GREEN 2023

The Seventh International Conference on Green Communications, Computing and Technologies

ISBN: 978-1-68558-097-1

September 25 - 29, 2023

Porto, Portugal

GREEN 2023 Editors

Sandra Viciano Tudela, Universitat Politècnica de València, Spain

GREEN 2023

Forward

The Eighth International Conference on Green Communications, Computing and Technologies (GREEN 2023), held on September 25-29, 2023, continues the series of events focusing on current solutions, stringent requirements for further development, and evaluations of potential directions. The event targets are bringing together academia, research institutes, and industries working towards green solutions.

Expected economic, environmental and society wellbeing impact of green computing and communications technologies led to important research and solutions achievements in recent years. Environmental sustainability, high-energy efficiency, diversity of energy sources, renewable energy resources contributed to new paradigms and technologies for green computing and communication.

Economic metrics and social acceptability are still under scrutiny, despite the fact that many solutions, technologies and products are available. Deployment at large scale and a long term evaluation of benefits are under way in different areas where dedicated solutions are applied.

We take here the opportunity to warmly thank all the members of the GREEN 2023 technical program committee, as well as all the reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and effort to contribute to GREEN 2023. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

We also thank the members of the GREEN 2023 organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope that GREEN 2023 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in the field of green communications, computing and technologies. We also hope that Porto provided a pleasant environment during the conference and everyone saved some time to enjoy the historic charm of the city.

GREEN 2023 Chairs

GREEN 2023 Steering Committee
Daniele Codetta-Raiteri, Università del Piemonte Orientale, Italy
Sanjeev Sondur, Oracle / Temple University, USA
Mamadou Baïlo Camara, University of Le Havre Normandy, France
Bernard Lee, HedgeSPA, Singapore

GREEN 2023 Publicity Chair
Lorena Parra Boronat, Universitat Politecnica de Valencia, Spain
Laura Garcia, Universitat Politecnica de Valencia, Spain

GREEN 2023

Committee

GREEN 2023 Steering Committee
Daniele Codetta-Raiteri, Università del Piemonte Orientale, Italy
Sanjeev Sondur, Oracle / Temple University, USA
Mamadou Baïlo Camara, University of Le Havre Normandy, France
Bernard Lee, HedgeSPA, Singapore

GREEN 2023 Publicity Chair
Lorena Parra Boronat, Universitat Politecnica de Valencia, Spain
Laura Garcia, Universitat Politecnica de Valencia, Spain

GREEN 2023 Technical Program Committee

Khelifa Abdelkrim, Unité de Recherche Appliquée en Energies Renouvelables - URAER | Centre de Développement des Energies Renouvelables - CDER, Ghardaïa, Algeria
Kouzou Abdellah, Ziane Achour University of Djelfa, Algeria
Ali Ahaitouf, USMBA, Morocco
Daniel Albiero, Universidade Estadual de Campinas (UNICAMP), Brazil
Angelo Algieri, University of Calabria, Italy
Mahmoud Amin, Manhattan College, USA
Zacharoula Andreopoulou, Aristotle University of Thessaloniki, Greece
Hala Alami Aroussi, Mohamed Premier University, Morocco
Carlos Alberto Astudillo Trujillo, State University of Campinas, USA
Abdellah Bah, ENSAM | Mohammed V University in Rabat, Morocco
Figen Balo, Firat University, Turkey
Hajji Bekkay, ENSA-Oujda | Mohammed First University, Morocco
Rachid Benchrifa, Université Mohammed V, Morocco
Lazhar Benmebrouk, Ouargla University, Algeria
Suman Bhunia, Miami University, Ohio, USA
Anne Blavette, SATIE (CNRS) | ENS Rennes | Univ. Rennes, France
Guillaume Bourgeois, La Rochelle Université, France
Khalid Bouziane, International University of Rabat, Morocco
Mamadou Baïlo Camara, University of Le Havre Normandy, France
M. Girish Chandra, TCS Research Whitefield, India
Dana Ciupageanu, University Politehnica of Bucharest, Romania
Daniele Codetta-Raiteri, Università del Piemonte Orientale, Italy
Luigi Costanzo, Università degli Studi della Campania Luigi Vanvitelli, Italy
Naouel Daouas, National Engineering School of Monastir, Tunisia
Rekioua Djamila, University of Bejaia, Algeria
Rachid El Bachtiri, USMBA University, Fez, Morocco
Hassan El Bari, Ibn Tofail University, Morocco
Hassan El Fadil, Ibn Tofail University, Kénitra, Morocco
Rafika El Idrissi, Mohammed V University in Rabat, Morocco

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

Cost and Carbon Reduction for Microsoft Azure Virtual Machines Using Workload Analysis 1
Daisy Wong, Oliver Zhang, and Jacky Huang

Reducing Carbon Footprint of AI Models Without Compromising Performance 8
Austin Deng, Xingzhi Huang, and Michael Lu

Comparable Machine Learning Efficiency: Balanced Metrics for Natural Language Processing 15
Daniel Schonle, Christoph Reich, and Djaffar Ould Abdeslam

Cost and Carbon Reduction for Microsoft Azure Virtual Machines Using Workload Analysis

1st Daisy Wong
Richard Montgomery High School
United States
w6daisy@gmail.com

2nd Oliver Zhang
Strake Jesuit College Preparatory
United States
oliver.zhang0712@gmail.com

3rd Jacky Huang
St. Anne's-Belfield School
United States
zzandkun@gmail.com

***Abstract*—Cloud computing has rapidly become the dominant platform for businesses across various sectors. However, many companies find it challenging to effectively control costs, often resulting in suboptimal resource allocation and unnecessary overspending. Moreover, the expansion of cloud computing has spurred a surge in electricity consumption, causing a corresponding rise in greenhouse gas emissions. This paper aims to reduce both the cost associated with Virtual Machines (VMs) in the cloud and the carbon footprint generated by cloud computing activities. To tackle this issue, we analyze the 2019 Azure cloud trace, which incorporates data from more than 2.6 million VMs and historical records of grid emissions intensity from the California ISO Northern Region. We also devise a machine learning model to predict costs based on core and memory size and formulate a waste metric that captured over 90% of the wastage in cloud workloads. In addition, we propose a cost reduction algorithm that helps to save nearly 4 million dollars. Furthermore, we developed a carbon awareness algorithm that could substantially reduce the carbon emissions of VMs by 51%.**

***Index Terms*—cloud computing, Microsoft Azure, virtual machines, cost reduction, carbon reduction, workload analysis**

I. Introduction

Cloud computing has experienced significant growth in recent years, driven by the growing demand for cloud computing infrastructure. According to a report by FMI [1], the global VM market is expected to grow at a Compound Annual Growth Rate (CAGR) of 20.3% between 2023 and 2033. The market was valued at US $5,174.3 million in 2022 and is expected to reach US $26,042.8 million by 2033.

As cloud computing continues gaining momentum, organizations increasingly rely on cloud-based infrastructure to support their computing needs. However, cloud costs can spiral out of control if not managed properly, with VM usage significantly contributing to cloud costs. The report by Flexera [2] has indicated that enterprises wasting over 30% of their cloud spending, with wasted spending totaling $14.1 billion annually. As a result, reducing cloud expenditure has become a top priority for organizations using the cloud.

To address this issue, we investigate the problem of reducing costs and carbon emissions of cloud computing. To our knowledge, only a few works have attempted to reduce costs using real VM workloads, such as [3]–[5]. Furthermore, most of the works on energy-efficient management systems for VMs in the cloud proposed various migration algorithms [12] [13]. Our study deviates from existing literature and instead looks at carbon reduction through VM scheduling. Specifically, we make the following contributions.

Real-world workload analysis. To understand the waste problem, we analyze the characteristics of approximately 2.7 million VMs in the Azure Public Dataset [14] to verify the existence of waste in cloud computing. Our workload analysis shows that the average utilization rate of all VMs is only 15.59%. Moreover, we observe a consistent pattern that as the VM requests more resources (such as memory and cores), its average utilization rate tends to decrease.

VM Cost Prediction Model. As the dataset does not indicate cost per VM, our study utilizes a linear regression algorithm to estimate the cost of each VM using their specific characteristics. The algorithm considers various factors, including public pricing models by Microsoft [15], how many gigabytes of memory the VM has, and the number of cores the VM has. By accurately predicting the cost of each VM, our algorithm reveals the estimated cost of each VM.

Waste Quantification. Building on top of the proposed cost prediction model, we further propose a metric to quantify the waste of each VM. In this study, we quantify the waste of each VM by considering the total cost and resource utilization. This metric provides users with a clear understanding of the extent of waste in the cloud infrastructure and enables them to prioritize cost optimization efforts. Our waste quantification offers key insights into users of the cloud and our analysis confirms that many users waste significant cloud resources, leading to unnecessary costs. This finding highlights the importance of cost optimization for VMs in cloud computing environments.

Cost Reduction Algorithm. Our study proposes an effective solution for reducing the cost of cloud computing. Using the established waste metrics, we find VMs with high wastage frequently possess larger cores that are underutilized. Therefore, our proposed strategy involves downgrading VMs by reducing their core size until it reaches the minimum core size or the utilization surpasses 100%. By implementing this approach, users can effectively reduce costs without compromising performance. The results from our simulations confirm that we have achieved cost savings of up to 17%.

Carbon Reduction Algorithm. Our research introduces an innovative carbon reduction algorithm that aims to maximize the efficiency of cloud computing while minimizing carbon emissions. To achieve this, we utilize the Carbon Intensity

 ISBN: 978-1-68558-097-1

Data provided by WattTime [17] to assess the relative carbon reductions achievable by each VM during its operation. This data measures the emission rate of electricity generators on the local grid at a specific time, represented in units of total pounds of emissions per megawatt-hour. The dataset we utilize contains carbon intensity levels for California ISO (CAISO) region in July, recorded at 5-minute intervals. Leveraging this information, our carbon reduction algorithm makes predictions about the carbon intensity expected in the next 24 hours. It then strategically assigns VMs to a time window where they can contribute to carbon emissions reduction effectively. By adopting this approach, users can actively minimize their carbon footprint without compromising the quality of service.

The rest of the paper is organized as follows. Section II discusses related work, and Section III presents a thorough analysis of the workload. The cost reduction algorithms are presented in Section IV, and Section V presents the carbon reduction algorithm. Section VI provides an in-depth discussion of the experimental results obtained. Finally, Section VII presents the conclusions drawn from this research and potential future work.

II. Related Work

Several studies have investigated cost reduction for VMs in the cloud, offering valuable insights into optimizing resource allocation and managing costs. For instance, Flexera's State of the Cloud 2022 report [2] highlighted the increasing usage of cloud services and provided recommendations for cost management from a company's perspective. Others such as Cortez et al. [3] explored resource management and proposed a workload prediction model that can leverage machine learning techniques to enhance resource allocation in large cloud platforms. Similarly, Hadary et al. [6] developed a system that employed machine learning algorithms to automate VM allocation for large-scale cloud deployments. However, these studies did not address cost reduction from the user's standpoint.

In the realm of published literature focusing on carbon reduction strategies, several approaches have been explored, primarily centered around individual or multiple data centers. Notable examples include [7] and [8]. However, some researchers, as demonstrated by [9], have taken into consideration distributed data centers with varying carbon footprints and power usage effectiveness. Meanwhile, Beloglazov and Buyya [10] have evaluated heuristics aimed at dynamically reallocating VMs to minimize energy consumption. Furthermore, several other studies have examined alternative methodologies, such as migration algorithms that may pose challenges for users rather than enterprises [11] [12].

In contrast, our study analyzes the specific workloads of individual users and presents a comprehensive solution tailored to their needs. We introduce a program that automatically leverages historical data, utilization rates, and other factors to recommend the most cost-effective VM for a given workload. By directly addressing user requirements, our solution bridges the gap in existing research and offers a practical approach to cost reduction. We also propose a program that suggests users to reschedule work when possible to improve environmental sustainability. Users can significantly enhance their cost and carbon management practices, achieving substantial cost reductions without compromising performance or quality. The unique focus of our paper on individual users sets it apart from previous studies, making it a valuable contribution to the cloud computing cost and carbon reduction field.

III. Workload Analysis

In this section, we characterize the workload of Azure VMs. Exploring their characteristics and focusing on which intrinsic aspects of application and function will help enable numerous platform optimizations.

A. Microsoft Azure Workload

Dataset. In this paper, we utilize the Azure Public Dataset Version 2 [13], a cloud trace generated in 2019 from the Microsoft Azure platform. The dataset includes detailed information of approximately 2.7 million VMs created by over 6,600 users in July 2019, such as the timestamp in seconds when the VM is first created (starting at 0), the timestamp in seconds when the VM is deleted, VM size in terms of maximum core and memory (GBs), the average and maximum CPU utilization, as well as the string IDs of users.

B. Cost Analysis

Tools. While the Azure Public Dataset Version 2 [13] provided relevant information concerning VM characteristics, it excluded essential information regarding the price of each VM, which is essential for calculating the cost of each VM. To solve this problem, we utilize third-party tools such as the pricing calculator offered by Azure Microsoft [14], Orange [15], and SciKit Learn [16] to develop a cost prediction model.

Core/CPU	Mem	Cost
2	2	--
2	4	0.091
2	8	0.134
4	8	0.191
4	32	0.297
8	32	0.537
8	64	0.594
24	64	--
30	70	--

Figure 1. Data used to calculate cost

Pricing Calculator. To analyze the cost of all VMs, we first need to find the pricing of each VM. The pricing calculator offered by Azure Microsoft [14] calculates the prices of each

VM based on numerous factors. We set the VMs to be in the region of the East US, Windows operating system, OS-only type, standard tier, and memory-optimized category for 730 hours. In choosing instances, we picked instances that matched the core and memory with the lowest temporary storage. However, not every VM reported on the Azure Public dataset Version 2 was provided on the pricing calculator. As seen in Figure 1, there were instances of core and memory pairs that were not provided on the pricing calculator. The cost in the figure was measured in cost per hour.

Core/CPU	Mem	Cost
1	0.75	0.02
2	3.5	0.12
4	7	0.24
8	14	0.25
12	48	0.595
16	64	0.896
32	128	1.792
...	...	...

Figure 2. First few row of our training dataset

Linear Regression. We use the linear regression tool from Sckikit Learn [16] to solve this problem. In addition to the data shown in Figure 1, we included other data from the pricing calculator in order to train our our model. The first few rows of our training dataset can be seen in Figure 2. We chose linear regression because the variables, namely core size and memory space, exhibit a linear relationship. Hence, linear regression is more suitable for capturing this linear association, as opposed to quadratic or alternative regression models. Through our linear regression model, we obtain the coefficient of determination R^2 as 0.98566, signaling the accuracy of our prediction model. Our model is determined through the following equation:

$$price = 0.01530 + 0.00055 \times core + 0.014222 \times mem$$

Where 0.01530 represents the intercept, 0.00055 constitutes the slope for the core variable, and 0.014222 forms the slope for the memory variable.

Additionally, compared to other machine learning techniques, linear regression handles overfitting relatively well using different procedures such as reduction techniques, regularization, and cross validation. While comparing our linear regression results to other machine learning techniques, we found the linear regression results to be more accurate.

Finally, with the price of VMs of different core sizes and memory space combinations, we are able to calculate the cost of each VM based on runtime using the following equation:

$$cost = price \times (\frac{runtime}{3600})$$

The VM cost is calculated as the product of the VM's price, which depends on its core size and memory space, and the duration in hours. The runtime is calculated by subtracting the deletion timestamp from the creation timestamp. To convert this duration into hours, we divided the result by 3600, as there are 3600 seconds in an hour.

C. Utilization Analysis

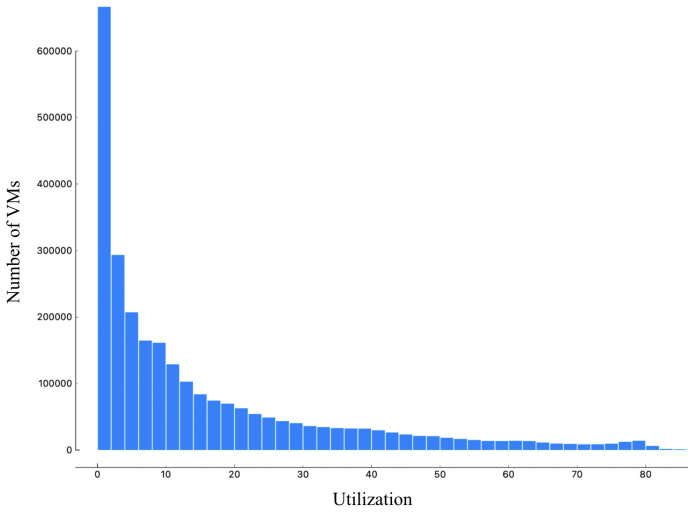

Figure 3. Utilization of VMs

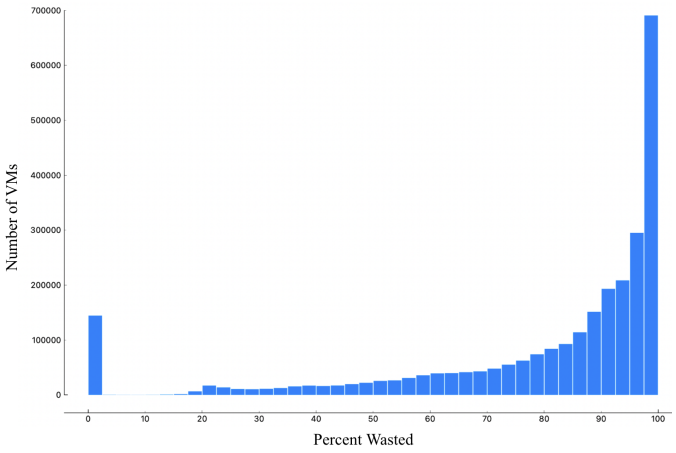

Figure 4. Percent wasted for VMs

User Analysis. Many Azure users created more than one VM. Based on different users, some VMs averaged with low core and memory count, while other VMs averaged with higher core and memory count. Although we observe that some users can efficiently utilize cloud resources, a large group of users also seem to be managing their resources ineffectively. We will refer to the first group of users as the "experienced" users and the second group as the "inexperienced users."

Experienced users all utilized VMs in an indistinguishable manner. Most importantly, their VMs consisted of low core

 ISBN: 978-1-68558-097-1

and memory count (often two cores and two GB). Their VMs ran for a short time rather than the entire 30 days analyzed in the dataset. Their experience was also evident in their high utilization rates, which signified that these experienced users were using their VM for a limited time before deleting it immediately. Inexperienced users utilized VMs in opposition to experienced users. Their VMs consisted of high core and memory count (many of which used the maximum 64 cores and 70 GB).

Additionally, these VMs had a duration averaging 30 days with extremely low utilization rates, sometimes running below 1%. The characteristics of the VMs created by inexperienced users portray the typical scenario of creating a VM and forgetting to shut it down when it is no longer needed. Consequently, these VMs cause significant waste.

VM Analysis. The distribution chart depicted in Figure 3 illustrates a negative exponential curve. Analyzing the chart more in-depth reveals that more than half (55%) of VMs had a utilization rate below 10%. The largest bar indicates about 24% of VMs being utilized at below 2%. More significantly, 24% of VMs had a utilization rate of less than 2%. The analysis shows cause for concern since most VMs show wasteful spending, and many of these VMs are generated by inexperienced users. The trends in Figure 3 correlate with the trends depicted in Figure 4. The percentage wasted for VMs was calculated through a cost over waste ratio. The data in Figure 4 acknowledges that 51% of VMs wasted more than 90% of their expenses. Moreover, 25% of VMs drained more than 97.5% of costs. Note that 5% of VMs had zero percent wasted due to their time of use being zero seconds. Overall the two trends detailed in Figures 3 and 4 emphasize the potential for a significant cost reduction, which this paper seeks to contribute.

D. Waste Analysis

The waste of cloud expenditure is calculated using the formula below:

$$waste = \frac{cost \times (100 - util)}{100}$$

According to our calculations, the combined cost of all VMs exceeds $23 million, with an estimated waste of over $20 million. These figures highlight that customers are squandering approximately 90% of their VM resources. The disparity between waste and cost emphasizes the urgency to reduce cloud cost and waste.

IV. Cost Reduction Algorithms

Recall Section III. C, inexperienced users utilize large VMs at lower utilization rates, leading to some of the most significant waste in the cloud. Inspired by this observation, we propose two cost reduction algorithms that are referred to as aggressive downgrading approach and passive downgrading approach.

Downgrading a VM involves reducing its core size to the smallest possible core. Instead of considering memory space, we prioritize core size when downgrading VMs since utilization issues usually relate to core size rather than memory space. The aggressive approach focuses on downgrading VMs based on cores and the lowest memory, while the passive approach considers downgrading VMs based on the core with the highest memory.

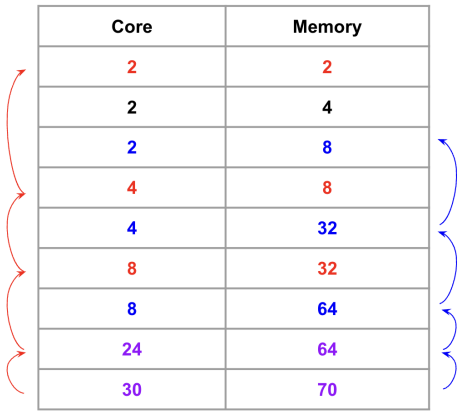

Figure 5. A list of VM core/memory combination

A. Sample Cases and Approaches

The two approaches are illustrated in Figure 5 through the pathways on the left and right sides of the table, respectively. In the first approach, which is more aggressive, the downgrading is performed based on the cores with the lowest memory. Let us consider an example where we attempt to downgrade a VM with a core size of 30 and 70 GB of memory. Following the “aggressive” approach (indicated by the red arrows on the left), the VM is downgraded to the core level with the smallest memory space.

On the other hand, the second approach adopts a passive strategy by downgrading based on the cores with the highest memory. If we were to apply the “passive” approach to downgrade a VM with a core size of 30 and 70 GB of memory, it would follow the pathway depicted on the right side of Figure 5, indicated by the blue arrows, resulting in a downgrade to the core level with the largest memory space.

B. Threshold Selection

It would be highly inefficient to downgrade every VM in the trace. To identify VMs eligible for downgrading, we established a criterion that selected VMs must have exceeded a 10% waste threshold, as indicated by the distribution chart shown in Figure 4.

Considering the limitations imposed by maximum utilization, average utilization, and core sizes, our proposed approaches necessitate VM utilization to fall within a range of 0% to 33%-50%. Let us assume a VM with 8 cores is downgraded to 4 cores. To calculate the new utilization, we divide the two core sizes, resulting in a quotient of 2, and multiply it by the old utilization rate. Since the maximum utilization possible for all VMs is 100%, the theoretical maximum utilization can only reach 50%. Similarly, if we

 ISBN: 978-1-68558-097-1

Algorithm 1 Aggressive Downgrading

Require: $percentWasted \geq 10$
  while $canDowngrade$ *and* $core >= 2$ **do**
    if $core = 2$ **then**
      $newMaxUtil = 2/2 \times maxUtil$
      $newUtil = 2/2 \times util$
      $core = 2$
      $mem = 2$
    else
      $newMaxUtil = oldCore/newCore \times maxUtil$
      $newUtil = oldCore/newCore \times util$
      if $newMaxUtil < 100$ **then**
        $maxUtil = newMaxUtil$
        $util = newUtil$
      else
        $canDowngrade = false$
      end if
    end if
    if $newMaxUtil > 100$ *or* $newUtil > 100$ **then**
      $core = coreLevel$
      $mem = memLevel$
    end if
  end while

Algorithm 2 Passive Downgrading

Require:
  $percentWasted \geq 10$
  $core \geq 2$ and $mem \geq 8$
  while $canDowngrade$ and $core >= 2$ **do**
    if $core = 2$ **then**
      $newMaxUtil = 2/2 \times maxUtil$
      $newUtil = 2/2 \times util$
      $core = 2$
      $mem = 8$
    else
      $newMaxUtil = oldCore/newCore \times maxUtil$
      $newUtil = oldCore/newCore \times util$
      if $newMaxUtil < 100$ **then**
        $maxUtil = newMaxUtil$
        $util = newUtil$
      else
        $canDowngrade = false$
      end if
    end if
    if $newMaxUtil > 100$ or $newUtil > 100$ **then**
      $core = coreLevel$
      $mem = memLevel$
    end if
  end while

consider a VM with 24 cores downgraded to the next level, 8 cores, following the same principle, the maximum utilization can only be 33.3%. These restrictions significantly contribute to the efficiency of targeting and downgrading VMs.

When the VM initiates the downgrade process, it computes and assigns the new maximum utilization, average utilization, memory, and cores to their respective variables. If either the maximum or average utilization surpasses 100%, the downgrade process is halted since utilization cannot exceed 100%. Likewise, if the VM reaches the minimum downgrade level with a core size of 2, the downgrade process is also concluded.

V. Carbon Reduction Algorithm

VMs consume massive amounts of energy, often from fossil fuel-based sources, thus leading to significant carbon emissions. This section presents a carbon reduction algorithm that identifies proper VMs and reschedules them to a low-carbon window to reduce their carbon footprint.

A. Carbon Intensity

Carbon intensity is the measurement of emissions generated relative to the energy consumed. In this paper, we use the Marginal Operating Emissions Rate (MOER), a widely used metric, to measure the carbon intensity in units of pounds of emissions per megawatt-hour. MOER measures the emissions rate of electricity generators on a certain grid at a certain time. Low MOER indicates that electricity is more environmentally friendly and vice versa. The Azure trace [13] was created with data from July 2019. Consequently, we utilized the corresponding MOER dataset from CAISO North during July 2019 as our carbon intensity data.

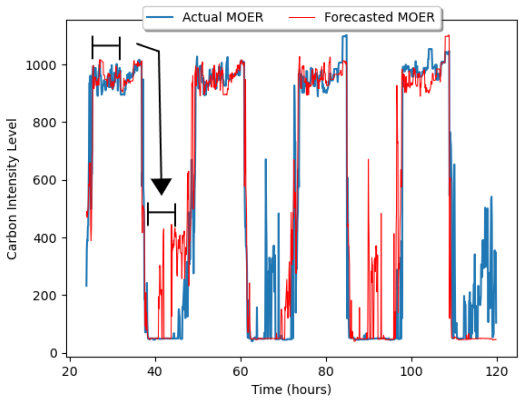

Figure 6. Forecasted and actual MOER from July 2-5, 2019

As illustrated by the blue line in Figure 6, MOER values in the CAISO North region vary greatly, but typically surges during the day when electricity demand is higher. This increased energy consumption by businesses and households often prompts grids to deploy additional generators, including those with higher emissions, to meet the heightened demand. Conversely, demand and MOER tend to decrease during the night, enabling a shift towards more sustainable electricity generation. This variability in MOER arises from the interplay

 ISBN: 978-1-68558-097-1

of numerous external factors that shape the dynamics of the electricity grid, which include weather and the makeup of generators within the grid (renewable or non-renewable).

B. Carbon Intensity Forecasting

Forecasting MOER allows workloads to be delayed until times of low MOER to reduce carbon emissions. Our forecasting algorithm uses a robust prediction methodology. Our methodology operates under the assumption that the MOER levels observed today will persist unchanged into tomorrow. Alternative forecasting models, such as the moving average approach, were also evaluated and tested. Nonetheless, due to challenges posed by data availability, the moving average model encountered limitations in precisely anticipating MOER levels.

The subsequent component of the algorithm centers on optimizing carbon reduction. This is achieved by identifying the interval with the lowest MOER level averages over the next 24 hours to execute the workload. Operational activities are postponed and rescheduled until the interval mentioned above. As shown in Figure 6, workloads during the high MOER times are shifted to times of lower MOER levels, thereby reducing carbon emissions.

There are three types of workloads in the Azure Public Dataset "Delay-sensitive," "Delay-insensitive," and "Unknown." As their names suggest, delay-sensitive workloads are time-sensitive, delay-insensitive workloads are not, and unknown workloads do not specify whether it is time-sensitive. Because users did not mark unknown type workloads as delay-sensitive, we assumed they could be postponed. For our simulation, we ran both "Delay-insensitive" and "Unknown" type workloads through the optimization algorithm as they were more amenable to rescheduling and could tolerate delays.

Algorithm 3 Carbon Reduction

Require: $RunTime > 0$
  $AverageMOER = Average\ MOER\ in\ Window$
  $Window \leftarrow Start\ Time\ to\ EndTime$
  for $\left(\frac{1Day}{5minutes}\right)$ **do**
    $Shift\ Window\ By\ 5\ Minutes$
    $NewAverage \leftarrow Average\ of\ Modified\ Window$
    if $NewAverageMOER < AverageMOER$ **then**
      $AverageMOER \leftarrow NewAverageMOER$
      $BestWindow \leftarrow Window$
    end if
  end for
  $Reschedule\ Task\ to\ BestWindow$
  $Savings = DefaultEmission - AverageMOER$
  $PercentSavings = 100 \times \left(\frac{Savings}{DefaultEmission}\right)$

Our carbon reduction program, found as Algorithm 3, works by identifying the length of the window and shifting it until it finds the best duration with the lowest MOER levels. The workload depicted in Figure 6 shows the results of optimization. The program matched the window length to the corresponding MOER levels. It then shifts the window and calculates the new average; if the new average is lower than the past lowest average, the new average is set as the lowest average, and the best window is set to that duration. The window shifts every 5 minutes to match the MOER data that only provides data in five minute intervals. The program then loops 720 times, the number of 5 minute intervals in a day, to find the lowest average and the corresponding window. Then, it calculates the percent saving after optimization.

VI. Experimental Results

In this section, we will discuss the experimental results from the simulations on the cost and carbon reduction programs.

A. Cost Reduction Results

We ran our simulations on the entirety of the 2019 Azure cloud trace [13]. First, we calculated the total cost of over 2.6 million VMs in the trace, resulting in a total of $23,144,128.53 before downgrading. Then, we ran two simulations to compare user savings using the two cost reduction algorithms presented in Section IV. After downgrading, the cost savings were astonishing. Through our first downgrading approach, 1,975,282 VMs were aggressively downgraded and saved users a total of almost $4 million or 17% in savings. Through our second downgrading approach, 730,436 VMs were passively downgraded and saved users a total of about $950,000 or 4% in savings.

B. Carbon Reduction Results

We ran the carbon reduction simulation on both"Delay-insensitive" workloads and "Unknown" type workloads. In the"Delay-insensitive" group, we observed savings ranging from 0 to 8%, with mean savings around 0.2%. In the "Unknown" group of workloads, the average savings was 55%. Percent savings overall averaged 51%. The drastic difference in outcomes is explained by the differences in run time each group has. "Delay-insensitive" workloads run for long durations, with an average of 26 days and peaks around 30. These lengthy workloads are too substantial to generate significant savings through optimization, as they are too large to accommodate the dips of MOER. Meanwhile, the average run time of "Unknown" type workloads is only approximately four and a half hours, making them more susceptible to generating sizeable savings during optimization.

VII. Conclusion and Future Work

This paper analyzed the Azure workload of over 2.6 million VMs. We developed two cost reduction algorithms and a carbon reduction algorithm. The experimental results have shown that our proposed algorithms can help reduce costs of up to 17% of cost by efficiently choosing core size and memory space and reducing on average 51% of carbon emissions by rescheduling workloads to a low carbon time. In the future, we would like to explore the relationship between cost savings and carbon reduction.

Our proposals faces a few limitations. While the cost reduction algorithm is limited by sheer number of VMs in

 ISBN: 978-1-68558-097-1

the dataset, the carbon reduction algorithm is limited by the challenge of accurately predicting MOER due to its complex nature influenced by various external factors. Our algorithm adopts a simplified approach, primarily relying on historical MOER data to extrapolate future levels. This simplification enhances computational efficiency and operational feasibility but sacrifices precision by disregarding the intricate dynamics underlying MOER fluctuations. This can be addressed in future work by developing a more sophisticated algorithm incorporating external factors to achieve more precise forecasting of MOER levels.

Our study was conducted with the assumption that we can change VM size and run time whenever it is needed. However, this does not reflect the real world scenarios 100% of the time. Thus although our solution aims to tackle memory space issues through the duo use of the aggressive and passive algorithm, real world applications of the cost reduction algorithm could still be constrained by VM availability and memory space; a VM may still be reduced to a level where the program does not have enough memory to run. Additionally, both the cost and carbon reduction algorithms would need user consent to downgrade their VMs, as some users may require a specific VM size without changes, or other users may not be able to reschedule their VM workloads.

Overall, we believe inexperienced users mentioned in Section III. C would benefit the most from our cost reduction program. Users with workloads that can be rescheduled would save the most carbon reductions. Both the cost and carbon reduction algorithms are recommended to be implemented in real cloud providers such Microsoft Azure, Amazon Web Services (AWS), Alibaba etc.

REFERENCES

[1] "Virtual Machine Market," FMI, Dec. 2022. https://www.futuremarketinsights.com/reports/virtual-machine-market (accessed Jul. 2023)

[2] "State Of The Cloud Report," Flexra, 2022.

[3] E. Cortez, A. Bonde, A. Muzio, M. Russinovich, M. Fontoura, and R. Bianchini, "Resource Central: Understanding and Predicting Workloads for Improved Resource Management in Large Cloud Platforms * CCS CONCEPTS," 2017. Accessed: May 2023. Available: https://www.microsoft.com/en-us/research/wp-content/uploads/2017/10/Resource-Central-SOSP17.pdf

[4] M. Shahrad et al., "Serverless in the Wild: Characterizing and Optimizing the Serverless Workload at a Large Cloud Provider," USENIX, 2020. Accessed: May 2023 Available: https://www.usenix.org/conference/atc20/presentation/shahrad

[5] B. Everman, M. Gao, and Z. Zong, "Evaluating and reducing cloud waste and cost—A data-driven case study from Azure workloads," Sustainable Computing: Informatics and Systems, vol. 35, p. 100708, Sep. 2022, Accessed: May 2023. [Online] doi: https://doi.org/10.1016/j.suscom.2022.100708.

[6] O. Hadary et al., "Protean: VM Allocation Service at Scale," www.usenix.org, 2014. Accessed: May 2023. [Online]. https://www.usenix.org/system/files/osdi20-hadary.pdf

[7] A. Berl et al., "Energy-Efficient Cloud Computing," The Computer Journal, vol. 53, no. 7, pp. 1045–1051, Aug. 2009, Accessed: May 2023. [Online] doi: https://doi.org/10.1093/comjnl/bxp080.

[8] A. Uchechukwu, K. Li, and Y. Shen, "Energy Consumption in Cloud Computing Data Centers," International Journal of Cloud Computing and Services Science, vol. 3, no. 3, Jun. 2014.

[9] A. Khosravi, S. Garg, and R. Buyya, "Energy and Carbon-Efficient Placement of Virtual Machines in Distributed Cloud Data Centers," in Proceedings of the 19th international conference on Parallel Processing, Aug. 2013, pp. 317–328.

[10] A. Beloglazov, "Energy-Efficient Management of Virtual Machines in Data Centers for Cloud Computing," 2010.

[11] M. Giacobbe, A. Celesti, M. Fazio, M. Villari, and A. Puliafito, "An Approach to reduce carbon dioxide emissions through virtual machine migrations in a sustainable cloud federation," IEEE Xplore, Apr. 01, 2015.

[12] S. Supreeth and K. Patil, "VM Scheduling for Efficient Dynamically Migrated Virtual Machines (VMS-EDMVM) in CLoud Computing Enviornment," KSII Transactions on Internet and Information Systems., vol. 16, no. 6, pp. 1892-1912, Jun.2022.

[13] E. Cortez, A. Bonde, A. Muzio, M. Russinovich, M. Fontoura, and R. Bianchini, "AzurePublicDatasetV2," Github, 2017. [Online]. Avaliable: https://github.com/Azure/AzurePublicDataset

[14] Microsoft, "Pricing Calculator — Microsoft Azure," Microsoft.com, 2023. https://azure.microsoft.com/en-us/pricing/calculator/ (accessed Jun. 2023)

[15] J. Demšar et al., "Orange: Data Mining Toolbox in Python," Journal of Machine Learning Research, vol. 14, no. 71, pp. 2349–2353, 2013, Accessed: Jul. 12, 2023. [Online]. Available: http://jmlr.org/papers/v14/demsar13a.html

[16] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python" Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011, Available: https://jmlr.csail.mit.edu/papers/volume12/pedregosa11a/pedregosa11a.pdf

[17] "API Reference," www.watttime.org https://ww.watttime.org/api-documentation/introduction (accessed Jun. 23, 2023).

 ISBN: 978-1-68558-097-1

Reducing Carbon Footprint of AI Models Without Compromising Performance

Austin Deng
Vandegrift High School
Austin, United States
email: austin12709@gmail.com

Xingzhi Huang
St. Anne's-Belfield School
Charlottesville, United States
email: zzandkun@gmail.com

Michael Lu
James Bowie High School
Austin, United States
email: michaelluaustin@gmail.com

***Abstract*—The widespread adoption of Artificial Intelligence (AI) models, such as ChatGPT, has resulted in a significant increase in energy consumption and carbon emissions associated with their training and inference. However, research on sustainable AI is still nascent. This paper aims to explore efficient approaches that can reduce the carbon footprint of AI models without compromising performance. Through an extensive analysis of four distinct categories of AI models (text to text summary, image classification, text to image, and image to text) across various sizes, our findings challenge the prevailing notion that larger AI models consistently outperform smaller ones. In specific AI tasks, we observe that small models can achieve comparable performance while significantly reducing carbon emissions. Moreover, we propose a carbon-aware solution that strategically directs computationally intensive AI tasks to regions with low carbon intensity, which can effectively reduce the environmental impact without compromising model quality. Our experimental results demonstrate a significant carbon savings while maintaining the desired performance levels.**

***Index Terms*—AI; Energy Efficiency; Carbon Emission.**

I. Introduction

The emergence of ChatGPT and GPT-4 has led to a remarkable surge in the popularity of AI. Within just two months, ChatGPT alone has attracted over 100 million users, showcasing the immense potential of AI models to revolutionize our daily lives and work. However, this surge in popularity has also resulted in a significant increase in energy consumption and carbon emissions attributed to AI. Unfortunately, the environmental consequences of AI models have not received sufficient attention, and efforts to mitigate their carbon footprint are still in the nascent stages of research.

One way to reduce the carbon emissions of AI is to use smaller and less energy-intensive models when possible. However, since it is commonly assumed that larger AI models consistently outperform smaller ones, smaller models are less preferred in practice.

In this study, we conduct a comprehensive analysis on 11 AI models spanning four different domains: text-to-text summary, image classification, text-to-image generation, and image-to-text generation. The analyzed text to text summary models include "t5-one-line-summary" and "t5-base-finetuned-summarize-news" [1]. For image classification models, we examine "Google-vit-base-patch16-224" [2], "Google-vit-base-patch16-384" [2], "Microsoft-cvt-13" [3], and "Microsoft-resnet-50" [4]. Regarding text-to-image generation models, we investigate "stable-diffusion-v1-4", "stable-diffsion-v1-5, and "stabilityai/stable-diffusion-2-1" [5]. We also analyze two image to text generation models ("trocr-base-printed" and "trocr-large-printed" [6]). Our experiments have demonstrated that resource efficient models can achieve comparable or superior performance in tasks such as image classification and image-to-text generation. Meanwhile, we observe that less power-hungry models can lead to a remarkable reduction in energy consumption, potentially up to 69%.

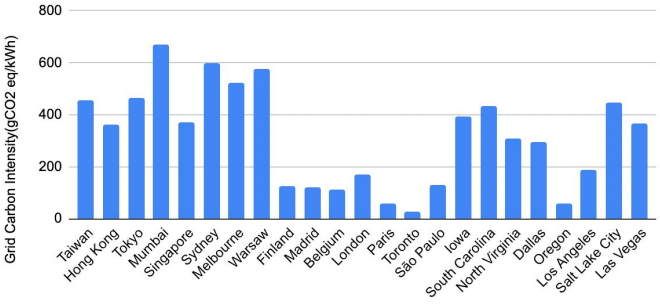

Fig. 1. Graph of carbon intensities for different locations

However, the same principle does not apply to text-to-text summarization and text-to-image generation models. In these cases, larger models (e.g., stable-diffusion-2-1) tend to generate higher quality images than smaller models such as stable-diffusion-v1-4. To balance the need for maintaining model quality while minimizing carbon emissions, we propose a carbon-aware solution. This solution strategically directs computationally intensive AI tasks to regions with lower carbon intensity in electricity production, thereby mitigating environmental impact without compromising the quality of the models. Figure 1 [7] illustrates the substantial variations in carbon intensity among different regions. For instance, the carbon intensity of Mumbai is approximately 23 times higher (670 gCO2eq) than that in Toronto (29 gCO2eq). This implies that deploying identical AI models in Toronto instead of Mumbai could potentially reduce carbon emissions by 95.67%.

This paper makes the following contributions:

- We quantitatively evaluate the energy consumption and carbon emission of 11 AI models.
- We reveal that using smaller models in image classification and image-to-text generation tasks can significantly reduce carbon footprint without compromising model quality.
- We propose a carbon-aware approach to mitigate the carbon emissions associated with AI tasks requiring large models, while ensuring no compromise on model quality.

 ISBN: 978-1-68558-097-1

The subsequent sections of this paper are structured as follows. Section II discusses related work, and Section III presents the detailed information about the AI models we evaluate. The methodologies for model quality evaluation, carbon emission measurement and carbon-aware model deployment are presented in Section IV. Section V presents experimental results and Section VI concludes this study.

II. Related Work

In recent years, the environmental impact of AI has garnered increasing attention, despite its status as a relatively young field. Numerous studies and reports have been published to better understand the significant environmental impact of AI training and inference.

Amodei et al. revealed that the computing demand for training AI models have increased 300,000 times in recent years [8]. The environmental impact of large AI models was highlighted by MIT Technology Review [10], stating that training a single AI model can emit 626,000 pounds of carbon dioxide, equivalent to the lifetime emissions of five average American cars [13]. However, few works have proposed ways to reduce the amount of carbon emissions generated by AI models. In [12], Schwartz et al. pointed out that traditional AI research (a.k.a. Red AI) focused on improving accuracy through the use of massive computational power while disregarding the cost and environmental impact. Red AI is not sustainable because the relationship between model performance and model complexity is understood to be logarithmic, meaning an exponentially larger model is required to gain a linear increase in performance [9]. For example, Mahajan et al. [11] reported that object detection accuracy increases linearly as the number of training examples increases exponentially. The opposite approach is Green AI, which emphasizes the importance of developing AI research that considers the computational cost and resource utilization [12]. According to Wu C. et al. [14], a deliberate and responsible approach is necessary when developing AI technologies, taking into account the environmental impact of innovations.

Although previous studies offered different approaches to enhance AI efficiency and decrease its carbon footprint through technological advancements, they generally overlooked the quantification of carbon emissions associated with distinct AI models. In contrast, our research takes a novel standpoint by quantitatively measuring the carbon emissions of 11 AI models. The results of our study demonstrate that using smaller AI models can be an effective and easily implementable strategy to reduce carbon emissions in AI systems. In situations where larger models are necessary for optimal performance, we propose an innovative carbon-aware solution, which can reduce carbon emissions by deploying AI models in regions with low carbon intensity.

III. AI Models

In this section, we describe the specific AI models we used for each of the four categories discussed earlier.

A. Text to Text Summary Models

Text-To-Text Transfer Transformer (a.k.a. T5) model was trained on up to 770 million parameters. The model takes a text as its parameter and can achieve four tasks: Translation, Corpus of Linguistic Acceptability, Semantic Textual Similarity Benchmark, and Summary. We evaluate two T5 models and focus on their summary functionality. The "t5-one-line-summary" [1] model was trained additionally on 370,000 research papers and can generate one line summary based on the abstract of the papers. It is conservatively estimated that the model has been downloaded 466,000 times a month from the Huggingface downloading data. The "t5-base-finetuned-summarize-news" [1] model was trained on news articles and the summarized versions of corresponding news articles. The model takes a small piece of news articles and can generate a brief summary of the article. The model is conservatively estimated to have 267,000 downloads a month.

B. Text to Image Generation Models

Stable Diffusion is a text-to-image generation AI model developed by Stability AI. It employs a technique where Gaussian noise is added to an image then removed in a manner that generates images corresponding to a given text or image prompt. We evaluate three different versions of Stable Diffusion models available on Huggingface: "stable-diffusion-v1-4", "stable-diffsion-v1-5, and "stabilityai/stable-diffusion-2-1" [5].

C. Image Classification Models

Image classification models aim to accurately classify the contents of different images. We evaluate four popular image classification models published on Huggingface. Both the "google-vit-base-patch16-224" model and the "google-vit-base-patch16-384" [2] model are derived from Vision Transformer (ViT). The "google-vit-base-patch16-224" model was trained using 224x224 resolution images while the "google-vit-base-patch16-384" model was trained on 384x384 resolution images. The "microsoft-cvt-13" [3] is an image classification model that adds convolutional neural networks to the ViT architecture. This model aims to help reduce the effects of distortions on the accuracy of ViT-based models. The "microsoft-resnet-50" model is a deep convolutional neural network that does not use Transformers to classify images. Instead, it explores the concept of using residual learning to train deeper models [4].

D. Image To Text Models

The image-to-text models use both image and text Transformers to recognize the words inside an image and print them out. Li et al. [6] pioneered the development of the "Transformer-based Optical Character Recognition" (TrOCR) model. To optimize its performance, these models have been fine-tuned using the Scanned Receipts OCR and Information Extraction (SROIE) dataset. We evaluate two TrOCR models presented by Microsoft in our study. The "trocr-base-printed" model is an encoder-decoder model which uses an image and

 ISBN: 978-1-68558-097-1

text Transformer to scan an image with text in it and prints a string of the word/phrase in the image. This is the base sized model for the Microsoft trocr-image-to-text models [6]. The "trocr-large-printed" model is the large sized model for the microsoft trocr-image-to-text models [6].

IV. Methodology

In this section, we describe our methodology for evaluating the quality of different AI models, measuring the energy usage and carbon emission of each AI model, and carbon aware deployment in the cloud.

A. Model Quality Evaluation

While reducing the carbon footprint of AI models is essential, it should not come at the expense of significantly compromising the quality of these models. Therefore, we carefully evaluate the quality of each AI model and only opt for smaller models when they can match or surpass the capabilities of larger models.

Specifically, the accuracy of each image classification model is evaluated by obtaining a random subset of 100 images from the CIFAR-10 dataset, which consists a set of 32x32 images that are labeled with one of the 10 categories: "airplane", "automobile", "bird", "cat", "deer", "dog", "frog", "horse", "ship", or "truck". Please refer to Figure 2 and Table 1 for the output of different models on a sample image of an automobile from CIFAR-10.

Fig. 2. Sample image of an "automobile" from the CIFAR-10 dataset

TABLE I
SAMPLE-IMAGE CLASSIFICATION MODEL OUTPUTS

Model	*Output*
Google-vit-base-patch16-384	moving van
Google-vit-base-patch16-224	sports car
Microsoft-cvt-13	beach wagon
Microsoft-resnet-50	cassette player

The accuracy of the small and large image to text models are evaluated by whether or not the model outputs the correct word in the image. We use 20 images downloaded from the internet with words in the image and finding the percentage of correct outputs of the 20 test images.

Assessing the quality of Text to Text Summary and Image Generation models can be challenging due to subjectivity, leading to varying ratings from different individuals. When comparing the outputs of three models prompted with "a photo of a beautiful desert landscape at night" (see Figure 3), it is evident that all three Stable Diffusion models successfully generated a desert landscape. However, the "stable-diffusion-v1-5" model (small) failed to produce a nighttime image and displayed several odd streaks of yellow throughout the image. Additionally, the resolution of the image generated by the "stable-diffusion-2-1" model (large) was significantly higher (768x768) than the other two images (512x512). In general, larger models for Text to Text Summary and Image Generation tend to produce higher quality outputs compared to smaller models.

Fig. 3. Images generated by CompVis/stable-diffusion-v1-4, runwayml/stable-diffusion-v1-5, and stabilityai/stable-diffusion-2-1 from left to right

B. Energy Usage and Carbon Emission Measurement

As AI models continue to gain widespread use across various sectors and industries, their environmental impact has grown significantly. Therefore, it is important to quantitatively measure both the energy used and the carbon emissions produced by training and deploying AI models. We leverage CodeCarbon [15] as a valuable tool for monitoring carbon emissions associated with various AI models. CodeCarbon provides a user-friendly API that facilitates the tracking of energy consumption and carbon emissions of different AI models. CodeCarbon enables us to monitor the power usage of underlying hardware components, such as GPUs and CPUs, at regular time intervals. In our study, we express carbon emissions in kilograms of CO2-equivalent per kilowatt-hour, and the power consumption is measured using the default sampling rate of 15 seconds.

C. Carbon Aware AI Deployment

When large models provide superior performance than smaller models, we propose to reduce their carbon footprint by deploying large AI models in regions with low carbon intensity. The carbon intensity of the consumed electricity is determined by taking into account the emissions from various energy sources used for electricity generation, encompassing both fossil fuels and renewables. The carbon intensity considers fossil fuels such as coal, petroleum, and natural gas, each linked to specific carbon intensities, signifying the amount of carbon dioxide released per kilowatt-hour of electricity produced. On the other hand, renewable or low-carbon fuels like solar power, hydroelectricity, biomass, and geothermal are also factored in.

Table II presents the Grid Carbon Intensity data provided by Google in grams of CO2 equivalent per kilowatt-hour (gCO2eq/kWh) for various cloud regions/locations [7]. The carbon intensity values indicate the amount of carbon dioxide equivalent emissions produced per unit of electricity consumed in each region. Lower carbon intensity values suggest that the electricity generation in those regions is more environmentally

 ISBN: 978-1-68558-097-1

friendly and emits fewer greenhouse gases. Looking at the data, we can observe that Toronto, Paris, Finland, Madrid, Oregon, London, and Belgium have relatively low carbon intensities. These regions appear to have a strong focus on renewable energy sources or nuclear power, resulting in significantly reduced carbon emissions. On the other hand, Mumbai, Sydney, Melbourne, Salt Lake City, South Carolina, and Warsaw have relatively high carbon intensities. These regions might be relying heavily on fossil fuels for electricity generation, leading to higher emissions.

TABLE II
GRID CARBON INTENSITY FOR DIFFERENT REGIONS

Cloud Region/Location	*Grid Carbon Intensity(gCO2eq/kWh)*
Taiwan	456
Hong Kong	360
Tokyo	464
Mumbai	670
Singapore	372
Sydney	598
Melbourne	521
Warsaw	576
Finland	127
Madrid	121
Belgium	110
London	172
Paris	59
Toronto	29
São Paulo	129
Iowa	394
South Carolina	434
North Virginia	309
Dallas	296
Oregon	60
Los Angeles	190
Salt Lake City	448
Las Vegas	365

In our experiments, we measure the energy consumption in kilowatt-hours (kWh) and the run time in seconds for each model. Then, using the carbon intensity data, we estimate the carbon emissions in various regions. The variation in carbon emissions across regions is multiplied by the actual usage of the AI models, allowing us to assess the potential CO2 savings.

V. EXPERIMENTAL RESULTS

In this section, we will present experimental results and the cloud platform that we use to deploy these AI models. Specifically, Subsection A discusses the cloud deployment details. Subsections B and C present the carbon reduction results of replacing larger models with small models. Subsections D and E illustrate the penitential carbon savings when deploying AI models on low carbon regions.

A. Cloud Deployment

In our experiments, all AI models are assessed using the A10 GPU within the Lambda Cloud [16], which provides us with instant access to cloud GPUs at highly competitive prices. The Lambda Cloud follows a pay-by-the-second billing model, ensuring that users are charged only for the actual time their instances are utilized. Furthermore, Lambda cloud pre-installs popular machine learning frameworks like TensorFlow, PyTorch, CUDA, and cuDNN, enabling us to promptly deploy models without any installation hassles. Additionally, the Lambda cloud supports deployment in multiple regions, each with varying carbon intensity levels. This feature allowed us to examine the efficacy of our proposed carbon-aware AI deployment approach.

B. Image Classification Models

We evaluate four image classification models, including “Google-vit-base-patch16-224” [2], “Google-vit-base-patch16-384” [2], “Microsoft-cvt-13” [3], and “Microsoft-resnet-50” [4]. The accuracy and energy consumption of all four models are presented in Table III, from which we can observe that two Google-vit models perform significantly better than the other two image classification models. However, they also consume significantly more energy. Surprisingly, even though the ”Google-vit-base-patch16-224” model uses 69% less electricity than the larger ”Google-vit-base-patch16-384” model, it achieves a higher accuracy level. This finding suggests that utilizing the ”Google-vit-base-patch16-224” model not only leads to better model quality but also generates less than one-third of the carbon emissions produced when using the ”Google-vit-base-patch16-384” model.

TABLE III
IMAGE CLASSIFICATION MODEL ACCURACY AND ENERGY USAGE

Model	*Accuracy*	*Energy Consumption (kWh)*
Google-vit-base-patch16-384	61%	0.1229
Google-vit-base-patch16-224	68%	0.0381
Microsoft-cvt-13	49%	0.0122
Microsoft-resnet-50	39%	0.0062

C. Image to Text Models

We analyze two image to text generation models, namely “trocr-base-printed” and “trocr-large-printed”. Both models exhibit nearly identical accuracy levels, with 96.59% for the large model and 96.37% for the base model [6]. However, our experiments reveal a significant disparity in energy consumption and processing time between the two models. Specifically, the base model outperform the large model in terms of energy efficiency and processing speed. For instance, when converting the image shown in Figure 4 to text, the large model takes 17.6 seconds and consumes 0.000585 kWh of energy, whereas the same task is accomplished by the base model in only 14.4 seconds, consuming just 0.000403 kWh of energy. Similarly, Figure 5’s image conversion is 18% faster and result in 22% energy savings when using the base model with no compromise on output quality.

These findings challenge the prevailing notion that larger models are inherently superior to smaller ones. In cases where both models achieve comparable accuracy, opting for the smaller model proves to be more efficient. Larger models not only demand more energy for running and training but

 ISBN: 978-1-68558-097-1

also take longer to process data. As such, the superiority of larger models is not universal, and in certain scenarios, smaller models can perform equally well, offering the additional advantages of reduced energy consumption and processing time.

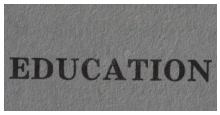

Fig. 4. Image with the text "Education" used to test the Text-to-Text Models

Fig. 5. Image with the text "Advantage" used to test the Text-to-Text Models

D. Carbon-Aware Deployment of Stable Diffusion Models

We evaluate three popular Stable Diffusion models including "stable-diffusion-v1-4", "stable-diffsion-v1-5, and "stabilityai/stable-diffusion-2-1" [5]. As previously discussed in Section III and illustrated in Figure 6, images generated by the large "stabilityai/stable-diffusion-2-1" model have much higher quality than the other two smaller models. Nevertheless, the "stabilityai/stable-diffusion-2-1" model consumes 4 times more energy than the other two smaller models, as illustrated in Table IV.

Fig. 6. An image of flying pigs generated by different Stable Diffusion models

TABLE IV
STABLE DIFFUSION MODEL ENERGY CONSUMPTION

Model	*Energy Consumption (kWh)*
stabilityai/stable-diffusion-2-1	0.01438
runwayml/stable-diffusion-v1-5	0.00321
CompVis/stable-diffusion-v1-4	0.00323

We leverage a carbon-aware solution to balance the need for maintaining model quality while minimizing carbon emissions. This solution uses the high quality "stabilityai/stable-diffusion-2-1" model but strategically deploys it to regions with lower carbon intensity, thereby mitigating environmental impact without compromising the quality of models.

This approach works because we now can easily deploy AI models at different regions using cloud computing. For example, deploying the "stabilityai/stable-diffusion-2-1" model at the "us-west4" region (based in Salt Lake City) results in a carbon intensity of approximately 448. In contrast, the "us-west1" region (based in Oregon) has a much lower carbon intensity of 60. The notable difference in carbon intensity is attributed to Oregon's utilization of renewable electricity sources, such as Hydro Power and Wind Power [17], while "us-west3", where Salt Lake City is located, relies significantly on fossil fuels like natural gas and coal to generate electricity [18].

To demonstrate the impact of carbon aware deployment in the cloud on carbon emissions, we conduct experiments with Stable Diffusion and estimated the carbon emissions for both "us-west3" and "us-west1." Although the energy consumption of individual requests might not seem substantial, considering Stable Diffusion models serve 10 million daily users worldwide, the accumulated energy and carbon savings become noteworthy. Assuming each user generates one image, we are looking at 10 million images being produced daily. Our experimental results indicate that this process would consume approximately 14,380 kWh of electricity per day. If the "stabilityai/stable-diffusion-2-1" model were to generate all its images using "us-west3," it would produce 6.4 million gCO2eq of greenhouse gases. On the other hand, if deployed on "us-west1," it would emit only about 860 thousand gCO2eq. By making environmentally conscious decisions about where the model is deployed, we manage to reduce carbon emissions by an impressive 86.6%.

E. Carbon-Aware Deployment for Summary Models

In this experiment, we evaluate two Text-To-Text Transfer ransformer summary models: "t5-one-line-summary" and "t5-base-finetuned-summarize-news" [1]. It is evident that the large "t5-base-finetuned-summarize-news" model generally provides higher quality summary than the small "t5-one-line-summary" model. Similarly, we can reduce its carbon footprint by deploying the large model in regions with low carbon intensity.

It is worth noting that the carbon-aware deployment approach can also benefit small AI models. Table V presents the carbon emissions of both small and large models when deployed in different regions. By choosing Oregon over Dallas as the deployment location for the "t5-one-line-summary" model, a single request can save 0.084 grams of CO_2, amounting to an 80% reduction. With an estimated usage of 466,000 times a month, the carbon aware deployment approach could save at least 39,144 grams of CO_2. Similarly, the "t5-base-finetuned-summarize-news" model could save a total of 106,533 grams of CO_2.

Since carbon intensity varies throughout the day due to electricity demand, computational tasks could switch between different cloud locations to optimize carbon savings.

 ISBN: 978-1-68558-097-1

TABLE V
ESTIMATED CO2 EMISSION IN DIFFERENT CARBON INTENSITY REGION

Region	***"one-line-summary"***	***"finetuned-summarize-news"***
Taiwan	0.162	0.771
Hong Kong	0.128	0.608
Tokyo	0.165	0.784
Mumbai	0.238	1.13
Singapore	0.132	0.629
Sydney	0.213	1.01
Melbourne	0.185	0.881
Warsaw	0.205	0.973
Finland	0.045	0.215
Madrid	0.043	0.204
Belgium	0.039	0.186
London	0.061	0.291
Paris	0.021	0.010
Toronto	0.010	0.049
São Paulo	0.046	0.218
Iowa	0.140	0.666
South Carolina	0.154	0.733
North Virginia	0.110	0.522
Dallas	0.105	0.500
Oregon	0.021	0.101
Los Angeles	0.068	0.321
Salt Lake City	0.159	0.757
Las Vegas	0.130	0.617

VI. CONCLUSIONS AND FUTURE WORK

The growing adoption of AI models has led to a notable rise in energy consumption and carbon emissions associated with their training and inference processes. Despite this concern, research on sustainable AI is still in its early stages. This study aims to investigate efficient approaches that can diminish the carbon footprint of AI models without sacrificing their performance.

We propose two methods to mitigate CO2 emissions while utilizing AI models. Firstly, by employing more energy-efficient models where feasible, we showcase instances where smaller AI models, consuming less energy, could deliver comparable or even superior performance to larger, more energy-intensive counterparts. Secondly, we advocate for a carbon-aware deployment of AI models. The geographical location where AI models are executed significantly influences their carbon intensity, as the carbon emissions generated during electricity production depend on the carbon intensity of the local energy grid. Adopting low carbon-intensity cloud services for running AI models can substantially reduce the carbon footprint of AI applications. This approach is applicable to both AI training and inference, thereby reducing the carbon emissions associated with electricity consumption. By implementing these carbon-reduction strategies, we can harness the power of AI for societal benefits while ensuring AI's carbon emissions remain sustainable. Our findings indicate that the utilization of smaller models can potentially reduce energy usage by up to 69% in specific scenarios, and the second method aligns AI's carbon footprint with the carbon intensity of the least carbon-intense cloud computing server.

Despite our paper's valuable contributions, we acknowledge certain limitations. The accuracy and energy usage measurements of AI models may not be entirely precise due to a relatively small sample size. To enhance our research, conducting in-depth experiments with a larger dataset could more accurately determine the models' accuracies. Moreover, future work should focus on quantifying the quality of both image generation models and summary models, enabling a more comprehensive comparison of their accuracies and energy usage. Another limitation pertains to the carbon intensity data, which is based on average values and may not account for real-time fluctuations. Carbon emissions during electricity production can vary due to several factors. To minimize carbon emissions more effectively, AI models might need to dynamically adapt to different cloud regions. Therefore, further investigation is necessary to further improve this approach.

Large computing organizations might see a more pronounced reduction in CO2 emission since they have more resources to optimize cloud usage and a wider range of AI models to choose from, but these solutions can be effective even on a small scale.

In conclusion, our study emphasizes the urgency of addressing the environmental impact of AI and presents viable strategies for reducing carbon emissions of employing AI models without compromising model quality. Embracing energy-efficient models and adopting carbon-aware deployment practices will contribute to a more sustainable and environmentally friendly integration of AI technology into our society.

REFERENCES

[1] C. Raffel, et al., "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer", pp. 11-41, 2021. https://www.jmlr.org/papers/volume21/20-074/20-074.pdf. Accessed Aug 10, 2023.

[2] A. Dosovitskiy et al., "An Image Worth 16x16 Words: Transformers for Image Recognition at Scale", pp. 3-7, 2021. https://arxiv.org/pdf/2010.11929.pdf. Accessed Aug 10, 2023.

[3] H. P. Wu, et al., "CvT: Introducing Convolutions to Vision Transformers", pp. 4-6, 2021. https://arxiv.org/pdf/2103.15808.pdf. Accessed Aug 10, 2023.

[4] K. M. He, X. Y. Zhang, S. Q. Ren, and J. Sun, "Deep residual learning for image recognition", in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778, 2016.

[5] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models", in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10684-10695, June 2022.

[6] M. H. Li et al., "TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models", https://arxiv.org/abs/2109.10282v5, 2021. Accessed Aug 10, 2023

[7] https://cloud.google.com/sustainability/region-carbon. Accessed Aug 10, 2023.

[8] D. Amodei and D. Hernandez, "AI and compute", https://openai.com/research/ai-and-compute, 2018. Accessed Aug 10, 2023.

[9] J. Huang et al., "Speed/accuracy trade-offs for modern convolutional object detectors", published in Proceedings of CVPR, 2017.

[10] K. Hao, "Training a single AI model can emit as much carbon as five cars in their lifetimes", published in MIT Technology Review, pp. 8-14, Jun. 2019.

[11] D. Mahajan et al., "Exploring the limits of weakly supervised pretraining", https://arxiv.org/abs/1805.00932, pp. 7-13, 2018. Accessed Aug 10, 2023.

[12] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, "Green AI", pp. 5-9, Dec. 2020.

 ISBN: 978-1-68558-097-1

[13] E. Strubell, A. Ganesh, and A. McCallum, “Energy and Policy Considerations for Deep Learning in NLP”, https://arxiv.org/abs/1906.02243, pp. 1-5, 2019. Accessed Aug 10, 2023.
[14] C. J. Wuet al., “Sustainable AI: Environmental Implications, Challenges and Opportunities”, https://arxiv.org/abs/2111.00364, pp. 1-8, 2021. Accessed Aug 10, 2023.
[15] Codecarbon https://codecarbon.io/. Accessed Aug 10, 2023.
[16] Lambda Cloud https://lambdalabs.com/service/gpu-cloud. Accessed Aug 10, 2023.
[17] Oregon Department of Energy, “2020 Biennial Energy Report”, 2020.
[18] U.S. Energy Information Administration, “Utah State Profile and Energy Estimates”, 2023

 ISBN: 978-1-68558-097-1

Comparable Machine Learning Efficiency: Balanced Metrics for Natural Language Processing

Daniel Schönle
IDACUS Insitute
Furtwangen University
Furtwangen, Germany
daniel.schoenle@hs-furtwangen.de

Christoph Reich
IDACUS Insitute
Furtwangen University
Furtwangen, Germany
christoph.reich@hs-furtwangen.de

Djaffar Ould Abdeslam
Institut IRIMAS
Université de Haute Alsace
Mulhouse, France
djafar.ould-abdeslam@uha.fr

***Abstract*—As machine learning becomes increasingly pervasive, its resource demands and financial implications escalate, necessitating energy and cost optimisations to meet stakeholder demands. Quality metrics for predictive machine learning models are abundant, but efficiency metrics remain rare. We propose a framework for efficiency metrics, that enables the comparison of distinct efficiency types. A quality-focused efficiency metric is introduced that considers resource consumption, computational effort, and runtime in addition to prediction quality. The metric has been successfully tested for usability, plausibility, and compensation for dataset size and host performance. This framework enables informed decisions to be made about the use and design of machine learning in an environmentally responsible and cost-effective manner.**

***Index Terms*—machine learning; nlp; efficiency; metric; software performance; automl.**

I. Introduction

Decades ago, the primary motivation behind the pursuit of computational efficiency was the limited computing power available at the time. Computing resources had to be used judiciously to overcome the constraints imposed by hardware limitations. The advent of powerful computing resources, particularly in the field of Machine Learning (ML), has shifted the focus to achieving superior prediction quality, relegating efficiency to a secondary concern. As ML continues to evolve, the future landscape of human-computer interaction will be profoundly influenced by the widespread adoption of Large Language Models (LLMs) [1]. This shift is also driven by the integration of ML into heterogeneous computing environments, such as edge computing on resource-limited hardware. From a green computing and sustainability perspective, the use of resource-intensive solutions such as transformer-based word embeddings or LLMs may not always be financially or environmentally viable [2]. As a result, there is a growing demand for efficiency, especially for the widespread application of ML. The objective of this publication is to contribute to the improvement of efficiency in ML by introducing robust metrics for measuring the efficiency of machine learning models.

ML research has focused on improving model quality, for which a number of metrics are available. Research on effective ML lacks standardised and comprehensive efficiency metrics. To ensure reproducibility and facilitate result comparison, best practices in ML research typically include detailed descriptions of experiments, encompassing datasets, preprocessing steps, machine learning techniques, hyper-parameters, and hardware setups [3]. The absence of a dataset-agnostic procedures and missing 'Golden Standard' datasets pose challenges in achieving true repeatability and fair comparisons in Natural Language Processing (NLP) [4]. Furthermore, evaluating the impact of novelties in ML process steps, such as improved preprocessing, on prediction quality is a complicated task, as the other ML steps may be influential.

The delicate balance between complexity and outcome is often overlooked in research efforts to utilise all available resources to reduce time-to-solution. Evaluating time and space complexity becomes subordinated to finding the best model or process. Numrich stated [5]: "Increasing productivity by minimising the total-time-to solution is a somewhat ill-defined statement of the problem. We propose an alternative statement: at each moment in time, use the resources available in an optimal way to accomplish a mission within imposed constraints." It becomes essential to establish metrics that address the efficiency concerns alongside prediction quality in the context of ML research.

We present a proposal to fill the existing gap by introducing novel metrics for measuring the efficiency of machine learning models. By incorporating resource consumption, computational effort, and runtime considerations into our efficiency metrics, we aim to provide a holistic perspective on the true efficiency of ML models. We demonstrate the process of defining a quality focused efficiency metric (Figure 1) and present the Quality COmpact (QCO) Efficiency Metric (Equations 4 & 5). We recognise the importance of dataset-agnostic evaluation and propose solutions to address this challenge and demonstrate the advantages of our metric for evaluating hyperparameter tuning. Our goal is to empower researchers and practitioners to make informed decisions that prioritise both prediction quality and efficiency, thus advancing the field of ML towards sustainable, green, and economically feasible solutions.

The structure of the paper is as follows: Section 2 (State of the Art) covers the research on efficiency types and metrics in ML. In Section 3 the efficiency metric is presented by elaborating on its objectives, followed by the theoretical foundations of *efficiency dimensions* and concepts, and finally the

 ISBN: 978-1-68558-097-1

definition of the efficiency metrics. In the subsequent Section 4, the metric for quality-focused efficiency is defined, adhering to a specified protocol. The score equations for QCO_F are presented, accompanied by a brief explanation of its usage. The Evaluation Section 5 uses two experiments to assess the performance of the metric. The results obtained are discussed in detail in Section 6, leading to the presentation of the conclusion (Section 7).

II. State of the Art

This section begins with approaches that deal with computational cost as a method. The goal of predicting computational cost incorporate with the prediction of financial cost. Then, approaches to resource effectiveness are considered. Their goal is to find algorithms that work in most cost-effective way. Finally, general approaches to efficiency metrics are examined.

Computational Cost or efficiency is based on the computational effort. Most statistical ML algorithms can be addressed and their time complexity or space requirements can be calculated. For example, the time complexity of gradient descent is $O(ndk)$, where d is the number of features and n is the number of rows. In the context of transformer-based approaches, the number of operations for multi-head attention can be calculated as n^2d+nd^2, where n is the sequence length and d is the depth [6]. Translating these statistical calculations into real training times is challenging due to numerous optimisations of modern CPUs and GPUs that change the type of computation and the number of operations [7][8][9]. The approach presented here defines work and duration dimensions based on actual measurements.

Computational Cost for Deep Learning is specific to deep learning, as it relies on complex neural network architectures, which makes direct computation of complexity difficult. Several approaches attempt to predict complexity, such as the proposal by Li et al. [10], which introduces two classes of prediction models for distributed SGD. The use of profiling information in this approach is similar to the method presented here, but with limited validity for deep learning optimised with distributed SGD.

Resource Efficiency is important for deep learning, where hardware requirements differ from those of statistical machine learning and are constantly evolving. Research aims to adapt deep learning to specific hardware. Yang et al. [11] developed a method to bridge this gap, focusing on computing the model locally near the sensor. In HPC, research such as Performance Metrics based on computational action (Numrich [5]) optimises the use of hardware. Resource efficiency focuses primarily on the optimal hardware usage of specific algorithms, ignoring algorithm complexity or runtime. The efficiency definition presented here addresses this aspect to provide comprehensive statements about the entire ML task.

Efficiency Comparison plays a role in the evaluation of novel approaches. For instance, Thomson et al. [12] present an optimisation for machine learning-based compilers that focuses on process speedup while overlooking the impact on resource consumption. Fischer et al. [13] propose a framework

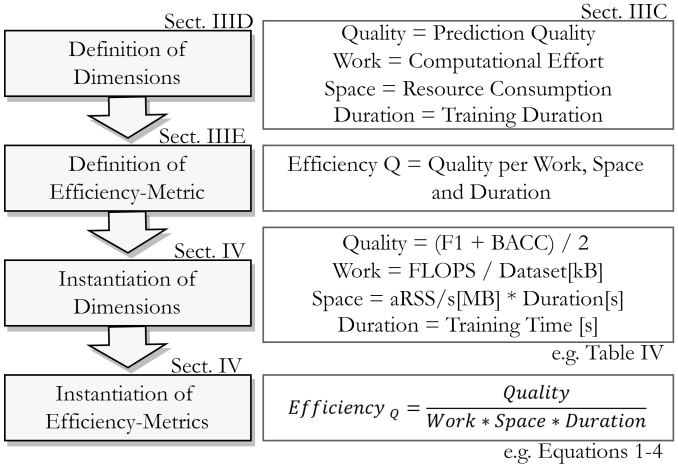

Fig. 1. Development of Quality Focused Efficiency Metric.

for evaluating the energy efficiency of ML without considering prediction performance. Kumar, Goyal & Varma [14] develop ML with a small footprint and compares efficiency based on model size, prediction quality, prediction time and prediction energy. Discussions of the novel approach primarily revolve around individual measurements, lacking an overall efficiency comparison. In contrast, Huang et al. [15] discusses the selection of an object detection architecture in terms of efficiency, defining it as a speed/memory/accuracy trade-off and evaluating it through two-dimensional trade-off curves. The proposed efficiency metric would provide a balanced and meaningful score for evaluating [14] and [15].

Efficiency metrics were 'invented' for HPC research, which deals with highly scaled hardware systems and highly specialised applications, making efficiency statements easier to derive and crucial. The difficulties have been recognised and discussed from an early stage [16] - to philosophical considerations [17]. Numrich of Cray Research developed an approach based on physical laws [18] [5], which inspired the proposed metric based on dimensions reflecting components of a physical law.

III. Efficiency Metric Proposal

This proposal encompasses two integral parts: the development of abstract efficiency metrics and the definition of the quality-focused efficiency metric. We introduce the objectives, limitations, and use cases of efficiency in ML, and establish basic efficiency types based on trade-off relationships. Drawing inspiration from the laws of physics, we define efficiency using *efficiency dimensions* for quality, work, space, load, and duration of the ML procedure, with each dimension comprising measurements of the ML process. We outline two types of metrics, namely the *efficiency vector*, which provides insight into the raw strengths and weaknesses of the ML process in terms of efficiency, and the *focused efficiency scores*, which are designed for ease of interpretation. To enhance the significance of scores, a defined procedure is employed to adjust dimensional weights and perform sophisticated measurement

TABLE I
LIMITATIONS OF *QUALITY COMPACT* (QCO) METRIC

Aspect	Valid Range or Category	Balanced
Dataset	Labelled Text Samples, Size <256 MB	Yes[1]
ML-Task	Text Classification, NLP-Tasks[4]	No[2]
Classifiers	All ML Techniques	No[3]
Host-setup	Non-HPC, Non-GPU, RAM <128GB	Yes[1]
Training Duration	48h	Yes[1]
Calculation Amount	63P FLOPS, 800M Minor Page Faults	Yes[1]

(1) Compensation, e.g., efficiency remains consistent regardless of dataset size. (2) Scores from different tasks are not comparable. (3) Provides comparable efficiency scores per ML technique. (4) untested

smoothing. As an example, we outline the metric equation for quality-focused efficiency and propose a metric definition protocol to achieve metric validity. This protocol is applied to define the quality-focused efficiency metric, including the definition of the equation for score calculation, appropriate measurement selection, smoothing of measurement values, and dimension weight development.

A. Objectives & Limitations

An objective of this approach is to enable its applicability to all ML techniques. Measurements should be available for common host setups. The efficiency should be balance effects of different host setups. Additional requirements need to be derived from use cases. Certain measurements depend on ML task characteristics, such as dataset size, or runtime conditions, such as duration (see Table I). Valid measurement ranges may be enforced by smoothing techniques.

The application of the ML process involves objectives and constraints. The following use cases have specific efficiency requirements.

1) Effects of changes in the ML process: The effects of different techniques, such as preprocessing techniques, need to be measured [19].
2) Select ML technique by efficiency: Identify the ML technique that achieves high classification quality while minimising the use of computational resources. [11].
3) ML technique for limited resources or private data: Select a Whitebox ML technique suitable for local model training [20].
4) Parameter optimisation: Effectiveness as a cost function in the optimisation of hyperparameters or setups [21].
5) Performance comparison: Compare the performance of an ML technique on different host setups to evaluate ML efficiency [22].
6) Predicting computational costs: Predicting the cost by predicting computational effectiveness of an ML technique in a production setup [23].

B. Dimensions

The concepts of efficiency in the use cases are different, but are defined on the basis of similar components. All concepts consider the trade-off between the performance of the model and the resources consumed during training or inference. The key components under consideration for the efficiency metrics are:

Accuracy or Performance. The efficiency of the machine learning model is correlated with its accuracy or performance. Standard metrics such as accuracy, precision, recall, F1-Score, or Area Under the ROC curve (AUROC) can be used to measure this, depending on the specific task.

Resource Utilisation. Efficiency should take into consideration the resources consumed during the training or inference process. This includes computational resources like CPU, GPU, or memory usage. Efficient models should minimising resource utilisation.

Relative Resource Utilisation. The load imposed on the host by the machine learning process provides a means of measuring the relative utilisation of hardware resources. A higher load indicates a more efficient use of available resources, as fewer resources are left unused.

Computational Effort. Efficiency is affected by the complexity of the ML process, or the amount of computation needed to train the model or compute a result for inference. Efficiency is improved when the computational effort is minimised.

Training Duration. The definition of efficiency can include the time to train the machine learning model. Faster training times can be beneficial, especially in scenarios where models need to be trained frequently or where time constraints exist.

Inference Latency. For models deployed in real-time or interactive applications, the time taken to make predictions or perform inference is critical. Low inference latency or fast response times can be important efficiency metrics in such cases.

In the context of cost-effectiveness in machine learning research, different dimensions or base units are considered. The need to define base units, such as distance and power, which can be used to define efficiency, has been discussed by Numrich [24]. This approach uses abstract dimensions, which provides adaptation through flexible adaption. The proposed efficiency metric uses the following *efficiency dimensions*, with a description of valid measurements:

Quality. (Or Performance) The machine learning model should achieve the desired level of accuracy as a performance indicator for addressing the given task or problem. Measurement can be conducted using appropriate evaluation metrics tailored to the specific task, including accuracy, precision, recall, F1-Score, or AUROC. Preferably scores that compensate unbalanced dataset [6].

Work. (Or Computational Effort, Computational Complexity) The number of computational operations, such as matrix multiplications, gradient computations, data transformations as well as the usage of computational-cache (e.g., CPU L1-Cache). The theoretical amount of work can be calculated by applying the theory of computational complexity. The real workload differs due to optimisation at the software and hardware level. [7]–[9]. The measurement shall count generated and processed compute steps,

optionally data transfers through memory and network. Computational steps can be counted direct (floating point operations or instructions) [6] or indirect by measuring side-effects of computation, e.g., memory management activity.

Load. (Or Relative Resource Consumption) Relative host usage reflects the $degree$ to which all available resources on the host are being used. This includes relative usage of compute units (CPU and GPU cores), and relative memory usage. It also includes information about load related memory management events such as major page faults.

Space. (Or Absolute Resource Consumption, Space Complexity) The $amount$ of data resources, such as memory and storage, needed by the machine learning process. Space usage of memory is measured by resource usage on the host system. This includes main memory usage and allocation such as virtual memory allocation, resident set size, working set size or stack size.

Duration. (Or Time Requirements) Time-related measurements, such as training time or inference latency and include time to complete the ML procedure or time spent on processing units.

Other non-dimension specific measures include the characteristics of the dataset, such as information about the number of samples and the size of the dataset. For special purpose metrics, sample attributes such as number of sentences, number of words and linguistic text attributes can be obtained.

C. Efficiency

Efficiency (Cost-Effectiveness) refers to achieving a high level of performance or accuracy while optimising the utilisation of resources and minimising associated costs. It aims to strike a balance between the effectiveness (performance) of the model and the costs or resources required to achieve that effectiveness. This approach covers three concepts of cost-effectiveness:

Solution Efficiency. Efficiency as the balance solution achievement and cost. Solutions are focuses like quality, costs include efforts done and resources consumed. Every aspect is provided by one or multiple efficiency dimensions. Solution efficiency with quality focus describes how much computational effort was used to achieve the prediction quality. This reflects the efficiency of the model, i.e., the algorithm and its implementation. Efficiency increases by doing less work in less time and achieving higher prediction quality. Other focuses is achieving low latency of ML inference.

Resource Efficiency. Efficiency as the degree to which resources are used. Resource efficiency is the capability of the ML procedure to use all available resources. It increases by adapting to host setup by using more existing resources. Important for designing hardware for specific ML Techniques and adapting ML algorithms to specific hardware [25]

TABLE II
INSTANTIATION PROTOCOL

Step	Objective
1	Select Efficiency Metric
2	Define Validity Requirements
3	Setup and Conduct Experiment
4	Define Dimensions
5	Analyse measurements
6	Assign measurements to Dimensions
7	Define Validity Ranges
8	Normalisation of Measurement-Values
9	Determine Dimensional Weights
10	Define Score compensation factor
11	Define Score Equation

TABLE III
VALIDITY REQUIREMENTS

Aspect	Count	Variables	Optional
Dataset	>=2	Size, Sample Count	Sample Length, Language
Vectorization	>=2	Algorithm	Dictionary Size, Model Size
Classifier	>=4	Algorithm, Classifier Tech.	Hyperparameters
Host-Setup	>=2	Hardware Conf., Operating System	Software Version

Synthetic Efficiency. Efficiency as a tool for measuring special aspects of performance to analyse specific attributes, such as text quality indicators [26] or performance comparisons [27].

Efficiency rules are defined based on the efficiency objectives:

1 Solution Efficiency
  1.1 The more quality is achieved in less time, work and effort, the higher the ML quality efficiency.
  1.2 The less time it takes to achieve more quality, the higher the ML-Speed-Efficiency.
  1.3 The less work required for more quality, the higher the ML-Work-Efficiency.

2 Resource efficiency
  2.1 The more load is used for more quality, less duration, less work, the higher the ML resource efficiency.

3 Synthetic efficiency
  3.1 The less computational work is necessary per data chunk the higher the ML model efficiency.

Beside the efficiency objectives, two diametral requirements on handling of the ML efficiency results are encountered: Interpretability and Usability. The more information a metric provides, the greater the need for interpretation. This approach provides metrics at two levels of complexity. (i) Efficiency is determined as a single scalar by the at-a-glance metric (compact metric score) while supporting weights for each dimension. (ii) The efficiency vector metric represents uninterpreted values per dimension.

D. Compact Efficiency Metrics

Efficiency in the field of ML shows variability depending on the specific application. Metrics are proposed for specific purposes and categorised according to their level of complexity. The group of compact metrics uses a subset of dimensions that contribute to the calculation of an efficiency score, which is determined with respect to the dominant dimension. The compact efficiency (CO) metric is defined in Definition 1.

Definition 1: It exists a compact efficiency score CO of a ML procedure M for focused dimensions F with a focus weight α and unfocused dimensions U (1); based on efficiency dimensions (D) quality q, work w, space s, load l and duration d with specific dimension weights β defined by (2).

$$[F]CO(M) = (F \times \alpha) \times U \tag{1}$$

$$[F]CO(M) = (q_M \times \beta_q) \times (w_M \times \beta_w) \times (s_M \times \beta_s) \times (l_M \times \beta_l) \times (d_M \times \beta_d) \times \psi \tag{2}$$

$$\text{where } D = \{r \in \mathbb{R} \,|\, r > 1\} \; and \; \{q, w, s, l, d \in D\}$$
$$F \subseteq D \quad \text{and} \quad U = D \backslash F$$
$$\psi = \text{Score-Compensation}$$

Quality Focused COmpact Efficiency Metric (QCO). A compact metric to reflect quality-focused efficiency. A score describes the best solution with a predefined high relevance of the quality dimension and low relevance of the work and duration dimensions. Relevant dimensions: Quality, Work, Space, Duration. Dominant dimension: Quality.

The QCO-score for an ML process M is derived from (1) & (2) for the Quality-Focus, as stated by (3). The quality dimension is represent by q, which measures the quality or performance of the machine learning model. w represents the dimension of computational effort, which quantifies the computational operations or effort required for the machine learning tasks. The resource consumption dimension s measures the amount of system resources required during the execution of the model. d represents the dimension of duration, which measures the time or duration required to train the model. The weight per dimension β is employed to adjust the importance of dimensions, while α represents the additional weight of the focus dimension, both derived from expert knowledge of the use case. The compensation factor ψ is introduced to optimise the readability of the score, where $1 > \psi \geq 0.1$. The dominance of quality q is reflected in the numerator, so efficiency is defined as the quotient of quality divided by work w, space s and duration d (terms in the denominator). The dimensions are intended to increase in importance with a growth proportional to their current size, so the weights β of the dimensions and the focus weight α are treated as exponents with the respective dimension as the base.

TABLE IV
HOST-SETUPS

No.	Type	CPU-Model	Clock	Threads	RAM
1	Virtualised	AMD Ryzen 7 5800U	1,9	8	16
2	BareMetal	Intel Core i5-6200U	2,3	4	8
3	BareMetal	Intel Core i7-7700	3,6	16	32
4	Virtualised	Intel Xeon Gold 6230	2,1	4	8
5	Virtualised	AMD EPYC 7742	2,2	16	16

[Clock in GHz, RAM in GB.]
OS: Linux, Language: Python3,
Libraries: Scikit-learn [28], DistilBERT [29], torch [30], pandas [31].

$$QCO(M) = \frac{q^{\alpha * \beta_q}}{\left(w^{\beta_w} + s^{\beta_s} + d^{\beta_d}\right)} * \psi \tag{3}$$

Resource Focused Compact Efficiency Metric (RCO). Compact metric to reflect resource-oriented efficiency. A score describes the best solution with a predefined high relevance of the relative load usage and a low relevance of the quality dimension. Relevant dimensions: Load, Quality, Work, Duration. Dominant dimension: Load.

Inference Focused Compact Efficiency Metric (ICO). Compact metric to reflect resource-oriented efficiency. A score describes the best solution with a predefined high relevance of duration, low relevance of the quality dimension and lowest relevance of work. Relevant dimensions: Quality, Work, Duration. Dominant dimension: Duration.

Algorithmic Focused Compact Efficiency Metric (ACO). Compact metric to reflect resource-oriented efficiency. A score describes the best solution with predefined high relevance of work and duration, low relevance of duration, quality, and dataset dimension. Relevant dimensions: Quality, Work, Duration, Dataset. Dominant dimension: Work.

E. Efficiency Vector Metric (EV)

The CO metrics condense efficiency information into a score. To provide information of the dimension specific performance the EV metric reveals the dimension scores of the CO metric. The EV is available per CO as a vector, to describe the efficiency in the vector space of the specific CO. For QCO the QEV is represented by a vector in a *Quality-Work-Space-Duration* space.

IV. Compact Metric Instantiation

In order to use the proposed efficiency metric, the abstract definitions need to be instantiated into explicit definitions by empirical method (Table II). This requires conducting an ML experiment that maps a specific use case and involves the collection of measurements. The instantiation of the metrics thus depends on the parameters of the experiment. The validity of the metric instantiation is positively correlated with the size of the use case, such as the number of datasets.

The instantiation stages for a CO-Metric are as follows: The experiment is designed, deployed and measurements captured.

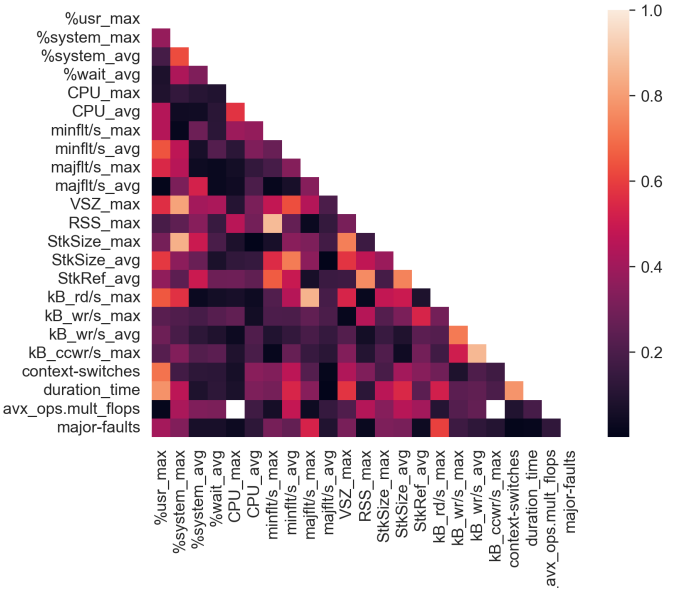

Fig. 2. Pearson Correlation Coefficients of empirical Measurement Values.

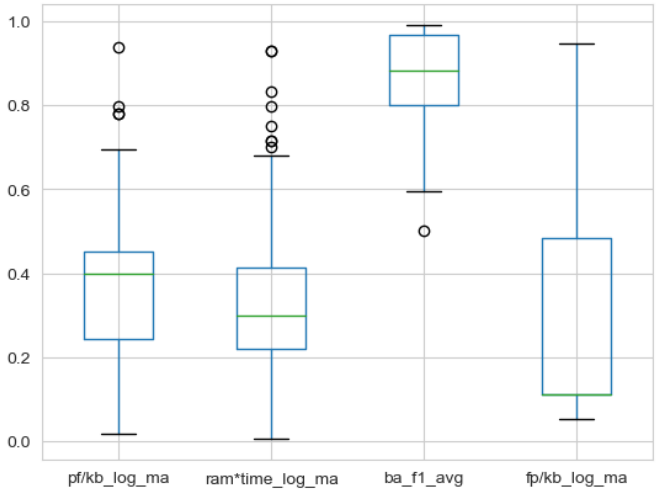

Fig. 3. Spread and Skewness per Dimension after logarithmic smoothing.

TABLE V
MEASUREMENTS

Type	DIM	IMP	TRANS	DEP
F1-Score	Quality	10	None	
Bal.-Acc.	Quality	10	None	
FLOPS	Work[CPU]	10	Log 63P	Data
MinorPF	Work[CPU]	5	Log 800M	Data
RSS (avg)	Space[Mem]	10	Log 128G	Time
CPU Time [ns]	Duration	10	Log 172T	
Data Size	-	10	Log 256M	

The formulas for the dimensions are defined and measured values are assigned (Table VI). Validity ranges are specified and the measured values are smoothed accordingly (Table VI). The determination of the dimensional weights and the score compensation factor ψ completes the Metric Equation (4).

For reasons of compactness, the instantiation is restricted to QCO metric.

A. The QCO-Metric Instance

The dimensions and the QCO score are instantiated according to the protocol given in Table II.

Use Case 1 requires efficiency as quality per work, space, and time. The QCO-Score is applicable. Experiment 1 has been set up based on Use Case 1 to instantiate a Quality

TABLE VI
QCO INSTANCES

Dimension	QCO_F	QCO_P(*)
Quality	(F1 + BACC) / 2	(F1 + BACC) / 2
Work	FLOPS / Dataset[kB]	Minor PF / Dataset[kB]
Space	aRSS/s[MB] * Duration[s]	aRSS[MB] * Duration[s]
Duration	Time on CPU [ns]	Time on CPU [ns]

(*) FLOPS-Measurement was not available on all hosts.

Focused Metric. The required validity for different datasets and ML procedures results in the empirical variance requirements presented in Table III. To gain comparison validity among host-setups, four different computing environments were set up (No. 1-4 as shown in Table IV).

Two datasets were selected, a spam classification [32] and am movie review classification [33]. Vectorization was done using the classical TF-IDF algorithm as wells as by word embedding based on BERT. The classifiers chosen were Support Vector Machines, Naïve Bayes (NB), Gradient Descent (GD) and Random Forest (RF). In addition, a transformer-based ML procedure (DistilBERT-Setup) was performed. In the DistilBERT-Setup, the model was fine-tuned on the datasets and used for vectorization and classification.

The measurements were provided by a set of Linux tools:

- `time`. Basic process measurement (CPU, Memory).
- `pidstat`. Advanced process measurement (CPU, Memory, IO-Usage).
- `perf`. Performance counter capture. (CPU, Memory).

Quality scores were computed separately from the ML procedure. Measurements were grouped for resource domain, e.g., memory consumption or computational work on CPU. The groups were filtered by correlation, the heatmap (Figure IV) shows Pearson Correlation Coefficients for selected measurements. `perf` was not supported on all host setups due to missing performance counters and conflicts with power saving methods. Two sets of measurements has to be set up which results in two QCO flavors: Floating Point Operation (QCO_F) based and Minor Page Fault (QCO_F) based. The selected measurements are listed in Table V.

Range definition is necessary for normalisation. Valid ranges for this QCO-Instance are listed in Table I. Normalisation is necessary as different units and types of data are used in calculation. Performing monotonic data transformation on dimensions values lead to a range between 0 to 2. The transformation range is based on the maximum values per

$$QCO_F(M) = \frac{((\frac{F1+BACC}{2})^6)}{(\log_{63P} FLOPS/DS[kB] + \log_{128G} RSS[MB] * log_{864M} D[s] + \log_{172T} TOC[ns]} * 10 \quad (4)$$

$$QCO_F(M) = \frac{((\frac{F1+BACC}{2})^6)}{(\log_{800M} MPF/DS[kB] + \log_{128G} RSS[MB] * log_{864M} D[s] + \log_{172T} TOC[ns]} * 10 \quad (5)$$

where $F1 =$ F1-Score, $BACC =$ Balanced Accuracy Score,
$FLOPS =$ Floating Point Ops., $MPF =$ Minor Page Faults,
$DS =$ Dataset-Size, $RSS =$ Resident Set Size,
$D =$ Duration, $TOC =$ Time on CPU

dimension. The valid ranges are thus not related to measurement ranges. The definition of valid measurement ranges (Table I) enables data transformations on measurement values. After transformation values are in an closed scale with minor decreased distribution (Figure 3).

The dimension-equations are defined by interpreting the dependencies of the measurements (Table V). Especially the dependency on duration and data size has been considered.

The dimension weight is used to adjust the importance of the focused metrics. The importance of quality is based on domain knowledge: Quality is about two times more important than work, space, and duration which delivers $\beta_Q = 6$. Readability compensation ψ is set to 10.

QCO for is defined for each measure set, which results in 4 and 5

B. QCO Metric Usage

1) Select QCO type according to available measurements. If CPU-Performance-Counters are available QCOF, otherwise QCOP. Respect expected validity ranges (Table I.
2) Perform training on a dataset subset while capturing measurements according Table VI.
3) Calculate efficiency by equations 4 5.

C. QCO Score Calculation

The Quality-Focused Score is calculated for FLOPS-based-score as QCO_F (4) and QCO_P for Page-Fault-based score (5).

V. Evaluation

The usability, plausibility and balance of the proposed metric is assessed in a comprehensive evaluation.

A. Experiments

In Experiment 2, binary classification tasks were performed by different vectorization and classifier technologies. Two datasets are selected for Experiment 2, both with moderate text length; SMS Spam Classification (25.000 samples)[32] and Movie Survey Classification (7.805 samples) [33]. The experiments were run on host 1 (Table IV) in two virtual hosts with different virtualisation technologies. The results of experiment 2 are shown in Table VII. To compare the QCOF and QCOP metrics in Experiment 2, two set of QCO had to be created as some FLOPS measurement were not available (QCO_1 & QCO_{2*}).

In Experiment 3, the metric was further evaluated by applying it to an optimisation problem similar to Use Case 4. The objective was hyper-parameter optimisation with efficiency as the cost function. The ML process involved fine-tuning a transformer model (DistilBERT [29]), word embedding and text classification. The experiment aimed to find the most efficient value for the Maximum Sequence Length (MSL) for the SMS spam detection task [32], which was run on Host 5 (IV).

B. Usability

Experiment 2 shows surprising results that can be explained by runtime conditions such as schedulers, competing processes and caching techniques. The experiment is not designed to make general statements about specific combinations of vectorization or classification methods. Consequently, the following statements apply only to this experiment, which does not preclude testing the usefulness of the efficiency metric. The word embedding method is on average superior to the TFIDF in terms of quality, but there are classifiers (NB, GD) that can compensate for the quality disadvantage and in some cases achieve the highest efficiency. This is due to the low workload. The transformer method requires significantly more work. It achieves high quality, but also takes the longest time. The Random Forest (RF) classifier has a low efficiency because it requires a lot of computation and time to achieve good quality. The Support Vector Machine (only linear kernel) classifier benefits most from the word embeddings and therefore achieves good efficiency. When comparing the combinations in terms of the time to work ratio (WO-Focus), the worst ratio (1.28) is found for IMDB/TFIDF/SVM and

 ISBN: 978-1-68558-097-1

TABLE VII
QCO EVALUATION RESULTS

DAT	VECT	CLF	DUR	$EVMetric$ QUA	$TIME$	SPA	WO_P	WO_F	QCO Metrics QCO_P	QCO_F	Rankings EXP$_1$	QCO_{1P}	EXP$_2$	QCO_{2P}	QCO_{2F}
SMS	TFIDF	NB	00:00:34	0,968	0,407	0,260	1,043	0,691	1,260	1,726	1	1	1	1	1
SMS	TFIDF	GD	00:02:36	0,972	0,583	0,371	1,043	0,631	1,190	1,678	2	2	2	2	2
IMDB	TFIDF	GD	00:00:19	0,885	0,339	0,222	0,616	0,610	1,145	1,153	3	3	3	3	3
SMS	DIST-T	DIST-T	00:01:34	0,982	0,525	0,384	1,249		1,099		6	4			
SMS	BERT	SVM	00:11:29	0,972	0,755	0,510	1,143	1,465	1,018	0,852	4	5	5	4	4
IMDB	DIST-T	DIST-T	02:38:32	0,982	1,058	0,795	1,058		0,970		5	6			
SMS	BERT	GD	02:04:00	0,978	1,030	0,696	1,140	1,637	0,956	0,752	8	7	4	5	6
IMDB	DISTIL	DISTIL	11:31:52	0,978	1,228	0,923	1,136		0,848		7	8			
IMDB	TFIDF	NB	00:01:18	0,845	0,504	0,330	0,618	0,581	0,766	0,797	9	9	6	6	5
SMS	BERT	NB	01:58:30	0,932	1,024	0,692	1,141	1,638	0,715	0,563	11	10	8	7	8
SMS	DISTIL	DISTIL	01:39:58	0,983	1,005	0,750	1,845		0,697		12	11			
SMS	BERT	RF	01:09:50	0,916	0,963	0,651	1,140	1,639	0,660	0,516	10	12	7	8	9
IMDB	TFIDF	RF	00:00:58	0,809	0,469	0,309	0,617	0,646	0,604	0,586	15	13	9	9	7
SMS	TFIDF	RF	00:03:02	0,787	0,601	0,376	1,043	0,931	0,335	0,363	16	14	12	10	10
IMDB	TFIDF	SVM	00:20:20	0,714	0,821	0,554	0,638	0,812	0,223	0,194	13	15	11	11	11
SMS	TFIDF	SVM	00:00:35	0,652	0,409	0,264	1,048	1,092	0,117	0,113	14	16	10	12	12

Cloumns: Dataset, Vectorizer, Classifier, Duration, Quality, Time, Space, WO_F= Work (FLOPS), WO_P = Work (Minor Page Faults), QCO Metrics, Rankings by Domain EXPerts, or QCO, SMS = SMS Spam Dataset[32], IDB = IMDB Dataset[33], BERT = BERT word embedding, DIST-T = finetuned DistilBERT word embedding (PyTorch) & classification, DISTIL = finetuned DistilBERT word embedding (TensorFlow + keras) & classification, GD = Gradient Descent, SVM = Support Vector Machine, NB = Naïve Bayes

the best for SMS/TFIDF/NB with 0.39. This leads to the conclusion that the measurement of time does not reflect the amount of work.

In Experiment 3, both QCO_F and QCO_P were successfully computed (see Table VIII). The most efficient MSL configuration consisted of 512 tokens, resulting in a high classification quality and moderate duration. On the other hand, the configuration with 126 tokens showed an increased workload and duration. The fastest result was obtained with an MSL of 256 tokens.

TABLE VIII
EFFIENCY OF DISTILBERT

SL	Measurements Duration	F1	Dimensions Q	W	S	T	Scores QCO$_F$	QCO$_P$
128	09:08:50	0,76	0,64	3,02	0,45	0,78	0,159	0,164
256	00:27:02	0,78	0,64	2,86	0,45	0,71	0,171	0,172
512	00:50:13	0,78	0,65	2,94	0,47	0,72	0,183	0,174

Text Classification Efficiency with DistilBERT with different maximum Sequence Length (SL). Smoothed Dimensions: Quality, Work, Space and Time. Efficiency Scores Quality Focused based on FLOPS (QCO_F) and Minor Page Faults (QCO_P)

C. Plausibility

QCO was successfully generated for all ML methods in Experiment 2. A comparative assessment of QCO based on expert rankings is used for evaluation. Domain experts ranked the dimensions, listed in Table VII Column Rank-EXP. Comparing expert and QCO rankings, a minimal deviation from the expert rank was observed for high quality ML methods, but the deviation increased with decreasing quality. This variance can be attributed to the expert's specific weighting of quality relevance, which is particularly evident in the DistilBERT setups.

D. Balance & Compensation

QCO achieved a balance of aspects through compensation (Table I). The results of Experiment 2 showed no anomalies for different datasets; even ML processes with large datasets achieved high efficiency. Moreover, significant differences in speed and computational complexity were observed for comparable efficiency, suggesting a balance in these aspects. Due to the small number of hosts available for evaluation, the balance on host setups could not be verified.

VI. DISCUSSION

This study proposes an efficiency metrics framework for machine learning techniques that addresses different aspects of cost-effectiveness, resource utilisation and model performance. The approach is intended to be adaptable and applicable to a variety of ML techniques and host setups. The objectives of the efficiency metric framework have been defined to address different real-world scenarios and use cases. The proposed efficiency metrics provides information for identifying the optimal ML technique and hyperparameters, selecting ML techniques for limited resources or private data, comparing classification performance across different host setups, and estimating computational costs. The metric framework introduces several dimensions that collectively capture the efficiency of ML techniques to achieve these goals. These dimensions include quality, work, load, space and duration, each of which contributes to the overall efficiency score. The dimensions are designed to measure different aspects of ML performance and resource utilisation, allowing for a comprehensive evaluation. One of the key advantages of the proposed framework is its adaptability to different ML techniques and tasks. The dimensions and metrics can be adjusted based on

 ISBN: 978-1-68558-097-1

specific use cases and requirements, ensuring relevance and accuracy in different contexts. This adaptability makes the metric framework suitable for a wide range of applications, from small-scale experiments to large-scale production systems.

The efficiency metrics introduced in the framework, such as QCO, FCO, ICO and ACO, provide different perspectives on efficiency. These compact metrics provide a clear, at-a-glance view of efficiency, making it easier for researchers and practitioners to evaluate and compare different ML techniques. In addition, the Efficiency Vector (EV) metric provides detailed information about the performance of ML techniques on individual dimensions, providing insights for further analysis and improvement.

The process of instantiating the efficiency metrics requires empirical investigation to ensure that the metric definitions are concrete and applicable to specific ML experiments. The validity of metric instantiation is emphasised, and the size of the experiment plays an important role in achieving reliable results. By conducting experiments on different datasets and host setups, the metric instantiation gains credibility and comparability.

Overall, the proposed efficiency metrics framework offers a promising approach for quantifying and comparing the cost-effectiveness of machine learning methods. By providing a comprehensive view of efficiency across multiple dimensions, it enables researchers and practitioners to make informed decisions regarding ML techniques, resource allocation, and model performance optimisation. The adaptability and applicability of the metrics in different contexts make them a valuable tool for advancing the field of ML and facilitating the development of efficient and effective ML models.

VII. Conclusion and Future Work

The successful calculation and evaluation of efficiency scores will pave the way for further achievements in the efficiency of machine learning research. By introducing complex dimensions that take into account measurement correlations, such as FLOPS to data volume or memory usage to duration time, we were able to potentially balance the metric and achieve a compensation of dataset size and host-setup.

When evaluating the QCO dimensions, we encountered a limitation due to insufficient samples. However, the FLOPS-based instance showed consistency and our attempt to use a small page fault measure to support the work dimension showed partial success. To gain further insight into efficiency correlations, future work could focus on dimensions that incorporate ML-specific attributes, such as model size.

It should be noted that the validation methods of the proposed metric currently rely on peer opinion. While this provides valuable insights, we recognise the importance of statistical validation to increase the credibility and robustness of the metric.

The efficiency of machine learning methods is undoubtedly influenced by expert opinion and the relevance of quality to the specific application. However, we need to be aware of the potential exponential increase in complexity if quality is used as the only guiding principle for development. Striking a balance between different efficiency dimensions is crucial to ensure a practical and rational approach to optimising machine learning processes.

Further research and statistical validation will contribute to the refinement and wider adoption of these efficiency metrics, ultimately advancing the field of ML and facilitating the development of efficient and effective machine learning models.

Acknowledgments

This Research was funded by the Federal Ministry of Education and Research of Germany in framework of FiSK (Project-Number 16DHB4005).

References

[1] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: State of the art, current trends and challenges," *Multimedia tools and applications*, vol. 82, no. 3, pp. 3713–3744, 2023.

[2] K. Raza, V. Patle, and S. Arya, "A review on green computing for eco-friendly and sustainable it," *Journal of Computational Intelligence and Electronic Systems*, vol. 1, no. 1, pp. 3–16, 2012.

[3] D. Rousseau and A. Ustyuzhanin, "Machine learning scientific competitions and datasets," in *Artificial Intelligence for High Energy Physics*, World Scientific, 2022, pp. 765–812.

[4] A. Ittoo and A. van den Bosch, "Text analytics in industry: Challenges, desiderata and trends," *Computers in Industry*, vol. 78, pp. 96–107, 2016.

[5] R. W. Numrich, "Performance metrics based on computational action," *The International Journal of High Performance Computing Applications*, vol. 18, no. 4, pp. 449–458, 2004.

[6] A. Vaswani *et al.*, "Tensor2tensor for neural machine translation," in *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, 2018, pp. 193–199.

[7] D. Ghimire, D. Kil, and S.-h. Kim, "A survey on efficient convolutional neural networks and hardware acceleration," *Electronics*, vol. 11, no. 6, p. 945, 2022.

[8] G. Tzanos, C. Kachris, and D. Soudris, "Hardware acceleration on gaussian naive bayes machine learning algorithm," in *2019 8th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, 2019, pp. 1–5.

[9] W. Fu, K. Wang, C. Zhang, and J. Tan, "A hybrid approach for measuring the vibrational trend of hydroelectric unit with enhanced multi-scale chaotic series analysis and optimized least squares support vector machine," *Transactions of the Institute of Measurement and Control*, vol. 41, no. 15, pp. 4436–4449, 2019.

[10] Z. Li, M. Paolieri, L. Golubchik, S.-H. Lin, and W. Yan, "Predicting throughput of distributed stochastic gradient descent," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 11, pp. 2900–2912, 2022.

[11] T.-J. Yang, Y.-H. Chen, J. Emer, and V. Sze, "A method to estimate the energy consumption of deep neural networks," in *2017 51st asilomar conference on signals, systems, and computers*, IEEE, 2017, pp. 1916–1920.

[12] J. Thomson, M. O'Boyle, G. Fursin, and B. Franke, "Reducing training time in a one-shot machine learning-based compiler," in *International workshop on languages and compilers for parallel computing*, Springer, 2009, pp. 399–407.

[13] R. Fischer, M. Jakobs, S. Mücke, and K. Morik, "A unified framework for assessing energy efficiency of machine learning," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, 2022, pp. 39–54.

[14] A. Kumar, S. Goyal, and M. Varma, "Resource-efficient machine learning in 2 kb ram for the internet of things," in *International conference on machine learning*, PMLR, 2017, pp. 1935–1944.

[15] J. Huang *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7310–7311.

 ISBN: 978-1-68558-097-1

[16] H. Westphal and S. Menge, "On the supervisory control of distributed high-performance computing systems in engineering," *WIT Transactions on Information and Communication Technologies*, vol. 11, 1970.
[17] D. F. Snelling, "A philosophical perspective on performance measurement," in *Computer benchmarks*, 1993, pp. 97–103.
[18] R. W. Numrich, "Computational force, mass, and energy," *International Journal of Modern Physics C*, vol. 8, no. 03, pp. 437–457, 1997.
[19] D. Schönle, C. Reich, and D. O. Abdeslam, "Linguistic driven feature selection for text classification as stop word replacement," *Journal of Advances in Information Technology*, vol. 14, no. 4, pp. 796–802, 2023. DOI: 10.12720/jait.14.4.796-802.
[20] S. P. Bayerl *et al.*, "Offline model guard: Secure and private ml on mobile devices," in *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, IEEE, 2020, pp. 460–465.
[21] M. Feurer and F. Hutter, "Hyperparameter optimization," *Automated machine learning: Methods, systems, challenges*, pp. 3–33, 2019.
[22] S. González-Carvajal and E. C. Garrido-Merchán, "Comparing bert against traditional machine learning text classification," *arXiv preprint arXiv:2005.13012*, 2020.
[23] C. Zhang, M. Yu, W. Wang, and F. Yan, "{Mark}: Exploiting cloud services for {cost-effective},{slo-aware} machine learning inference serving," in *2019 USENIX Annual Technical Conference (USENIX ATC 19)*, 2019, pp. 1049–1062.
[24] R. W. Numrich, L. Hochstein, and V. R. Basili, "A metric space for productivity measurement in software development," in *Proceedings of the second international workshop on Software engineering for high performance computing system applications*, 2005, pp. 13–16.
[25] M. Capra, B. Bussolino, A. Marchisio, G. Masera, M. Martina, and M. Shafique, "Hardware and software optimizations for accelerating deep neural networks: Survey of current trends, challenges, and the road ahead," *IEEE Access*, vol. 8, pp. 225 134–225 180, 2020.
[26] C. Kiefer, *Quality indicators for text data*, BTW 2019 – Workshopband, 2019. DOI: 10.18420/btw2019-ws-15.
[27] E. M. Dharma, F. L. Gaol, H. Warnars, and B. Soewito, "The accuracy comparison among word2vec, glove, and fasttext towards convolution neural network (cnn) text classification," *J Theor Appl Inf Technol*, vol. 100, no. 2, p. 31, 2022.
[28] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[29] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.
[30] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A matlab-like environment for machine learning," in *BigLearn, NIPS Workshop*, 2011.
[31] W. McKinney, "Data Structures for Statistical Computing in Python," in *Proceedings of the 9th Python in Science Conference*, S. van der Walt and J. Millman, Eds., 2010, pp. 56–61. DOI: 10.25080/Majora-92bf1922-00a.
[32] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the study of sms spam filtering: New collection and results," in *Proceedings of the 11th ACM symposium on Document engineering*, 2011, pp. 259–262.
[33] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, USA: Association for Computational Linguistics, 2011, pp. 142–150.

 ISBN: 978-1-68558-097-1