performance computing.* Parallel Computing, 2021. **101**: p. 102736.
[52] Rashid, Z.N., et al. *Cloud-based Parallel Computing System Via Single-Client Multi-Hash Single-Server Multi-Thread*. in *2021 International Conference on Advance of Sustainable Engineering and its Application (ICASEA)*. 2021. IEEE.
[53] Shen, L., et al., *Se-moe: A scalable and efficient mixture-of-experts distributed training and inference system.* arXiv preprint arXiv:2205.10034, 2022.
[54] Kim, K., et al., *Scale-Train: A Scalable DNN Training Framework for a Heterogeneous GPU Cloud.* IEEE Access, 2022. **10**: p. 68468-68481.
[55] Fedus, W., B. Zoph, and N. Shazeer, *Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity.* The Journal of Machine Learning Research, 2022. **23**(1): p. 5232-5270.
[56] Xiang, Y., et al., *Nebula-I: A general framework for collaboratively training deep learning models on low-bandwidth cloud clusters.* arXiv preprint arXiv:2205.09470, 2022.
[57] Zhang, Z., et al., *MiCS: Near-linear scaling for training gigantic model on public cloud.* arXiv preprint arXiv:2205.00119, 2022.
[58] Athlur, S., et al. *Varuna: scalable, low-cost training of massive deep learning models*. in *Proceedings of the Seventeenth European Conference on Computer Systems*. 2022.
[59] Ma, R., et al., *FPGA-based AI smart NICs for scalable distributed AI training systems.* IEEE Computer Architecture Letters, 2022. **21**(2): p. 49-52.
[60] Mungoli, N., *Scalable, Distributed AI Frameworks: Leveraging Cloud Computing for Enhanced Deep Learning Performance and Efficiency.* arXiv preprint arXiv:2304.13738, 2023.