CO_2 Emissions of AI Applications: An Investigation on its Measurement

Pankhuri Verma and V. Dinesh Reddy* and Marco Aiello

Abstract The rapid expansion of Artificial Intelligence (AI) has led to a significant increase in the use of Data Centres (DCs), which are essential for processing and storing vast amounts of data. However, this surge in AI deployment has raised environmental concerns about increased Carbon Dioxide (CO_2) emissions. Various solutions have been proposed to address the energy efficiency of DCs such as advanced cooling systems or selecting training locations with lower cooling needs or greener power supplies. To achieve further improvements, one needs to be able to measure actual emissions at the code level so that an optimisation strategy can be designed and evaluated. To address the issue, we explore an innovative approach to precisely measure the CO_2 emissions of AI applications. By introducing a linear regression energy estimation model based on Performance Monitoring Counters (PMCs) we calculate the CO_2 emission of AI applications. PMCs such as the total number of instructions and the total number of cycles of the computer processor are considered ideal for energy estimation due to their strong correlation with the processor's energy consumption and minimal overhead on resource utilisation. For this research, only the Central Processing Unit (CPU) and Dynamic Random Access Memory (DRAM) are considered, as they consume the maximum energy compared to other parts of the processor. This approach is easily extendable to GPUs. In the presented evaluation, the energy estimation model produced an error of only 0.158% for CPU and 0.272% for DRAM.

Pankhuri Verma
Service Computing, IAAS, University of Stuttgart e-mail: st180247@stud.uni-stuttgart.de

V. Dinesh Reddy*
Service Computing, IAAS, University of Stuttgart, Germany e-mail: dinesh.vemula@iaas.uni-stuttgart.de

Marco Aiello
Service Computing, IAAS, University of Stuttgart, Germany e-mail: marco.aiello@iaas.uni-stuttgart.de

1 INTRODUCTION

The current success of the AI is calling for substantial expansion in high-performance computing clusters and DCs because of their intensive processing requirements and energy consumption. Advancements in AI models have significantly improved in domains such as machine translation, speech recognition, and object detection. However, AI models are computationally demanding due to their large datasets, extensive model sizes, and numerous parameters used for training. Developing these models also requires thorough experimentation with various hyperparameters, resulting in increased load on the DC processors and consequent CO_2 emissions.

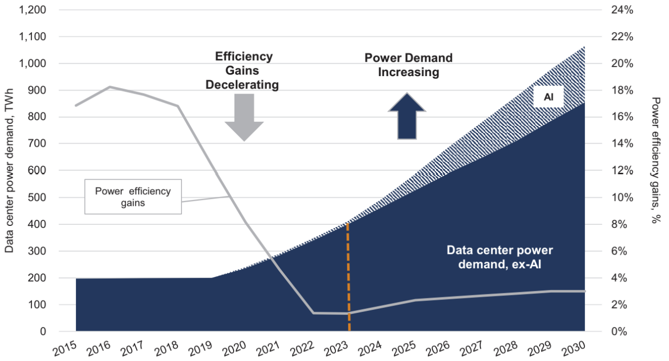


Fig. 1: Power demand of DCs over the years [1]

In the analysis presented by Goldman Sachs, the DC power demand has increased from 1%-2% in 2022 to 3%-4% in 2023 [1]. It is anticipated that the demand for power in DCs will increase by 160% between the years 2023 and 2030. Fig. 1 shows an illustration of the DC power demand in terawatt-hours, and % gains. The study also contains a prediction about the growth in power demand in DCs resulting in an increase in CO_2 emissions from DCs by more than 100% by 2030 compared to 2022. It is alarming to see the forecast presented by Goldman Sachs stating that power demand from AI will increase by approximately 200 TWh from 2024 to 2030, with AI expected to account for around 20% of the total DC power demand by 2030 [2]. The study also estimates that a ChatGPT search consumes about 6 to 10 times more power than a traditional Google search. To put things into perspective, the carbon footprint of training LLMs like Bidirectional Encoder Representations from Transformers (BERT) on Graphics Processing Unit (GPU) is comparable to the emissions from a New York to San Francisco flight [3]. As the usage of these models expands, the environmental impact intensifies, contributing to global climate change. Therefore, it is imperative to develop and implement strategies to reduce these emissions.

Historically, developers have focused on energy-efficient scheduling and resource allocation to improve energy utilisation of DCs rather than employing optimisation strategies at the code level [4]. They have primarily emphasised the software qualities such as performance and accuracy of AI applications without considering energy efficiency. This has often led to highly energy-intensive applications. Therefore, it is essential to be aware of energy consumption and to optimise the code. As a step in this direction, the goal and contributions of the present paper are:

- Identifying correlations between PMCs and AI energy consumption.
- Development of a linear regression energy estimation model for AI applications based on PMCs.
- CO_2 estimation of AI applications execution.

The rest of the paper is structured as follows. In Section 2, we overview related work about software energy usage. Section 3 contains a discussion about the methodology for using PMCs to determine the energy consumption of AI applications. Section 4 illustrates the methodology regarding CO_2 emission estimation. The paper closes with Section 5 summarising the findings of the presented research and discussing directions of future work.

2 RELATED WORK

The pioneering work by Tiwari et al. was the first of its kind to use an instruction-based power analysis of software operations [5]. The methodology was groundbreaking because it shifted the focus from traditional hardware-centric power analysis to a software-oriented perspective. These models provided insights into how different instructions impact overall power usage. The research by Bellosa introduced PMCs as an effective indicator of power usage [6]. García-Martín et al. provide an overview of the methods for estimating energy usage of Machine Learning (ML) applications [7]. The research showcased the most recent energy estimation software tools, various methods for estimating energy usage, and energy estimation models. In another paper by García-Martín et al. , the energy estimation techniques were categorised based on the different ML scenarios, like processing big datasets, training and inference stages. They discussed the various types of analytical and empirical methods used to measure energy consumption. Simulation and PMC were also discussed as the best practices for measuring energy consumption [8].

Contreras and Martonosi proposed a power estimation model for the Intel PXA255 processor that used PMCs to estimate CPU and DRAM power consumption [9]. It linked PMCs such as instructions executed, data dependencies, instruction cache misses, and Translation Lookaside Buffer (TLB) misses with an error rate of 4%. This study attempted to estimate the energy consumption of software by running various traditional benchmarks to generate energy models. However, AI applications differ from traditional benchmark programs in terms of computational demand and energy usage, due to the large dataset size, various model parameters and weights.

Building upon this foundational research, our study focuses specifically on AI applications. Therefore, specific AI models have been executed to gather the dataset for our energy model creation. Unlike other approaches, our work is the first of its kind to use AI models to estimate energy consumption during model training. This approach allows a more precise and tailored understanding of the energy requirements unique to AI workloads.

3 PMC-BASED ENERGY ESTIMATION MODEL

We use PMCs to measure the energy consumption of AI applications. PMCs are dedicated hardware registers used for tracking executed instructions, cache hits and misses in modern CPUs that monitor various performance-related events. They are used to collect valuable information regarding software and hardware performance attributes. The present research aims to find the correlation between hardware PMCs and the energy consumption of AI applications. Our experiments have been performed on Intel® CoreTM i7-8565U CPU running at 1.80 GHz (142, 0x8e). For our analysis, we focus primarily on the training process of AI applications, as it is the most resource-intensive part of model development [10]. Energy consumption of processor components such as the CPU and DRAM is only taken into consideration as they have the most immediate influence on the AI training process. We do not consider components like Solid State Drives (SSD) and Hard Disk Drive (HDD) as they do not directly influence the running AI processes [9].

3.1 Dataset Generation

Our methodology utilises the process of running various AI benchmark programs to generate a dataset comprising PMC data and corresponding CPU and DRAM energy measurements. The ML models used in this study to generate our dataset are Linear Regression, Logistic Regression, K-Nearest Neighbour (KNN), Support Vector Machines (SVM), Decision Trees and Neural Networks. The dataset comprises values of various PMCs such as total instructions, total cycles, cache hits and misses and energy consumption of CPU and DRAM. We employ these models to obtain the energy and performance of only the model training stage. This facility is not provided by traditional ML benchmarks currently. Each model was executed with a range of dataset sizes, hyperparameters, and features to represent the various operational profiles of ML applications and a dataset comprising approximately 2.5k data points was gathered.

While these models were executed on the Intel processor, we used the widely-known Performance Application Programming Interface (PAPI) interface and the Running Average Power Limit (RAPL) interface to measure the PMC data and energy consumption of CPU and DRAM, respectively.

PAPI: The PAPI interface is a portable way to access hardware PMCs. It tracks over 100 predefined events through high-level and low-level interfaces [11]. Our research utilises the PAPI high-level events called PAPI_TOT_INS (total instructions) and PAPI_TOT_CYC (total CPU cycles) that measure the impact of system modifications on performance and have low overhead due to parallel processing [8].

RAPL: The RAPL interface is a feature found in modern Intel processors that monitors power usage of the computing unit such as CPU socket package, DRAM, and GPU [12]. It has been designed to measure the energy consumption of specific code snippets, making it ideal for fine-grained analysis. The primary advantage of RAPL is that it measures energy consumption without interfering with already running computational processes [13].

These Application Programming Interfaces (APIs) were called using methods in the Python package called pyRAPL [14, 15] and pyPAPI to simultaneously measure the energy consumption and PMCs, respectively.

3.2 Selection of PMCs

The Intel CoreTM i7-8565U CPU running at 1.80 GHz (142, 0x8e) processor provides access to 59 different PMCs via the PAPI interface. However, during our research, we discovered that not all PMCs show a strong correlation with the energy consumption of CPU and DRAM energy. Therefore, a correlation between energy consumption and the PMC data was examined in the dataset using Spearman's Rank Correlation Coefficient (ρ). A coefficient of +1 indicates a strong positive correlation, while -1 indicates a strong negative correlation. Based on the ρ value, we chose only those PMCs that show a strong correlation with energy consumption to avoid redundancy. It was observed that the total CPU cycles and CPU energy have a ρ of 0.922 and total instructions and CPU energy have a ρ of 0.861. Similarly, the total CPU cycles and DRAM energy have a ρ of 0.869 and total instructions and DRAM energy have a ρ of 0.751. Other PMC events exhibited poor correlation with CPU and DRAM energy, having ρ values of 0.55 or less. Therefore, we have used only total instructions and total CPU cycles for our energy model creation.

Total Instructions: Total instructions are the individual operations carried out by a CPU according to the program.

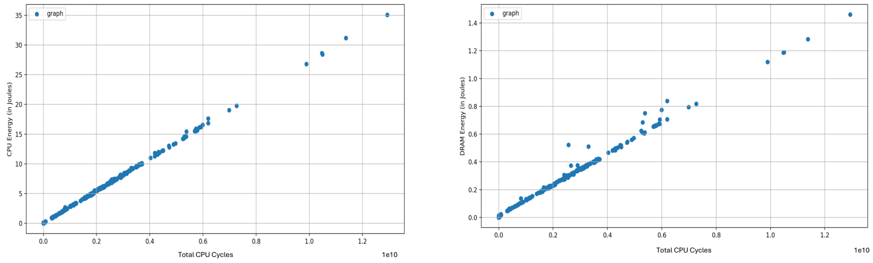
Total CPU Cycles: The number of clock cycles that the CPU uses to complete tasks is measured by CPU cycles.

Table 1 shows the dataset comprising total instructions, total CPU cycles, CPU energy, and DRAM energy of 10 data instances.

Supporting the above correlation values, Fig. 2 depicts a perfect linear relationship of CPU cycles with CPU energy and DRAM energy. Similarly, Fig. 3 illustrates a linear relationship of total instructions with CPU energy and DRAM energy. This linear trend aids our argument for the creation of a linear energy model based on total instructions and total CPU cycles.

Index	Total Instructions	Total CPU Cycles	CPU Energy	DRAM Energy
1	5744069	5457638	0.1201597	0.0149658
2	5704238	5362627	0.0356077	0.0042114
3	5723654	5562773	0.0313781	0.0048767
4	5751021	5371274	0.026184	0.0041076
5	5775481	6344354	0.0481506	0.004895
6	5811632	6053471	0.0405944	0.004718
7	5833511	5218797	0.0323302	0.0042725
8	5858551	5349801	0.0264892	0.0043945
9	5877574	5374618	0.0291869	0.0042053
10	5908991	5284704	0.0176758	0.0038942

Table 1: 10 data instances of the dataset consisting of total instructions, total CPU cycles, CPU energy and DRAM energy used to train the linear regression energy estimation model.



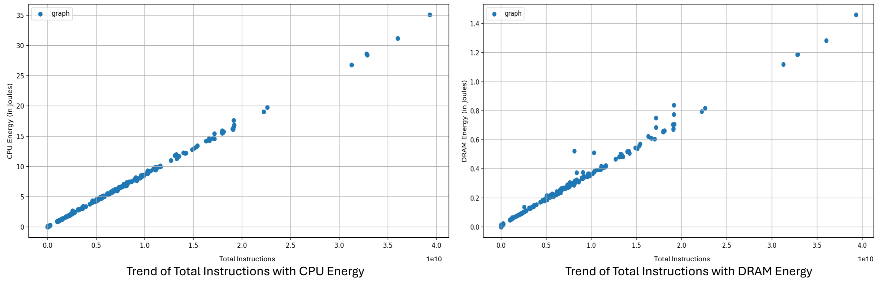
Trend of Total CPU Cycles with CPU Energy

Trend of Total CPU Cycles with DRAM Energy

Fig. 2: Linear correlation between Total CPU Cycles and CPU Energy and DRAM Energy

3.3 Linear Regression Energy Estimation Model

We create a linear regression model to estimate the energy usage of AI applications operating on Intel processors. This process involves using the generated dataset to create our linear energy model and determining the model weights. The dataset is passed through the data pre-processing stage, where the extreme outliers are removed and the dataset is normalised using Min-Max Scaling [16] to bring all the features on a uniform, unit scale [0,1]. Consequently, the dataset is divided into training (55%), validation (25%), and testing (20%) sets. A linear regression algorithm is employed to generate the energy models using total instructions and total



Trend of Total Instructions with CPU Energy

Trend of Total Instructions with DRAM Energy

Fig. 3: Linear correlation between Total Instructions and CPU Energy and DRAM Energy

CPU cycles as the independent variables and CPU energy and DRAM energy as the dependent variables. The model is fine-tuned using the validation dataset to achieve the highest accuracy and test data is utilised to test the model's accuracy.

The following equations represent the linear model used to estimate the CPU and DRAM energy consumption in micro-joules (μJ):

$$CPU\ energy = b_{1,0} + (b_{1,1} \times tot_{\text{ins}}) + (b_{1,2} \times tot_{\text{cyc}}\ ) \tag{1}$$

$$DRAM\ energy = b_{2,0} + (b_{2,1} \times tot_{\text{ins}}) + (b_{2,2} \times tot_{\text{cyc}}) \tag{2}$$

where:

- tot_{ins} denotes the total number of instructions executed.
- tot_{cyc} denotes the total number of CPU cycles executed.
- $b_{1,0}$, $b_{1,1}$, and $b_{1,2}$ represent the regression weights of the CPU model.
- $b_{2,0}$, $b_{2,1}$, and $b_{2,2}$ represent the regression weights of the DRAM model.

Table 2 shows the values of CPU and DRAM model weights that will be used to precisely estimate the energy consumption of the CPU and DRAM when an AI model is running.

Using Equations 1 and 2 we can estimate the precise value of energy consumption during the training phase of any AI application based on the total number of instructions and total number of CPU cycles.

Weight	Value
$b_{1,0}$	0.00042871
$b_{1,1}$	0.84030965
$b_{1,2}$	0.16071597
$b_{2,0}$	0.00153388
$b_{2,1}$	0.84543874
$b_{2,2}$	0.18049153

Table 2: Values of CPU and DRAM regression model weights.

3.4 Analysis of Energy Models

The results of our energy models are shown in Table 3. The energy models showcase an error of only 0.158% and 0.273% for CPU and DRAM, respectively. The achieved level of accuracy is notably superior compared to previous research in this domain [9]. Both energy models perform very well with a CPU Mean Absolute Error (MAE) of 0.00060 and DRAM MAE of 0.00255. The R Squared (R^2) scores of 0.9998 and 0.9925 for CPU and DRAM, respectively, also support the argument that these models are a good fit for measuring the energy consumption of AI applications.

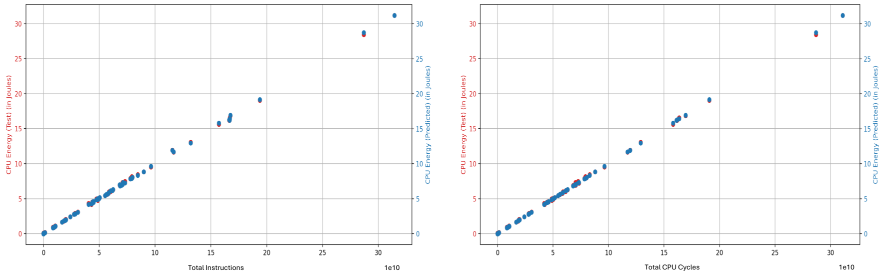
Energy Model	MAE	R^2 Score	Error%
CPU	0.00060	0.9998	0.158%
DRAM	0.00255	0.9925	0.273%

Table 3: Metrics for performance evaluation of CPU and DRAM energy models.

The graphs in Fig. 4 show the CPU energy model's performance based on the total number of instructions and total CPU cycles. The blue data points represent the predicted CPU energy while the red data points represent the test CPU energy. A close match between the predicted and test data points validates our model accuracy with only 0.158% error rate. Similarly, the graphs in Fig. 5 show the DRAM energy model's performance based on the total number of instructions and the total CPU cycles. The predicted (blue) and test (red) DRAM energy data points coincide, with an error rate of only 0.273%.

4 CO_2 EMISSION ESTIMATION OF AI APPLICATIONS

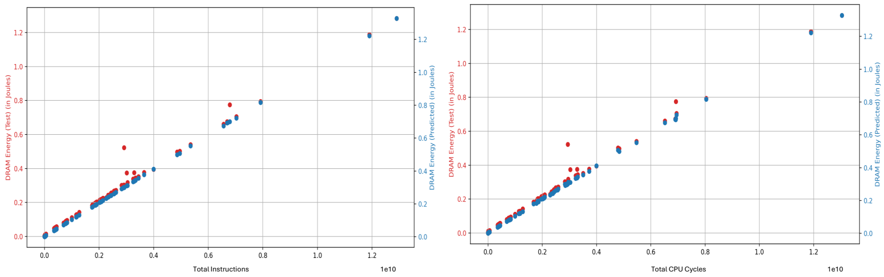
The next step is the assessment of CO_2 emissions resulting from the AI applications. There is a non-strict relationship between CO_2 emissions and energy usage due to various factors such as geographic location, and energy mix [17]. We use the



Test and Predicted CPU Energy for different number of Instructions

Test and Predicted CPU energy for different number of CPU Cycles

Fig. 4: Performance of CPU Energy Model - Predicted vs Test energy values



Test and Predicted DRAM energy for different number of Instructions

Test and Predicted DRAM energy for different number of CPU Cycles

Fig. 5: Performance of DRAM Energy Model - Predicted vs Test energy values

carbon intensity in our analysis to accurately represent the geographical variations in emissions.

Carbon Intensity is a coefficient showing the weight of CO_2 emissions, expressed in kilogram (kg), for each Kilowatt-hour (KWh) of electricity produced. The carbon intensity is determined by the energy mix of a region that includes fossil fuels and renewable energy sources like solar power, biomass and more [17].

The carbon intensity of a geographical region can be used to measure the precise CO_2 emissions. Equation 3 captures the way to precisely calculate the CO_2 emission of a computing unit by multiplying the total CPU and DRAM energy consumption by, for instance, Germany's carbon intensity.

$$\textit{Total } CO_2 \textit{ Emission} = \sum (Energy_i) \times \textit{Carbon Intensity} \tag{3}$$

where:

- i represents the computing unit (CPU and DRAM)
- $Energy_i$ denotes the energy consumed by the computing unit (in KWh).
- *Carbon Intensity* denotes the carbon intensity of the corresponding region.

Since our experiments are conducted on an Intel machine running at our university in Germany, the carbon intensity of Germany was used to measure the precise value of CO_2 emissions from any AI application. As of April 29, 2024, the carbon intensity coefficient for the German region is 385.389, according to Codecarbon [18].

To illustrate our methodology of calculating CO_2 emission, we have run a simple linear regression model on the sklearn California housing dataset. The total instructions executed while running the model is 1884248.8 and the total CPU cycles is 2219907.

We use our generated formula for CPU and DRAM energy to calculate the total energy consumption and CO_2 emission of the model.

$$\begin{aligned} \textit{CPU energy} &= b_{1,0} + (b_{1,1} \times tot_{ins}) + (b_{1,2} \times tot_{cyc}) \\ &= 0.00042871 + (0.84030965 \times 1884248.8) \\ &\quad + (0.16071597 \times 2219907) \\ &= 1939719.34\ \mu J \\ &= 1.93971934\ J \end{aligned} \tag{4}$$

$$\begin{aligned} \textit{DRAM energy} &= b_{2,0} + (b_{2,1} \times tot_{ins}) + (b_{2,2} \times tot_{cyc}) \\ &= 0.00153388 + (0.84543874 \times 1884248.8) \\ &\quad + (0.18049153 \times 2219907) \\ &= 1993897.47\ \mu J \\ &= 1.99389747\ J \end{aligned} \tag{5}$$

$$\begin{aligned} \textit{Total } CO_2 \textit{ Emission} &= \sum (Energy_i) \times \textit{Carbon Intensity} \\ &= ((1.93971934 + 1.99389747)/3600000) \times 385.389 \\ &= 0.000421103\ kg \end{aligned} \tag{6}$$

Based on Equation 4, we calculate the energy consumption of CPU, which is 1.93971934 J. The DRAM energy was calculated as 1.99389747 J (Equation 5). Further, the total energy was used to calculate the CO_2 emission. Energy in Joules is converted to KWh for CO_2 emission calculation. According to Equation 6, 0.000421103 kg of CO_2 was emitted to train a linear regression model on the California housing dataset.

Our streamlined approach simplifies CO_2 emission estimation, making it valuable for AI development and deployment. This PMC-based energy model sets a new standard in AI energy management for the development of sustainable AI applications. This detailed level of analysis facilitates the reduction of energy consumption, leading to more efficient and eco-friendly AI systems.

5 CONCLUSIONS

Energy consumption is an important factor to consider when developing ML algorithms. The majority of research focuses on improving the accuracy of algorithms while neglecting their energy requirements. With the present research, we aim to give a new perspective on the AI industry by providing insights into the energy consumption pattern of ML models. Our research contributes to the scientific understanding of AI energy consumption and offers practical methodologies that can help developers precisely measure the energy consumption of AI applications based on PMCs.

From our research results, one can conclude that processor-specific PMCs and AI applications' energy consumption are directly correlated. A framework for estimating the energy consumption and, consequently, the CO_2 emissions of different AI models was provided by the use of a linear regression energy model. By applying these regression models, developers can predict and manage the energy consumption of AI operations more effectively, leading to significant energy savings and reducing the environmental impact associated with running intensive AI tasks in the real world.

The research done so far has established an initial understanding of energy consumption and CO_2 emissions in AI applications on CPUs. However, future work will focus on GPUs and TPUs as they have better processing capabilities and are commonly used in the field of AI. Further study will be done on GPU and TPU-based PMCs to estimate the energy consumption of AI applications.

References

1. Brian Singer, Derek R. Bingham, Brendan Corbett, Carly Davenport, and Alberto Gandol. Ai/data centers' global power surge and the sustainability impact. https://www.goldmansachs.com/intelligence/pages/gs-research/

ai-data-centers-global-power-surge-and-sustainability-impact/report.pdf, 2024. Accessed: 2024-05-21.

2. Carly Davenport, Brian Singer, Neil Mehta, Brian Lee, and John Mackay. Ai, data centers and the coming us power demand surge. *https://www.goldmansachs.com/intelligence/pages/gs-research/generational-growth-ai-data-centers-and-the-coming-us-power-surge/report.pdf*, 2024. Accessed: 2024-05-21.
3. Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. 2019.
4. V. Dinesh Reddy, G. R. Gangadharan, G. S. V. R. K. Rao, and Marco Aiello. Energy-efficient resource allocation in data centers using a hybrid evolutionary algorithm. pages 71–92, 2020.
5. V. Tiwari, S. Malik, A. Wolfe, and M.T.-C. Lee. Instruction level power analysis and optimization of software. *10.1109/ICVD.1996.489624*, pages 326–328, 1996.
6. Frank Bellosa. The benefits of event-driven energy accounting in power-sensitive systems. 2000.
7. Eva García-Martín, Crefeda Faviola Rodrigues, Graham Riley, and Håkan Grahn. Estimation of energy consumption in machine learning. *Journal of Parallel and Distributed Computing*, 134:75–88, 2019.
8. Eva García-Martín, Niklas Lavesson, Håkan Grahn, Emiliano Casalicchio, and Veselka Boeva. How to measure energy consumption in machine learning algorithms. In *ECML PKDD 2018 Workshops*, pages 243–255, Cham, 2019. Springer International Publishing.
9. Gilberto Contreras and Margaret Martonosi. Power prediction for intel xscale® processors using performance monitoring unit events. In *Proceedings of the 2005 International Symposium on Low Power Electronics and Design*, ISLPED '05, page 221–226, New York, NY, USA, 2005. Association for Computing Machinery.
10. Kawsar Haghshenas, Brian Setz, and Marco Aiello. Co2 emission aware scheduling for deep neural network training workloads. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 1542–1549, 2022.
11. Phil Mucci, Shirley Moore, Christine Deane, and George Ho. Papi: A portable interface to hardware performance counters. 1999.
12. Abdelhafid Mazouz, David C. Wong, David Kuck, and William Jalby. An incremental methodology for energy measurement and modeling. In *Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering*, ICPE '17, page 15–26, New York, NY, USA, 2017. Association for Computing Machinery.
13. University of Maine System. Running average power limit energy reporting. *https://www.intel.com/content/www/us/en/developer/articles/technical/software-security-guidance/advisory-guidance/running-average-power-limit-energy-reporting.html*, 2022. Accessed: 2024-05-22.
14. University of Lille. Welcome to pyrapl's documentation! *https://pyrapl.readthedocs.io/en/latest/*, 2019. Accessed: 2024-05-22.
15. University of Lille. pyrapl version 0.2.3.1 project description page. *https://pypi.org/project/pyRAPL/*, 2018. Accessed: 2024-05-22.
16. Lucas B.V. de Amorim, George D.C. Cavalcanti, and Rafael M.O. Cruz. The choice of scaling technique matters for classification performance. *Applied Soft Computing*, 133:109924, January 2023.
17. Kadan Lottick, Silvia Susai, Sorelle A. Friedler, and Jonathan P. Wilson. Energy usage reports: Environmental awareness as part of algorithmic accountability. 2019.
18. CodeCarbon. Germany carbon intensity. *https://github.com/mlco2/codecarbon/blob/master/codecarbon/data/private_infra/global_energy_mix.json*, 2020. Accessed: 2024-04-28.