Comparison of Benchmarks for Machine Learning Cloud Infrastructures

Manav Madan*, Christoph Reich*
Institute for Data Science, Cloud Computing and IT-Security (IDACUS)
Furtwangen University of Applied Science
Furtwangen, Germany
e-mail: {manav.madan, christoph.reich}@hs-furtwangen.de

***Abstract*—Training of neural networks requires often high computational power and large memory on Graphics Processing Unit (GPU) hardware. Many cloud providers such as Amazon, Azure, Google, Siemens, etc, provide such infrastructure. However, should one choose a cloud infrastructure or an on-premise system for a neural network application, how can these systems be compared with one another? This paper investigates seven prominent Machine Learning benchmarks, which are MLPerf, DAWNBench, DeepBench, DLBS, TBD, AIBench, and ADABench. The recent popularity and widespread use of Deep Learning in various applications have created a need for benchmarking in this field. This paper shows that these application domains need slightly different resources and argue that there is no standard benchmark suite available that addresses these different application needs. We compare these benchmarks and summarize benchmark-related datasets, domains, and metrics. Finally, a concept of an ideal benchmark is sketched.**

***Index Terms*—Machine Learning, Machine Learning Benchmark, MLPerf, AIBench, Deep learning, Survey**

I. INTRODUCTION

Training of neural networks requires high computational power and large memory. Graphics Processing Units (GPUs) can significantly speed up the training process for many Deep Learning models. Training models for tasks such as Image classification, Video analysis, and Natural language processing involve computationally intensive matrix multiplications and other operations that can take advantage of a GPU's massively parallel architecture. It can take days to train a Deep Learning model that performs intensive computational tasks with large datasets on a single processor. However, if the program is designed to transfer these tasks to one or more GPUs then the training time is reduced to a few hours instead of a few days. Many cloud providers such as Amazon, Azure, Google, Siemens, etc, are providing such infrastructures. These hardware resources vary in terms of memory, storage, and processing power capacity. On these cloud platforms, one can acquire the required resources. The question is, which fits best to the specific machine learning application. Benchmarks can help to compare these cloud infrastructures.

A benchmark is defined as either an individual program or a set of programs that measure systems performance with respect to a reference [1]. In order to use a benchmark, one has to run the individual program or the set of programs on the target machine which would generate a report characterizing the performance of the System Under Test (SUT). In terms of a computer, this performance could be related to I/O processing, running a graphics application, solving some linear equations, etc. A benchmark usually consists of four parts, which are scenario, evaluation criteria, evaluation metrics, and benchmarking score [1]. The scenario provides a detailed description of the setup environment. Evaluation criteria define important rules that specify the requirements which should be met to use the benchmark successfully. A metric quantifies a specific quality of the SUT which is the focus of the benchmark. Finally, the benchmarking score is a numerical value given to the SUT, which quantifies how well it performed according to the metric and through this numerical value one can compare the SUT with other similar systems.

A benchmark suite is defined as a collection of individual programs that help in comparing two systems or algorithms with each other. Benchmarking hardware and software provide a better understanding of the application for which they are designed and they also help to improve overall system's quality by measuring performance and highlighting bottlenecks in key areas. The past demonstrates that benchmarks have usually accelerated progress in their respective field [2]. Benchmarking is also of uttermost importance for the field of Machine Learning (ML) (with ML we imply both machine and deep learning) as with great pace new algorithms and specialized hardware are being introduced. With no standardized set of rules to compare these advancements, this might eventually slow the progress in this field. To keep up with the rapidly evolving field of ML, hardware and software vendors are coming up with specialized solutions focusing only on this domain [3]. To encourage further advancements more benchmarking tools are needed for these workloads. This paper aims to provide a comparison between seven ML benchmarks, which are MLPerf [4], DAWNBench [5], DeepBench [6], DLBS [7], TBD [8], AIBench [9], and ADABench [10] to make it easier for the new users to select the most optimal one as per their needs. These benchmarks are designed for specific applications and have their advantages and disadvantages. According to our knowledge, no effort to date has been done to compare all of these. The rest of this paper is structured as follows: we summarize related work in Section II. In Section III, we explain benchmarking from the ML perspective and list all the metrics and datasets that are usually employed by different benchmarks. Seven individual benchmarks found in the literature are presented in Section IV. In Section V, we compare these seven benchmarks and provide a thorough summary. In section VI, we reflect on the points that are lacking in current benchmark suites before concluding in Section VII.

 ISBN: 978-1-61208-845-7

II. RELATED WORK

As many benchmarks already exist for characterizing modern computer systems, we provide a brief overview of two such benchmarks that resulted in breakthroughs in microprocessors and hardware design [2]. First, the Systems Performance Evaluation Cooperative (SPEC) [11] benchmark. SPEC was founded in 1988 as a non-profit consortium of major computer vendors to provide an effective and fair comparison of advanced high-performance computing systems. Benchmark consisted of a set of programs (individual application benchmarks) where each carries equal weightage [12]. Second, the LINPACK benchmark [13] by Jack Dongarra, first introduced back in 1976. LINPACK comes under the category of an algorithmic benchmark that measures the floating-point performance of computers. It consists of subroutines that aim to solve a system of linear equations [12].

These benchmarks aimed to judge the relative performance of the hardware under test compared to some predefined system used as a reference. In the case of the LINPACK benchmark, the aim might be to know which microprocessor is the fastest, and generally a processor with a higher core count and a faster clock speed will outperform the others. ML workloads lack this simplicity. These workloads often utilize much complex hardware systems and algorithms which ultimately make benchmarking a difficult task [14]. This is enlightened in the third chapter. As new domains adopt ML in their life cycle, there is a constant need for benchmarking tools to evaluate different algorithms and hardware platforms to encourage further advancements.

There have been some efforts in summarizing different benchmarking principles for ML [14] but a thorough comparison between benchmarking suites is still missing. In addition to this, most available ML benchmarks do not utilize any real-world datasets that represent today's industrial need. For example, Mattson et al. [4], Zhu et al. [8], Gao et al. [9] all use ImageNet dataset [15] for benchmarking the computer vision domain but ImageNet might not be a good choice anymore for comparing Image classification [16]. Therefore in this paper, we provide a summary of commonly used benchmarks with their use cases and metrics to enlighten the fact that none of the benchmarks are completely fulfilling the industrial need.

III. BENCHMARKS FOR MACHINE LEARNING INFRASTRUCTURES

In this section, we introduce benchmarking from an ML perspective.

A. Benchmarking for ML Training and Inference

Benchmarking is a way to recognize the particular qualities and shortcomings of various approaches and frameworks. In ML, it can be associated with two individual tasks of the ML workflow, which do not overlap with each other. a) *Training:* For training an ML model, learnable parameters of the model have to be updated. This requires a forward and a backward pass wherein forward pass samples in mini-batches are shown to the model. In backward pass, intermediate results are stored in the memory which eventually adds a significant load on the hardware accelerators (usually GPUs). b) *Inference:* On the other hand, the inference is about evaluating a single data sample on the trained model at once. Therefore training usually requires expensive hardware with multiple cores whereas inference can be conducted even on simpler edge devices.

These two distinct processes have their separate benchmarks. In this paper, we focus mainly on training benchmarks as training is usually a resource expensive process. Training benchmarks compare different software solutions for a given task (e.g., Image classification) to know which one performs the best according to a particular metric. From a hardware perspective, training benchmarks focus on evaluating how fast a particular system can train a model to reach some predefined state-of-art performance for a given task. The inference benchmarks usually measure latency that translates to how fast a system can produce results in production once it has been trained.

B. Uniqueness of Machine and Deep Learning

Benchmark suites like SPEC [11] have established themselves as a source of guidance that has helped in standardizing requirements in the field of computing. Such benchmarks were successful, because of an end-to-end approach followed by the benchmark and also because of the lack of stochasticity involved in the domain [4]. ML on the other hand does not follow a common recipe. Even two runs of the same model under the same setting can produce different results [4]. Another source of randomness is the software frameworks in which the Ml model is built. In recent years there have been many such mathematical libraries that are capable of implementing a model in different ways.

The stochasticity involved in ML emerges as a major challenge when it comes to benchmarking with respect to training. This aspect is unique to ML training and is not encountered in traditional computing. ML is capable of offering multiple correct solutions for a single problem, unlike traditional technologies that offer only one perfect solution [4]. The other aspect of ML that makes benchmarking even harder is the diversity of problems that are present in the field. For example, it is not necessarily true that a system capable of solving Computer vision tasks efficiently will also be efficient for Natural language processing (NLP). Therefore, a training benchmark should aim to provide a standard evaluation criterion that considers different trade-offs (for e.g., performance vs speed vs different domains) when comparing systems or algorithms together.

Some of the requirements that an ML training benchmark should fulfill are:

- Provide a fair comparison between hardware systems and algorithms on common domains and datasets.
- Provide a fair comparison between different ML frameworks (for e.g., PyTorch vs TensorFlow) when running the same algorithm for a particular domain.
- Standardize a set of rules which could be followed by the user to ensure reproducibility of results.

 ISBN: 978-1-61208-845-7

- Provide a quantitative analysis between system level operations (for e.g., convolution, pooling) to know where the bottlenecks are present.
- Should measure systems on the basis of scalability (one server vs multiple servers) and should ensure transparency by using adequate metrics for each domain respectively.
- Should be able to especially handle stochasticity involved in machine learning workloads. One way of doing this to chose a metric that is consistent with the number of runs on average.
- Should be representative of industrial needs as many benchmarks in literature use datasets that are far too simple for the domain they represent.
- Should be transparent that providers of hardware or infrastructure accepting the benchmark
- Should be open source that everyone can validate the correctness of the implementation.

C. Classifications of ML/DL Benchmarks

An ML benchmark can also be categorized into one of three levels shown in Figure 1.

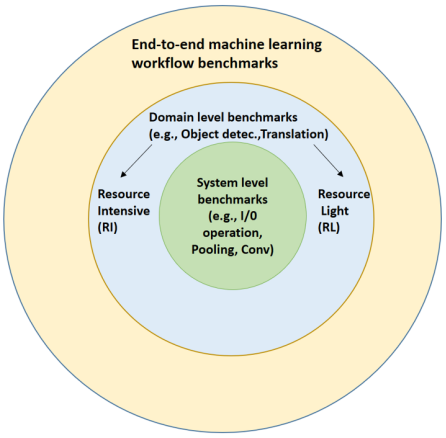

Figure 1. Category of ML benchmarks.

The first category, the *System Level (SL) benchmark* represents the lowest level, the fundament of the whole benchmarking chain. These benchmarks help to gather more insights at a basic level such that bottlenecks involved in basic operations could be found. One noteworthy example of such an operation is the activation functions involved in ML. Coleman et al. [17] showed that the rectified linear units (ReLU) [18] activation function in particular is an expensive operation prolonging the overall training process.

The second category, the *Domain level (DL) benchmark* targets specific domains that can utilize different small scale operations mentioned above. These benchmarks are important for evaluating hardware and software from a broader perspective to reflect upon the memory and computation requirements needed for each domain respectively. They are subdivided into two categories namely, **Resource Intensive (RI)** and **Resource Light (RL)**. A domain such as Image classification belong to the **Resource Intensive** subcategory, as it requires high GPU and memory whereas NLP comes under **Resource Light** subcategory due to reduced memory requirement. All of the commonly found ML benchmarks in the literature belong to the *Domain Level* benchmark category. It is important to note that these benchmarks still target only a handful of domains and also not all the domains are targeted in all of the commonly used benchmarks.

The last category, *End-to-end machine learning workflow benchmarks* focus on evaluating systems from an end to end perspective. Such benchmarks consider the whole benchmark as loosely coupled modules that could be easily changed and extended. These modules include data pre-processing pipeline, data input pipeline, different domain-specific set of operations (Domain level benchmarks), inference, training, model serving, and finally important non-artificial intelligence (AI) related modules which are critical for the application in focus. Such benchmarks provide extensive information about the SUT from training to production. In literature, AIBench Benchmark [9] is one of the few benchmark suites that comes under this category. They define this benchmark suite as a combination of essential attributes extracted out of different industry-scale applications. These particular applications define at first hand that which of the Domain level benchmarks should be used for that particular use case.

D. Metrics for ML-Benchmark

A typical ML workflow starts by gathering more insights about the data and problem at hand. This is followed by dataset formation and algorithm selection. The next step is to evaluate how well the algorithm performs and this is evaluated based on a specific metric. For example, in the task of binary classification, classification accuracy defines the fraction of samples in the test set that were predicted correctly. Using the only accuracy could be misleading due to the fact that this metric does not consider scenarios such as class imbalance. Similarly, in the case of benchmarking there are several metrics that can be employed. Choosing one over the other should be done carefully as this could be domain-specific or problem-specific. Table I provides a list of such metrics that are most commonly used by some of the machine learning benchmarks. It is important to note that most of the benchmarks chose one or two out of all the mentioned metrics.

The most commonly adopted metric out of all the above-mentioned ones is Time-to-Accuracy (TTA) metric. Coleman et al. [17] show that this metric generalizes nearly as well on unseen data. They also portray that even with all the stochasticity involved in the training procedure, the TTA metric stabilizes well with a low coefficient of variation (he ratio of the variance to the mean) concluded after multiple iterations.

E. Benchmark Datasets

A dataset is also a principal part of an ML benchmark as they help to test the system from the domain-specific

 ISBN: 978-1-61208-845-7

TABLE I
THE DIFFERENT METRICS THAT ARE USED BY TRAINING BENCHMARKS FOR COMPARING DIFFERENT SYSTEMS OR ALGORITHMS.

Name	Definition
TTA	*Time To Accuracy*: This metric measures the time (in seconds) to reach the predefined accuracy on validation set. The task and the algorithm are fixed during TTA measurement.
TTE	*Time To Epochs*: This metric measures the wall clock time (in seconds) taken to train some specific predefined epochs. The task and the algorithm are fixed during TTA measurement.
Energy Consumption	Energy consumed (in watts per second) till some accuracy is reached on the validation set.
Accuracy	This metric is used to compare novel algorithms with the state-of-the-art algorithms on a fixed task and a dataset in order to improve the best known results. It is defined as the number of correctly predicted samples out of the total samples present in the test set.
Cost	This metric is associated with instances in a cloud infrastructure. It describes the cost (in some currency) required for the training of an algorithm to reach a specified accuracy on validation set.
Throughput	Throughput defines the number of data points present in the training set that are processed per second on a system.
Batch time	It is the average time taken in ms to process one batch of data, i.e., the number of samples before the model is updated.
Flops	This metric measures either the floating point operations required for a particular operation (like convolution in convolutional neural networks (CNN)) or the total number of operations executed in whole training process.
GPU utilization	Fraction of time (in ms) the GPU is active in whole training process.
CPU utilization	This metric measures the average utilization of central processing unit (CPU) across all cores.
Memory Consumption	This metric aims to examine which of the operations or components utilize most of the memory. This will help in optimizing the training process.
Total time per operation	This metric calculates the time (in ms) required to complete a particular operation (convolution, pooling, etc.).

TABLE II
SOME COMMON DATASETS THAT ARE USED BY TRAINING BENCHMARKS FOR COMPARING DIFFERENT SYSTEMS OR ALGORITHMS.

Name	Characteristics	Domain
ImageNet [15] Cifar10 [19]	Imagenet: close to 1.2 million images, 1000 clases in total. Cifar10: 6000 images (32*32) per class, 10 classes in total. classes in total.	Image classification
COCO [20] Pascal VOC2007 [21]	COCO: more than 2M (5 captions per image) instances in 80 object categories. Pascal VOC: 9963 images, with each image containing set of objects from 20 different classes.	Object detection.
WMT English-German [22]	Translation dataset based on the data from statmt.org.	Language Translation
1TB ClickLogs [23]	Contains instances of feature values and click feedback for millions of display ads divided into 24 files.	Recommendation
Go [4]	MiniGo, data is generated while self-playing on a 9×9 game board.	Reinforcement learning
SQuAD [24]	Close to 10k instances of questions and answers.	Question Answering
LibriSpeech [25]	Contains approximately 1000 hours of English speech with a sampling rate of 16 kHz.	Speech recognition

point. Mostly standard open-sourced datasets are used by the majority of benchmark suites. Each dataset reflects the targeted domain. The table below provides a summary of such datasets with information about the domain they target.

IV. ML BENCHMARKING SUITS

In this section, we summarize seven different machine and deep learning benchmarks that have emerged in past as a joint effort from academia and industry to standardize benchmarking.

A. MLPerf

MLPerf is a consortium of commercial and academic organizations that has emerged as an industry standard for measuring ML systems. It offers both training and inference benchmark suites. The training benchmark suite (version 0.7) is a collection of eight machine learning models from 6 different domains. In the current version 0.7, there are two different sets of benchmark suites where one targets regular systems and the other is for High-Performance Computing (HPC) systems. It is the first benchmarking effort that aims to provide fair evaluations of training and inference performance for hardware, software, and services under prescribed conditions that guarantee reproducibility. There are two divisions; open and closed, where different vendors can submit their results. The goal of the closed division is to do a one-to-one comparison between hardware platforms or software frameworks. To use the closed division one has to utilize the same model and optimizer provided in the reference implementation. This forces one to follow certain guidelines under which the same preprocessing steps, same model, and training method should be used. On the other hand, the open division is for encouraging further advancements by allowing arbitrary preprocessing steps, new models, and training methods [4]. This benchmark can be considered as a combination of multiple Domain level benchmarks.

B. DAWNBench

DAWNBench benchmark suite can be regarded as a predecessor of MLPerf. It was designed for measuring end-to-end ML training and inference tasks. DAWNBench was introduced in November 2017 as a benchmark and a competition. Similar to MLPerf, DAWNBench also provides a reference set of common ML workloads. This benchmark was the first to use the Time-to-Accuracy (TTA) metric to measure performance

and allowed users to optimize model architectures, optimization algorithms, software frameworks, and hardware platforms. But it lacked rules, i.e., closed division in comparison to MLPerf [17]. Similar to MLPerf this benchmark suite can also be considered as a collection of the Domain level benchmarks.

C. DeepBench

DeepBench was released in 2016 from the Baidu research group. It is an open-sourced benchmarking tool focused on measuring the performance of the hardware at the kernel level. It can be considered as a System level benchmark. It aims to find which basic operations involved in deep neural network training are most time-consuming. The initial release only focused on benchmarking only training performance across multiple hardware platforms but the new version includes inference also. The benchmarking tool is available as a Github repository with reference implementations [6].

D. DLBS

Published in 2017, Deep Learning Benchmarking Suite (DLBS) is part of a large, comprehensive set of tools known as HPE's Deep Learning Cookbook. The cookbook aims to provide a guide for choosing ideal hardware and software for DL for both training and inference. It contains a web-based tool for analyzing the performance of deep neural networks, a benchmark suite that is available freely on Github, and reference designs for some selected workloads. The benchmarking suite itself consists of command-line programs that run different domain specific neural networks in multiple frameworks. The results for various hardware platforms, frameworks, and models are available online. Besides, the benchmark suite is also capable of producing results for untested hardware. Another interesting point about this benchmark is that it allows user-specific customized datasets, and one can use a synthetic dataset if no dataset is available [7]. This benchmarks suite also comes under the category of Domain level benchmark.

E. TBD

TBD (TrainingBenchmark for DNNs) benchmark suite is a joint effort from EcoSystem Research Group at the University of Toronto and Project Fiddle at Microsoft Research, Redmond. The benchmark suite was first introduced in 2018 with memory profiling tools for interpreting memory bottlenecks across three frameworks (CNTK, TensorFlow, MXNET) and recommendations for hardware and software selection for deep learning training. The suite consists of eight DNN models that overall cover six major domains. It is also a combination of Domain level benchmarks.

F. AIBench

AIBench, a Datacenter AI benchmark suite is one of the benchmark that comes under the category of End-to-end machine learning workflow benchmark. It consists of 17 Domain level benchmarks and 14 System level benchmarks that target nine real-world applications with 17 AI domains. The benchmark suite consists of loosely coupled modules that are flexible and easily configurable for multiple applications. Currently, two workflows are covered by the benchmark suite, first the E-commerce Search Intelligence, and second the Online Translation Intelligence [26].

G. ADABench

ADABench Is another benchmark suite that comes under the category of End-to-end machine learning workflow benchmark. It focuses on the complete end-to-end pipeline of ML workloads that comprises several additional steps including training, data integration (data input pipeline), data cleaning (data preprocessing pipeline), feature extraction, and model serving. There is no open-sourced implementation available for this benchmark but it is one of the benchmark suites that target industry-relevant domains like predictive maintenance as one of their use cases [10].

V. COMPARISON

Table III provides a summary of benchmarks mentioned in the previous section. The columns represent (from left to right), the name of the benchmark, the datasets used by these benchmarks, domains that the benchmark suite target with their category where SL, DL (RI & RL), and E represents System level, Domain level, and End-to-end machine learning workflow benchmark, and finally the metrics these benchmark's use. Starting with MLPerf, this benchmark suite uses a single metric, i.e., TTA, and targets only a few domains (6 vs. 17 in AIBench). TTA might be a good metric for IT-companies, which have abundant hardware resources to spare, and the cost of running these models not being a critical factor. But for some, cost as a metric could be a decisive factor in determining what kind of cloud infrastructure they want to invest in. Contrarily, DAWNBench uses cost as a metric in addition to TTA but targets only two domains. Coming to DeepBench, the micro-benchmark suite uses Teraflops and total time per execution of operations as metrics. This benchmark suite covers only a small set of operations that are involved in DL training.

From Table III, we can also see that only the DLBS benchmark offers the use of user provided datasets but on the other hand it only target two domains i.e. Language translation and Image classification. Furthermore, the results for already tested hardware platforms are not provided by the benchmark creators. For AIBench, the component benchmarks and the datasets used are not mentioned in the Table III individually due to the vast number of domains targeted by this benchmark. The domains it targets in the component benchmark are Image classification, Image generation, Language translation, Image-to-Text, Image-to-Image, Speech recognition, Face embedding, 3D Face recognition, Object detection, Recommendation, Video prediction, Image compression, 3D object reconstruction, Text summarization, Spatial transformer, Learning to rank, and Neural architecture search. However, AIBench lacks in providing fixed rules for reproducing results. In comparison to MLPerf, there are no definite rules mentioned for data preprocessing nor which hyperparameters could be changed

 ISBN: 978-1-61208-845-7

for each model individually. We consider tasks with image and video datasets as the most resource intensive therefore domains such as Image Classification and Object detection are sub-categorized as Resource intensive (RI). Reinforcement learning also uses image data and has additional complex tasks of control/action with some kind of update scheme therefore it is also added under the RI subcategory,

TABLE III
THE TABLE PROVIDES A SUMMARY OF THE SEVEN BENCHMARKS MENTIONED IN THIS PAPER.

Name	Dataset	Domain & Category	Metric
MLPerf	• ImageNeT • COCO • WMT Eng-Ger • 1TB Click Logs • Go	• Image classification (RI) • Object detection (RI) • Language Translation (RL) • NLP (RL) • Recommendation (RL) • Reinforcement learning (RI)	TTA
DAWN Bench	• ImageNet, Cifar10 • SQuAD	• Image classification (RI) • Question answering (RL)	TTA, Cost(in USD), Inference latency, Inference cost
Deep Bench	No real data used	• GEMM (SL) • Convolutional (SL) • Recurrent layers (SL) • All Reduce (SL)	Tera FLOPS, Total Time per operation (ms)
DLBS	Synthetic and real data (User provided dataset)	• Language translation (RL) • Image classification (RI)	Through -put, Batch time (ms)
TBD	• ImageNeT • IWSLT15 • LibriSpeech • Pascal VOC2007 • Downs. ImageNet • Atari	• Image classification (RI) • Machine Translation (RL) • Speech recognition (RL) • Object detection (RI) • Adversarial networks (RI) • Reinforcement learning (RI)	Through -put, GPU-Utilization, CPU-Utilization, F32-Utilization, Memory consumption
AIBench	• 17 different datasets	• 17 component benchmarks (E) • 14 micro benchmakrs (E)	TTA, TTE, Energy Consumption
ADA Bench	• Kaggle dataset • SMART dataset backblaze • Self generated • MovieLens	• Customer Service Management (RI) • Predictive Maintenance (RL) • Regression (RL) • Clustering (RL) • Classification (RI) • Recommendation (RL)	Throughput

VI. CRITICAL DISCUSSION TOWARDS AN IDEAL ML BENCHMARK

The benchmark suites mentioned in this paper lack a standard set of metrics. Even from the algorithmic point of view, a benchmark should offer multiple choices for a single domain. For example, all the mentioned benchmarks use classification accuracy as a metric for the Image classification domain. There are no options available to use Precision or Recall as a metric even with the fact that using accuracy only could be misleading. Besides, none of the mentioned benchmarks meet industrial needs as they do not allow user-specific datasets to be used (DLBS has that functionality but it targets only two domains). Domains like Image segmentation and Predictive maintenance are missing from the Domain level benchmarks. Furthermore, other than the MLPerf benchmark, no suite provides guidelines for having a fair comparison. These rules in MLPerf specify prime components such as the framework required for a particular domain (according to the reference implementation), loss function, and detailed information about the hyperparameter settings.

There is additionally a lack of support for testing cloud frameworks for ML. There are notable differences between different cloud providers. Platform such as Microsoft Azure [27] offer flexible compute options but have no built-in models whereas Google Cloud Platform (GCP) [28] offers auto ml tools with built-in models. The best cloud platform for ML is highly dependent on the application at hand. One has to carefully study the workflow used by these different providers and their data privacy regulations. Using cloud platforms would require data ingestion pipelines and additional processes which further increase the complexity. None of the benchmarks mentioned in this paper offer insights on any of these topics and neither do they provide any results for already tested cloud environments. Important metrics for comparing cloud platforms such as monetary cost, compute and storage performance are still missing from all the mentioned benchmarks (other than DAWNBench which has cost as a metric). Even prominent benchmarks such as MLPerf offer no guidelines for this cause. Even if one can transfer the reference implementation on a particular cloud framework, the datasets used, in some domains require high memory that adds to the overall cost. Furthermore, no metric available to compare the storage performances or cost of respective platforms discourages the idea of shifting the current benchmarks to cloud platforms.

We define an ideal benchmark that allows multiple domain specific metrics, e.g., in Image classification, one should be able to use ROC-AUC score instead of accuracy in TTA. It should fulfill the requirements mentioned in Section III. Moreover, it should also standardize rules through that other users could adapt their datasets for specific domains. These rules should also identify things that could be altered (e.g., hyperparameters, framework) to provide more flexibility but restrict changes that would damper the reproducibility aspect. Furthermore, these rules should be packed together with a

 ISBN: 978-1-61208-845-7

reference implementation in a containerization format that is easily transferable to different machines without requiring fresh installations of every library required by the benchmark. Finally, it should also provide support for testing ever-growing cloud frameworks. This could be achieved by providing support for transferring containers of reference implementations on different cloud platforms with additional monetary metric.

VII. CONCLUSION

Rapid growth in ML has opened a vast number of options in hardware platforms for the user. In addition to local machines, various cloud computing platforms such as Amazon Web Services, Microsoft Azure, Google Cloud Platform, IBM Cloud, etc, are available for ML. These various cloud platforms offer the possibility of modeling storage and compute capacity that can be scaled according to the need of the users. The benchmarks mentioned in this paper other than DAWNBench do not target the cloud platforms directly.

In this paper, we have summarized seven prominent benchmarking suites that help in making an informed decision about which hardware or software is the best for a specific application. Some of the benchmarking suites are still in their development phases and in the future, they can accelerate further progress in their respective fields. Inferring from the last section, the different benchmark suites employ different metrics but there seems to be no agreement on a standardized set. None of the benchmarks provide any implementation for domains like predictive maintenance which is highly relevant for the manufacturing industry. With the addition of more domains, the inclusion of cost as a metric, improvement on documentation, and support for cloud platforms; the MLPerf and AIBench benchmark suites have the potential to become the go-to benchmarking suite for all ML applications.

ACKNOWLEDGEMENT

The contents of this publication are taken from the research project ”(Q-AMeLiA) - Quality Assurance of Machine Learning Applications”, funded by the Ministry of Science, Research and the Arts of the State of Baden-Württemberg (MWK BW) under reference number 32-7547.223-6/12/4, and supervised by Hochschule Furtwangen University (Prof. Dr. Christoph Reich, IDACUS). The responsibility for the content is with the authors.

REFERENCES

[1] S. Bouckaert, J. Gerwen, I. Moerman, S. C. Phillips, and J. Wilander, “Benchmarking computers and computer networks,” *EU FIRE White Paper*, 2010.

[2] J. L. Hennessy and D. A. Patterson, *Computer architecture: a quantitative approach*. Elsevier, 2011.

[3] K. Ovtcharov, O. Ruwase, J.-Y. Kim, J. Fowers, K. Strauss, and E. S. Chung, “Accelerating deep convolutional neural networks using specialized hardware,” *Microsoft Research Whitepaper*, vol. 2, no. 11, pp. 1–4, 2015.

[4] P. Mattson, C. Cheng, C. Coleman, G. Diamos, P. Micikevicius, D. Patterson, H. Tang, G.-Y. Wei, P. Bailis, V. Bittorf *et al.*, “Mlperf training benchmark,” *arXiv preprint arXiv:1910.01500*, 2019.

[5] C. Coleman, D. Narayanan, D. Kang, T. Zhao, J. Zhang, L. Nardi, P. Bailis, K. Olukotun, C. Ré, and M. Zaharia, “Dawnbench: An end-to-end deep learning benchmark and competition,” *Training*, vol. 100, no. 101, p. 102, 2017.

[6] B. Research, “DeepBench,” https://github.com/baidu-research/DeepBench, 2018, [retrieved: March,2021].

[7] H. P. L. (HPL), “DLBS,” https://github.com/HewlettPackard/dlcookbook-dlbs, 2018, [retrieved: March,2021].

[8] H. Zhu, M. Akrout, B. Zheng, A. Pelegris, A. Phanishayee, B. Schroeder, and G. Pekhimenko, “Tbd: Benchmarking and analyzing deep neural network training,” *arXiv preprint arXiv:1803.06905*, 2018.

[9] W. Gao, F. Tang, L. Wang, J. Zhan, C. Lan, C. Luo, Y. Huang, C. Zheng, J. Dai, Z. Cao *et al.*, “Aibench: an industry standard internet service ai benchmark suite,” *arXiv preprint arXiv:1908.08998*, 2019.

[10] T. Rabl, C. Brücke, P. Härtling, S. Stars, R. E. Palacios, H. Patel, S. Srivastava, C. Boden, J. Meiners, and S. Schelter, “Adabench-towards an industry standard benchmark for advanced analytics,” in *Technology Conference on Performance Evaluation and Benchmarking*. Springer, 2019, pp. 47–63.

[11] K. M. Dixit, “The spec benchmarks,” *Parallel computing*, vol. 17, no. 10-11, pp. 1195–1209, 1991.

[12] J. Gray, *Benchmark handbook: for database and transaction processing systems*. Morgan Kaufmann Publishers Inc., 1992.

[13] J. J. Dongarra, “Performance of various computers using standard linear equations software in a fortran environment,” *ACM SIGARCH Computer Architecture News*, vol. 11, no. 5, pp. 22–27, 1983.

[14] W. Dai and D. Berleant, “Benchmarking contemporary deep learning hardware and frameworks: A survey of qualitative metrics,” in *2019 IEEE First International Conference on Cognitive Machine Intelligence (CogMI)*. IEEE, 2019, pp. 148–155.

[15] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, “Imagenet large scale visual recognition challenge,” *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[16] L. Beyer, O. J. Hénaff, A. Kolesnikov, X. Zhai, and A. v. d. Oord, “Are we done with imagenet?” *arXiv preprint arXiv:2006.07159*, 2020.

[17] C. Coleman, D. Kang, D. Narayanan, L. Nardi, T. Zhao, J. Zhang, P. Bailis, K. Olukotun, C. Ré, and M. Zaharia, “Analysis of dawnbench, a time-to-accuracy machine learning performance benchmark,” *ACM SIGOPS Operating Systems Review*, vol. 53, no. 1, pp. 14–25, 2019.

[18] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org, [retrieved: March,2021].

[19] A. Krizhevsky, “Learning multiple layers of features from tiny images,” *University of Toronto*, 05 2012.

[20] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Doll’a r, and C. L. Zitnick, “Microsoft COCO: common objects in context,” *CoRR*, vol. abs/1405.0312, 2014. [Online]. Available: http://arxiv.org/abs/1405.0312

[21] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results,” http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html, [retrieved: March,2021].

[22] O. Bojar, C. Buck, C. Federmann, B. Haddow, P. Koehn, J. Leveling, C. Monz, P. Pecina, M. Post, H. Saint-Amand, R. Soricut, L. Specia, and A. s. Tamchyna, “Findings of the 2014 workshop on statistical machine translation,” in *Proceedings of the Ninth Workshop on Statistical Machine Translation*. Baltimore, Maryland, USA: Association for Computational Linguistics, June 2014, pp. 12–58. [Online]. Available: http://www.aclweb.org/anthology/W/W14/W14-3302

[23] C. A. Lab, “Criteo 1TB Click Logs dataset,” https://ailab.criteo.com/criteo-1tb-click-logs-dataset/, [retrieved: March,2021].

[24] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, “SQuAD: 100,000+ Questions for Machine Comprehension of Text,” *arXiv e-prints*, p. arXiv:1606.05250, 2016.

[25] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: an asr corpus based on public domain audio books,” in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5206–5210.

[26] BenchCouncil, “AIBench: A Datacenter AI Benchmark Suite, BenchCouncil,” https://www.benchcouncil.org/AIBench/, 2019, [retrieved: March,2021].

[27] Microsoft, “Azure Machine Learning,” https://azure.microsoft.com/de-de/services/machine-learning/, [retrieved: March,2021].

[28] Google, “Google Cloud Platform,” https://cloud.google.com/products/ai, [retrieved: March,2021].

 ISBN: 978-1-61208-845-7