How can we estimate the energy consumption of training an AI model?

Presentation and comparison of existing tools

Mathilde JAY - MIAI meetings - February 2022
PhD candidate - mathilde.jay@univ-grenoble-alpes.fr - MIAI Edge Intelligence - INRIA DataMove & Avalon
PhD supervisors: Denis Trystram, Laurent Lefevre

Methodology

Estimation

From hardware characteristics
(Thermal design power)

Software power models

From hardware
performance counters

Energy consumption

External measures

Power mesures from
outside the hardware

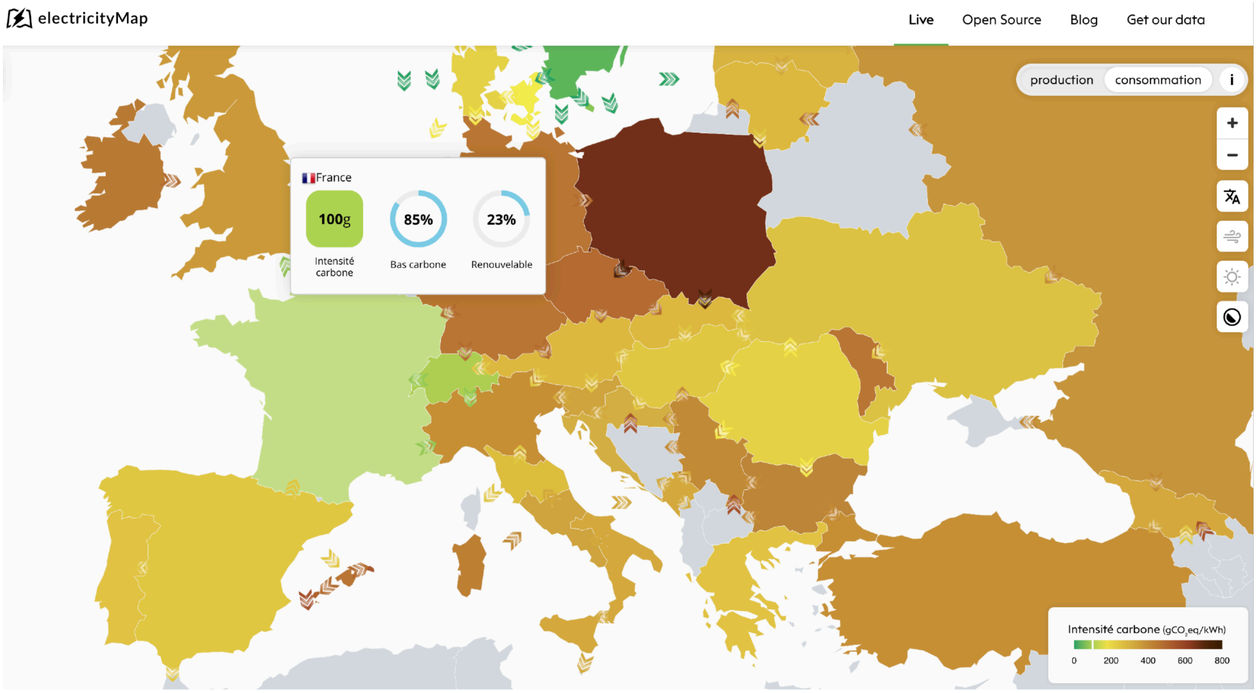
Taking into account external factors

$$Energy_{training} = Energy \times \text{External Factors}$$

- With External Factors:
  - PUE: Power Usage Effectiveness (To take into account everything that is necessary for data centers to run, like cooling)
  - PSF: Pragmatic Scaling Factor (To take into account hyper parameter search)

From energy to carbon emissions

- **Energy** kWh
- **Carbon emissions** gCO2 eq
- **Carbon intensity** gCO2eq/kWh



$$Emissions_{carbon} = Energy_{training} \times Intensity_{carbon}$$

Use case for comparison

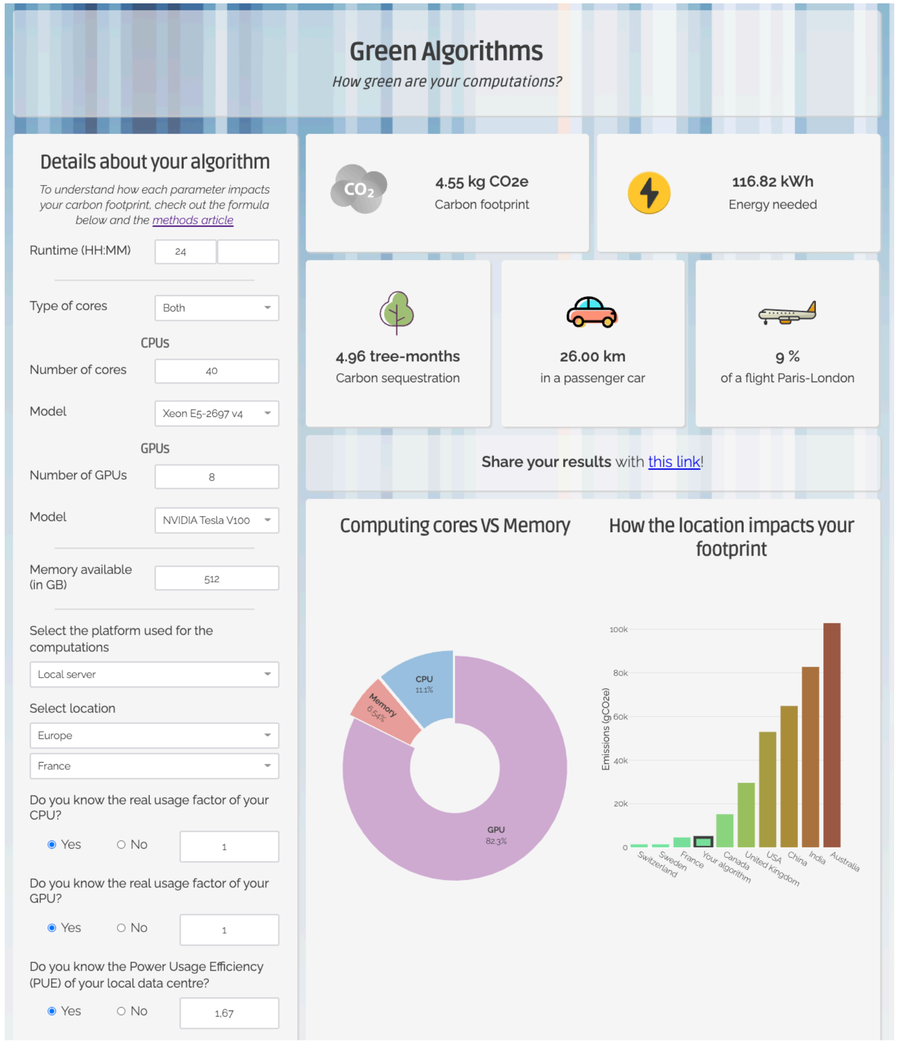
- **Application**
  - Handwritten digit recognition: Image classification
  - Model CNN: 2 convolutional layers, 2 fully connected layers (1 199 882 parameters, 5Mb)
  - Dataset: MNIST
  - Trained during 5 epochs (~ 120 seconds)
  - Library: Pytorch
- **Hardware**: gemini cluster (Grid'5000)
  - CPU: Intel Xeon E5-2698 v4 (Broadwell, 2.20GHz, 2 CPUs/node, 20 cores/CPU)
  - GPU: 8 x Nvidia Tesla V100-SXM2-32GB (32 GiB)
- **External factors**
  - Expected PUE: 1.58
  - Expected Carbon Intensity: 50 gCO2/kWh

Comparison of tools

- **Hardware estimations**
  - Green algorithms
  - ML CO2 Impact
- **External mesures**
  - Wattmeters
- **Based on software power models**
  - CodeCarbon
  - Experiment-impact-tracker
  - CarbonTracker
  - Energy Scope

Hardware estimation: Online tools

Green Algorithms (https://green-algorithms.org)



$$Energy = runtime \times ((power_{CPUs} + power_{GPUs}) \times usage + power_{memory})$$

$$Energy_{training} = Energy \times PUE \times PSF$$

$$Emissions_{Carbon} = Energy_{training} \times Intensity_{Carbon}$$

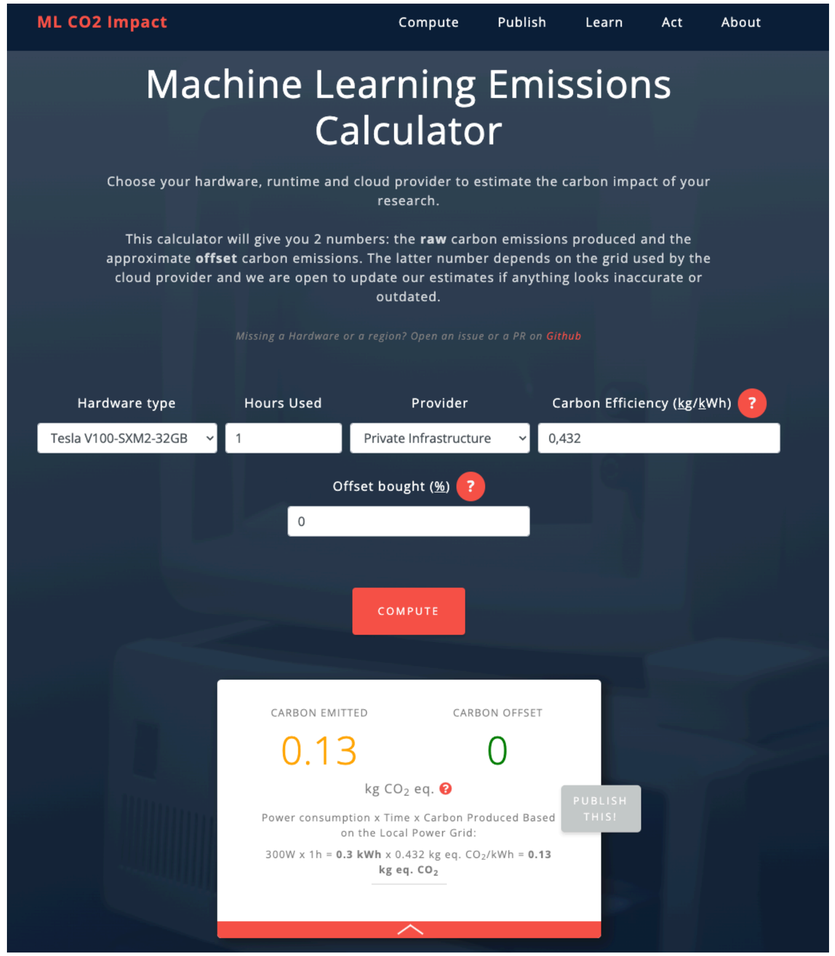
- No install
- Independent of the training

- Need to know the hardware

Hardware estimation: Online tools

ML CO2 Impact

https://mlco2.github.io/impact/?#compute



$$Energy_{training} = runtime \times power_{hardware}$$

$$Emissions_{carbon} = Energy_{training} \times Intensity_{Carbon}$$

+

- No install
- Can be done before the training (or after)

−

- Limited (can only specify one hardware at the time and not memory)
- Global carbon intensity
- Can not specify PUE for private infrastructures

Green algorithm vs ML CO2 impact

Different maximal usage power ?

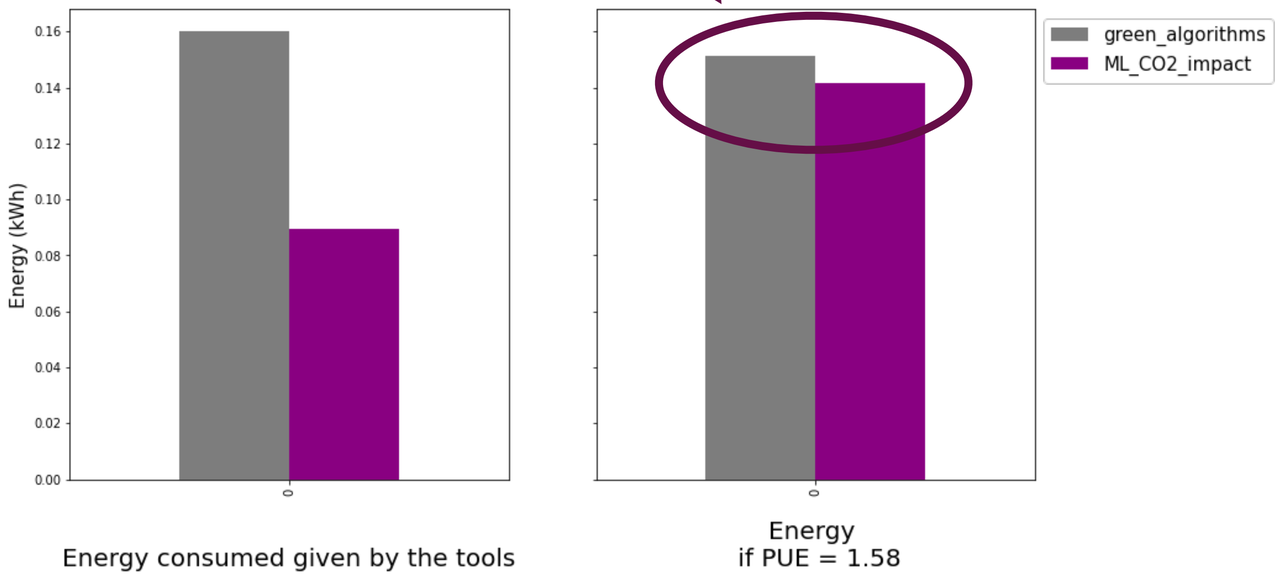
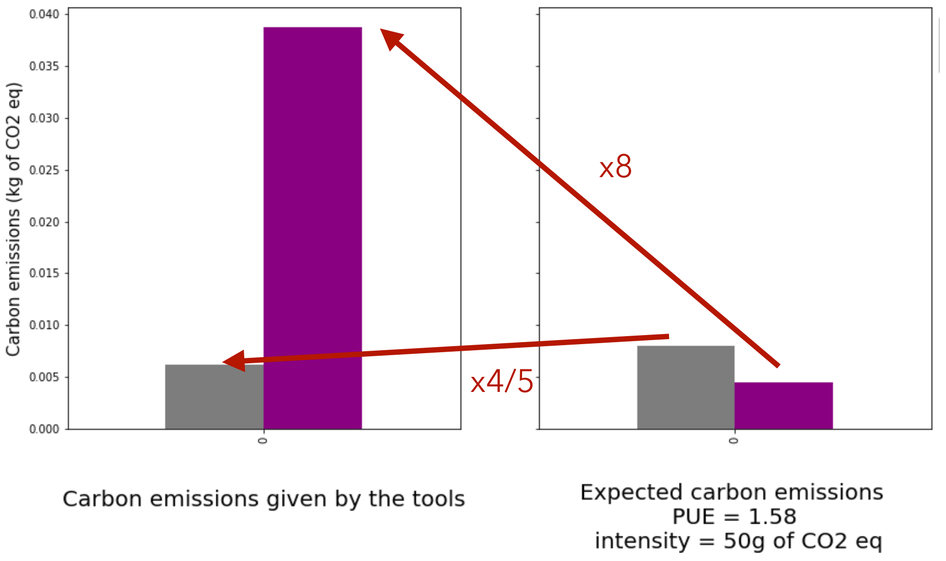


Figure 1: Energy: Given versus Expected

Impact of carbon intensity source



	GA	MCI
PUE	1.67	1
Carbon intensity (gCO2/kWh)	39	432
Source	ElectricityMap 2020	
Components	GPUs, CPUs, Memory	GPUs, CPUs

Table 1: Parameter values of GA and ML CO2 impact

External measurement

Wattmeters available on Grid5000

- **Wattmeters** on the power supply
  - One measurement every 20 milli-seconds
  - Precision at 0.1W
- Others (not studied here)
  - **PDU** (power distribution unit)
  - Intelligent Platform Management Interface (IPMI) & **BMC** controllers
    - One measurement every five seconds
    - Precision at 1W

+

- Exact value
- Independent of hardware

−

- Hard to install (except on Grid5000)
- At the plug level
- Expensive

Software power models

RAPL interface

- Power model from performance counters
- Model-Specific Registers (MSR) containing
  - Temperature, power, frequency, …
  - Per processes and components
- Only Intel Processor (Sandy Bridge and later)

Nvidia SMI

- Equivalent for Nvidia GPU

+ • Small granularity

A lot of technologies are based on those models

- Open-source python packages
  - CodeCarbon
  - Experiment-impact-tracker
  - CarbonTracker
- Profiling software
  - Energy Scope (Hervé Mathieu - Inria Bordeaux)

− • Only on specific hardware

Python packages

- Quick install
- Good documentation
- Compute carbon emissions

- Fixed energy parameters
- Poor transparency
- Need privileged access

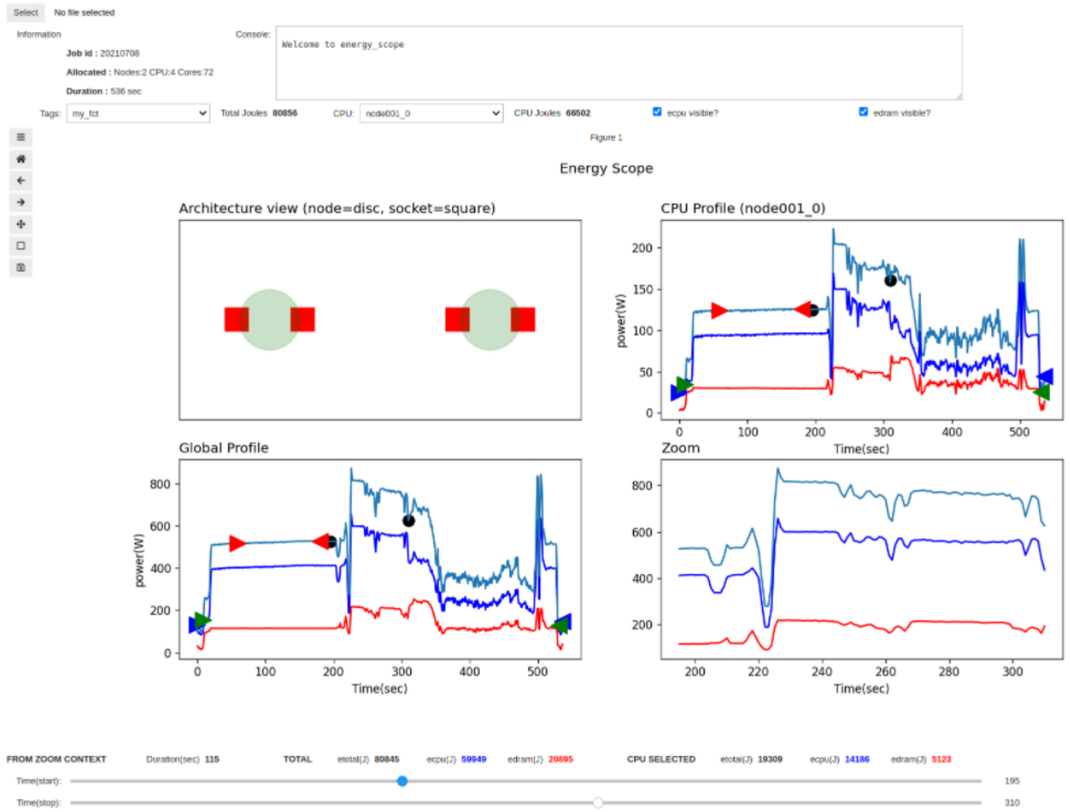
```
import impactlib
# Loading dataset and processing it
tracker = impactlib.init()
tracker.start()
# Training
tracker.stop()
# Exporting results:
tracker.energy_consumption
tracker.carbon_emissions
```

$$Energy = \sum interval \times (power_{CPUs} + power_{GPUs} + power_{memory})$$

$$Energy_{training} = Energy \times PUE$$

$$Emissions_{carbon} = Energy_{training} \times Intensity_{carbon}$$

Profiling software
Energy Scope



+

- Good visualisation
- One data point every second
- Granularity per component

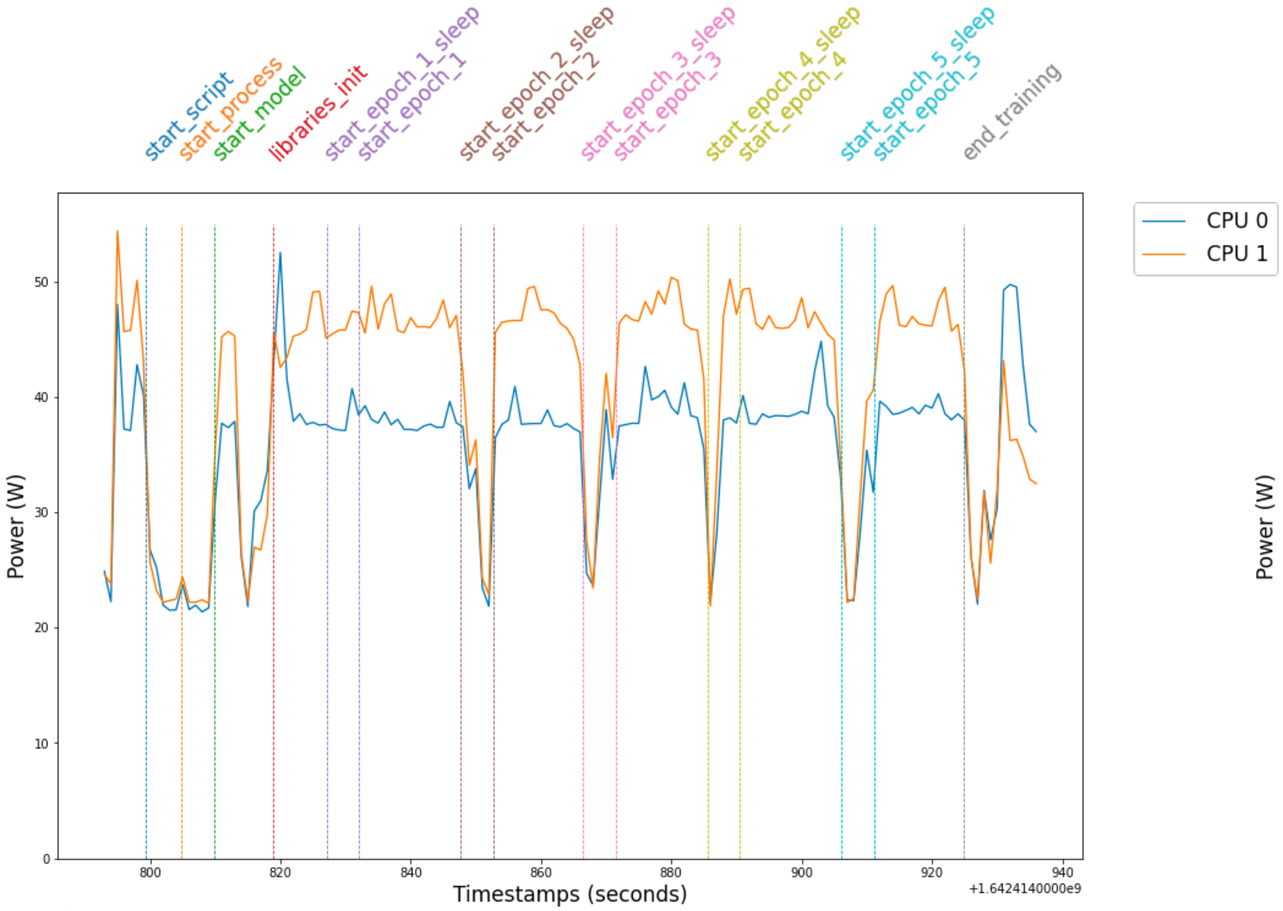
−

- Does not compute carbon emissions
- Still in development

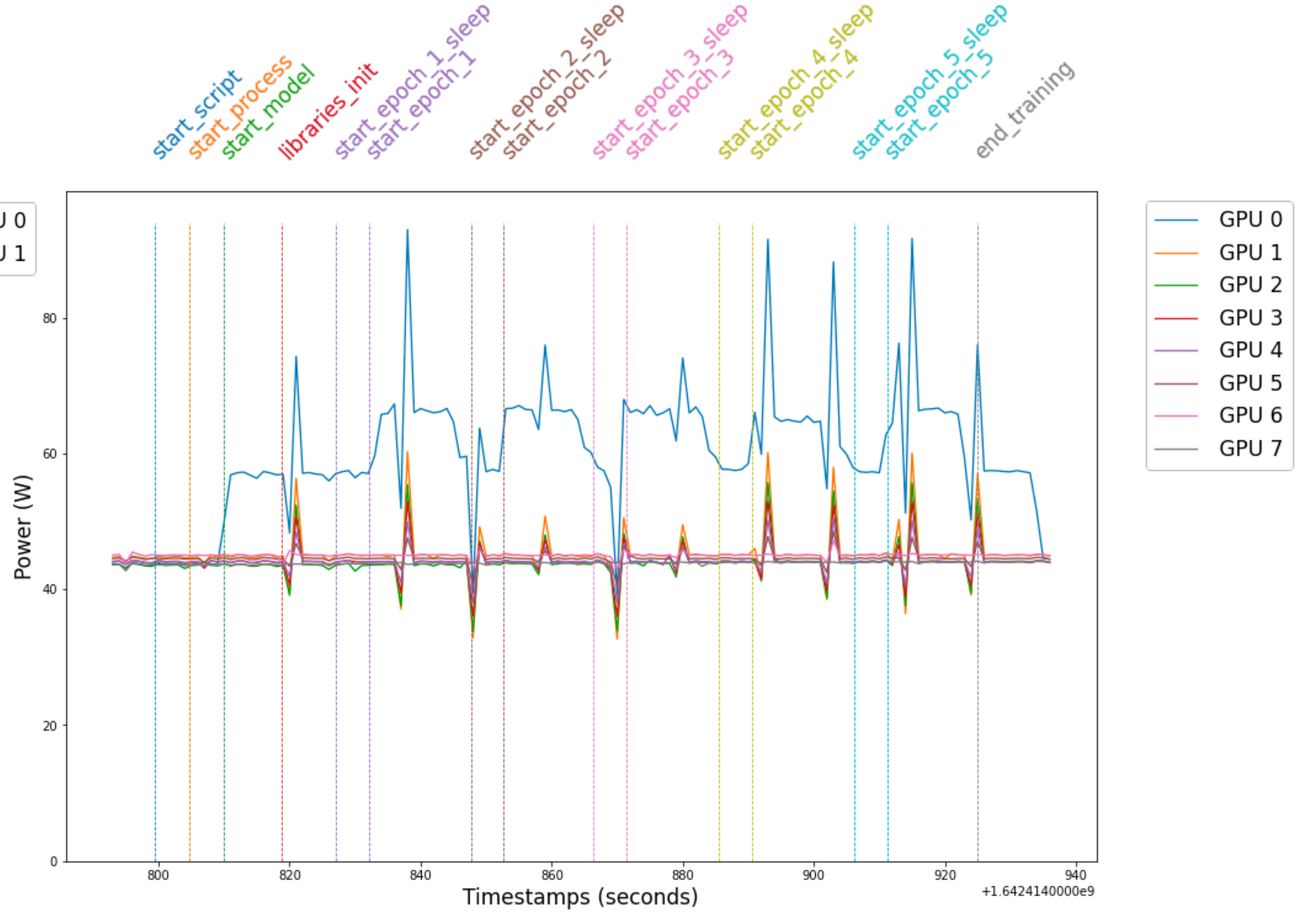
$$Energy_{training} = \sum interval \times (power_{CPUs} + power_{GPUs} + power_{memory})$$

Real-time monitoring

Energy Scope granularity



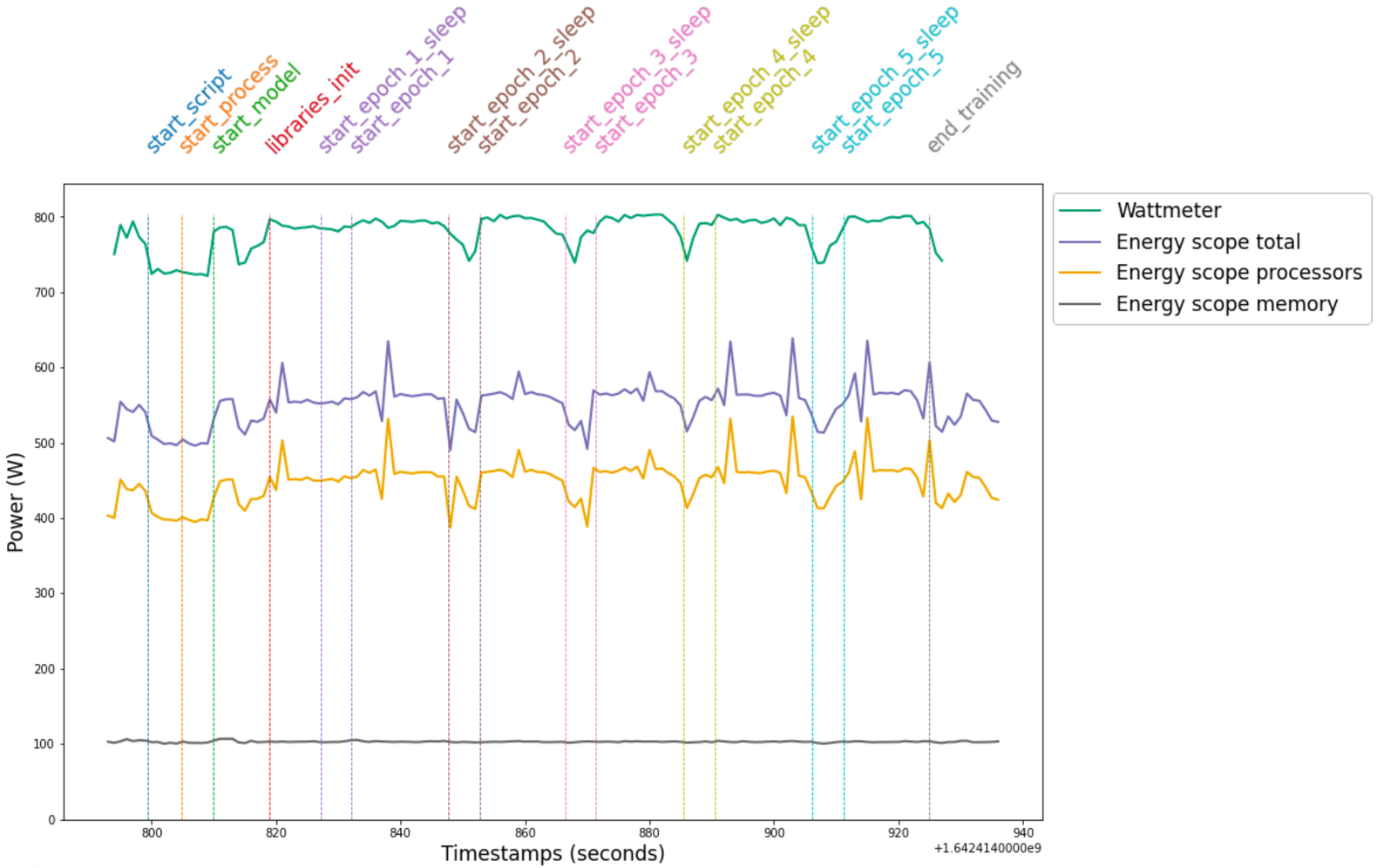
Evolution of the power of both CPU during an experiment estimated by energy scope.



Evolution of the power of all GPUs during an experiment estimated by energy scope.

Comparison: Real-time monitoring

Wattmeters vs Energy Scope



Evolution of the power during an experiment from wattmeters and energy scope.

Comparison

Focus on energy

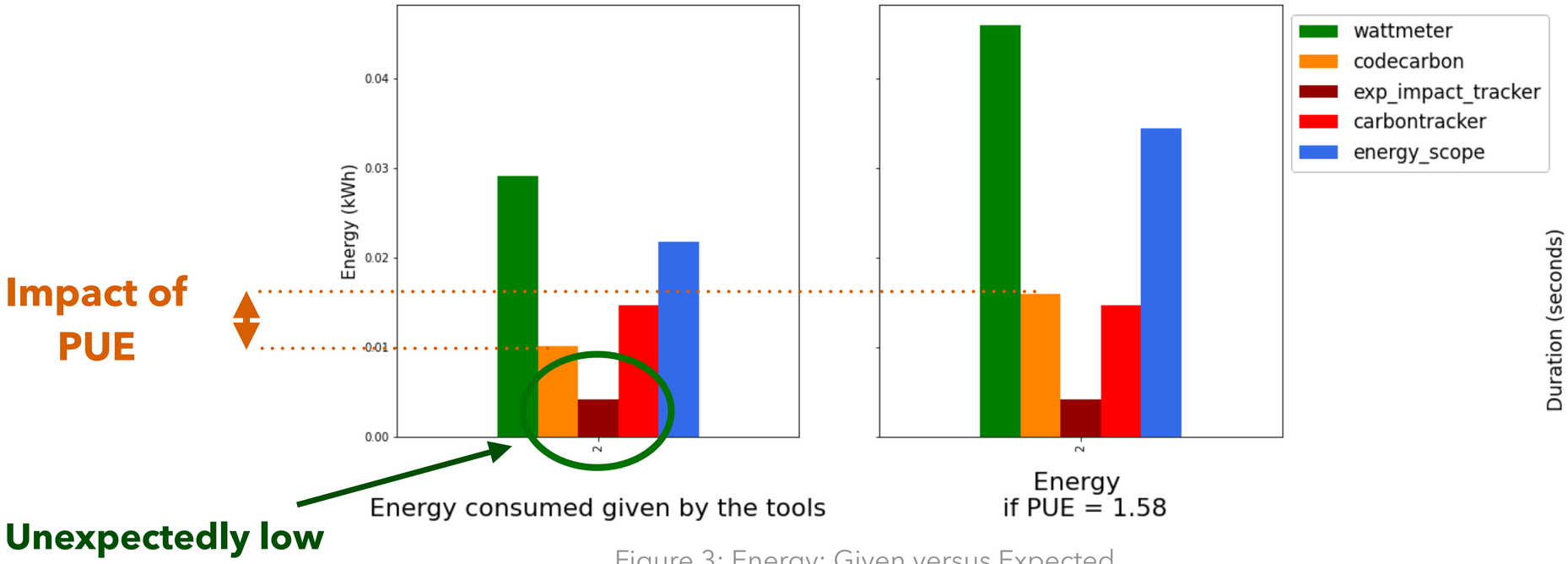


Figure 3: Energy: Given versus Expected

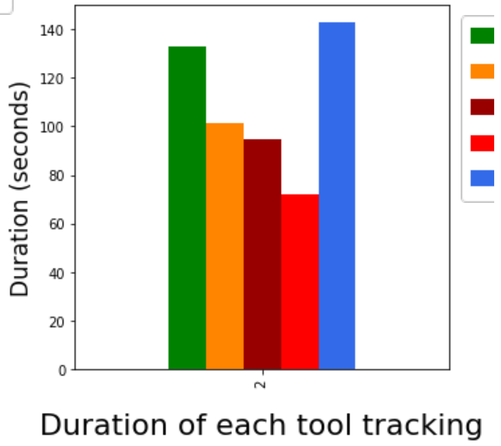


Figure 4: Tracking duration

	W	CC	EIT	CT	ES
PUE	X	X	1.58	1.58	X
Components	Whole machine	GPUs, CPUs, Memory	GPUs, CPUs, Memory	GPUs, CPUs, Memory	GPUs, CPUs, Memory

Table 2: Energy parameter values

Comparison

Focus on carbon emissions

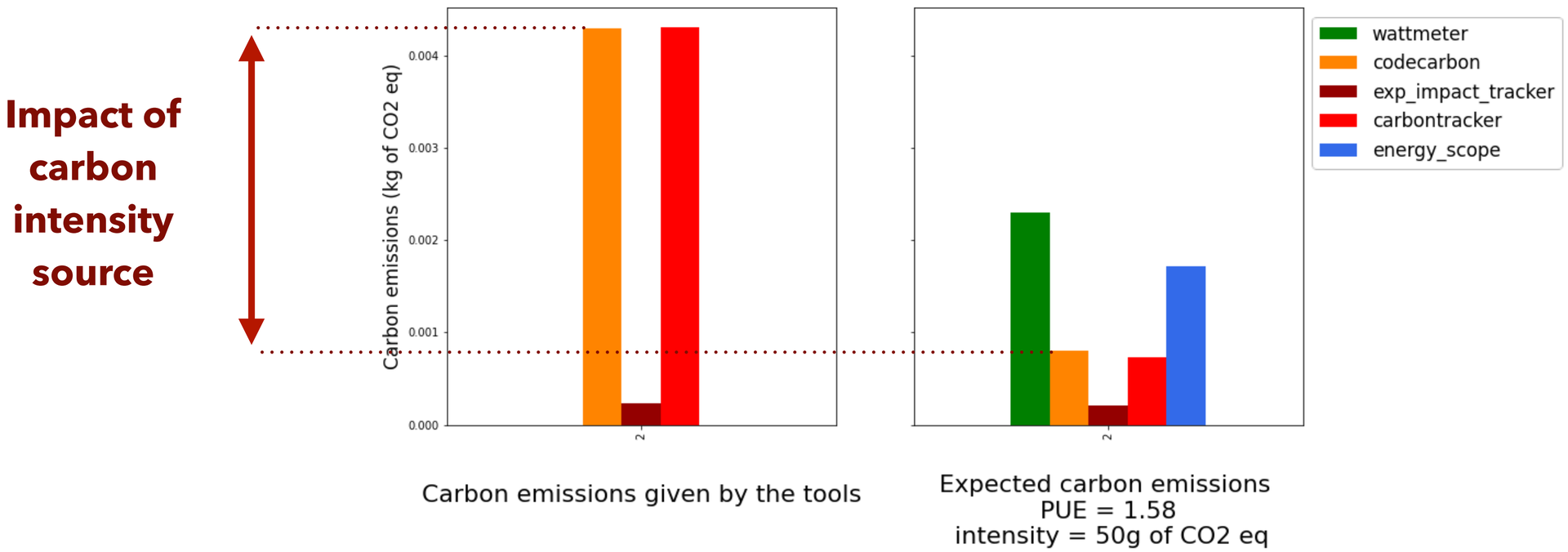


Figure 5: Carbon emissions: Given versus Expected

	W	CC	EIT	CT	ES
Carbon intensity (gCO2/kWh)	0	242.6	47.60	294.21	0
Carbon intensity source	0	?	2018 electricityMap	EU-28 in 2017	0

Table 3: Carbon emission parameter values

Comparison of all selected tools

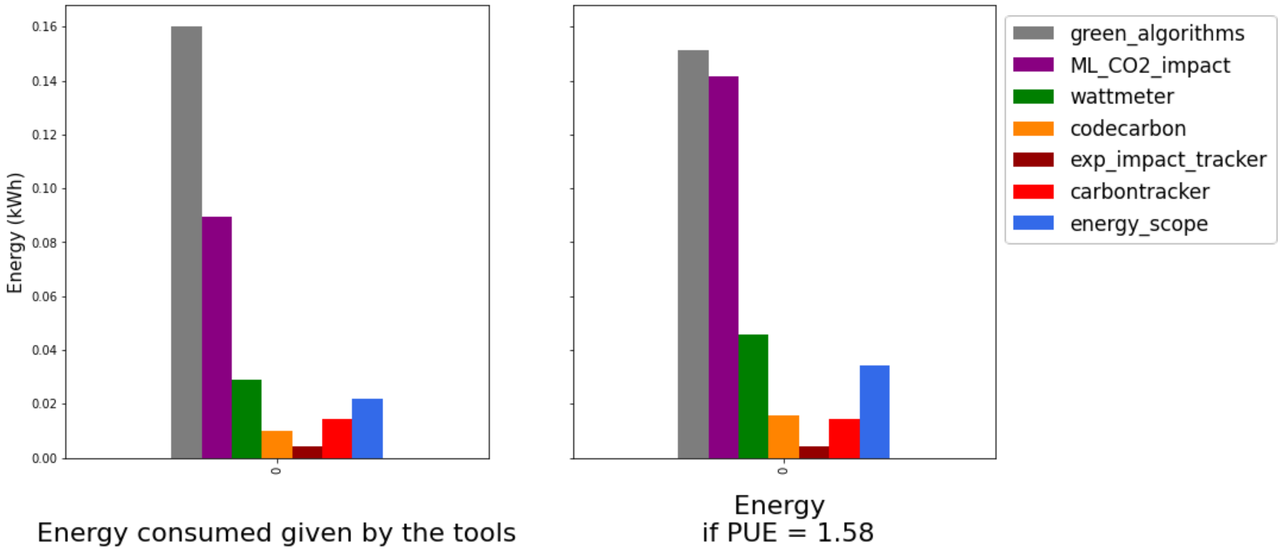


Figure 6: Energy: Given versus Expected

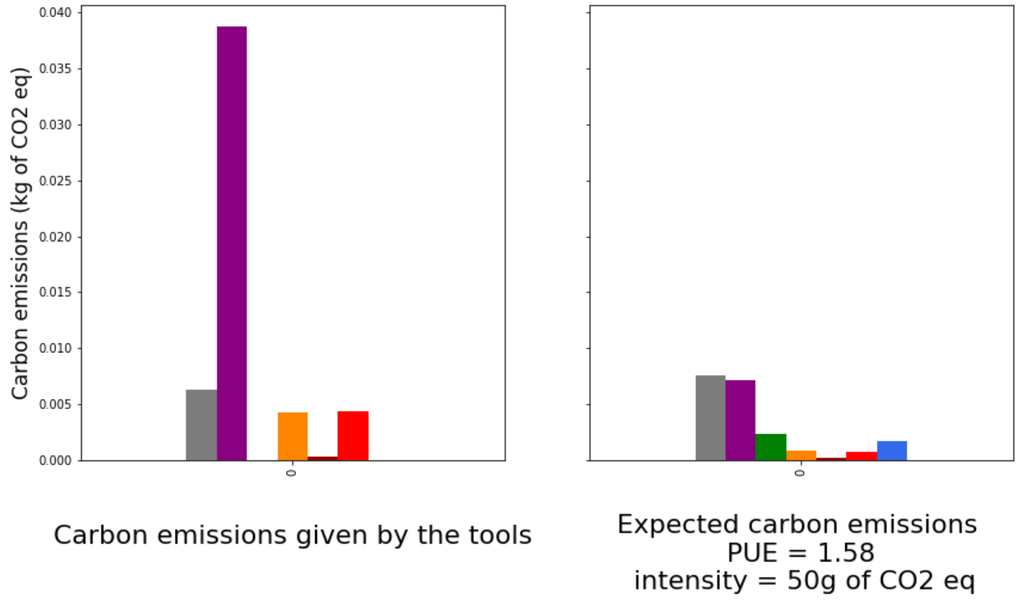


Figure 7: Carbon Emissions: Given versus Expected

	GA	MCI	W	CC	EIT	CT	ES
Carbon intensity (gCO2/kWh)	39	432		242.6	47.60	294.21	
Carbon intensity source	ElectricityMap 2020	?		?	2018 electricityMap	EU-28 in 2017	
PUE	1.67	X	X	X	1.58	1.58	X

Table 4: Energy and carbon emission parameter values

Evaluation

- If I had run this during 24h
  - Energy = 116,82 kWh

You could boil more than 1000L of water with this energy.

- Carbon footprint = 4 550 gCO2 eq

Driving 26 km by car emits the same carbon emissions.

Conclusion

- Various tools to compute energy consumption of training an AI model

- Provide it when publishing your model!

- Limits
  - Life Cycle Assessment
  - Cost of production and transport
  - Inference cost and the usage of the model

How to reduce this impact

- Be careful of which Data Center (PUE) you use, and of the location of it (carbon intensity) [GA][CC][EIT][CT]
- Algorithm optimization (random search, memory) [GA][CC][CT]
- Report energy metric and make energy efficient configurations more accessible [CC][EIT]
- Run your algorithms when carbon intensity is low [CC][CT]
- Look for models with high efficiency, not only high accuracy
- Adapt your algorithm and the settings to the hardware [CC][CT]
- Use pretrained models [CC]
- Offset emissions [GA]

References

- Ecoinfo: https://ecoinfo.cnrs.fr/
- Survey on tools: Bannour Nesrine, Sahar Ghannay, Aurélie Névéol, et Anne-Laure Ligozat. « Evaluating the carbon footprint of NLP methods: a survey and analysis of existing tools ». In *Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing*, 11-21. Virtual: Association for Computational Linguistics, 2021. https://aclanthology.org/2021.sustainlp-1.2.
- GREEN ALGORITHMS
  - https://green-algorithms.org//?runTime_hour=24&runTime_min=0&locationContinent=Europe&locationCountry=France&locationRegion=FR&PUEradio=Yes&PUE=1.67&coreType=Both&numberCPUs=40&CPUmodel=Xeon%20E5-2697%20v4&usageCPUradio=Yes&usageCPU=1&numberGPUs=8&GPUmodel=NVIDIA%20Tesla%20V100&usageGPUradio=Yes&usageGPU=1&memory=512&platformType=localServer&PSFradio=Yes&PSF=1
  - Lannelongue, Loïc, Jason Grealey, and Michael Inouye. "Green Algorithms: Quantifying the Carbon Footprint of Computation." *Advanced Science* 8, no. 12 (2021): 2100707. https://doi.org/10.1002/advs.202100707.
- ML CO2 IMPACT: https://mlco2.github.io/impact/?#compute
- CODECARBON:
  - https://github.com/mlco2/codecarbon
  - medium article: https://medium.com/bcggamma/ai-computing-emits-co%E2%82%82-we-started-measuring-how-much-807dec8c35e3

References

- CARBONTRACKER:
  - https://github.com/lfwa/carbontracker,
  - Anthony, Lasse F. Wolff, Benjamin Kanding, et Raghavendra Selvan. « Carbontracker: Tracking and Predicting the Carbon Footprint of Training Deep Learning Models ». *ArXiv:2007.03051 [Cs, Eess, Stat]*, 6 juillet 2020. http://arxiv.org/abs/2007.03051.
- EXPERIMENT IMPACT TRACKER
  - https://github.com/Breakend/experiment-impact-tracker
  - Henderson, Peter, Jieru Hu, Joshua Romoff, Emma Brunskill, Dan Jurafsky, and Joelle Pineau. "Towards the Systematic Reporting of the Energy and Carbon Footprints of Machine Learning." *ArXiv:2002.05651 [Cs]*, January 31, 2020. http://arxiv.org/abs/2002.05651.
- ENERGY SCOPE:
  - https://sed-bso.gitlabpages.inria.fr/datacenter/energy_scope.html (to request code)
  - http://energy-scope.bordeaux.inria.fr/ (INRIA server)
- Sources for carbon intensity:
  - https://www.carbonfootprint.com/docs/2018_8_electricity_factors_august_2018_-_online_sources.pdf
  - https://www.epa.gov/egrid/egrid-summary-tables
  - https://electricitymap.org/

Criteria		Green algorithms	ML CO2 Impact	Wattmeters	CodeCarbon	experiment-impact-tracker	CarbonTracker	Energy Scope
Developement	**Year**	2021	2019		2020	2020	2020	2021
	By who	UK researchers	Canadian researchers (MILA)		Canadian researchers (MILA)	PhD student from Standford University	PhD student from University of Copenhagen	Herve Mathieu, INRIA Bordeaux
Technical	**Availability of source code**	Yes	Yes		Yes	Yes	Yes	No
	Online availability	Yes	Yes	No	No	No	No	No
	Easiness of installation and use	No install needed	No install needed	Poor	Quite good	Quite good	Good	Fair
	Quality of documentation	Fair	Fair	Good	Fair	Fair	Good	Fair
	Output formats	Graphs and comparisons	Texts and Latex code	JSON	CSV file - Visualisation with Comet	JSON file - Code interface	Logs - Code interface	Online server - Jupyter Notebook - JSON files
Configuration	**Provide carbon emissions**	Yes	Yes	No	Yes	Yes	Yes	No
	Carbon intensity source	Local, from electricityMap (2020 value)	Local, listed, including electricitymap		Local, global default value (didn't find source)	Local, from electricityMap (2018 value)	Various sources, European default value (EU-28 2017)	
	Carbon intensity value for my experiments (gCO2eq/kWh)	39	432		424.6	47.6	294.21	
	Possible manual configuration of carbon intensity	No	Yes		No	Yes but not easy	No	
	External factors taken into account (PUE, PSF)	PUE, PSF	Only for cloud providers		None	PUE	PUE	
	Possible manual configuration of external factors	Yes, default value being PUE = 1.67 (2019), PSF = 1	No		No	Yes, default value PUE = 1.58	No, default value PUE = 1.67	
	Type of equipement	PC, server, Cloud	PC, server, Cloud		PC, server, Cloud	PC, server, Cloud	PC, server, Cloud	
	Other	convertion to other metrics	convertion to other metrics		Interval between measures is configurable, convertion to other metrics, if tracking tools are unavailable will use TDPs from detected hardware, if hardware unknown will provide a default value		Interval between measures is configurable, predictions after one epoch, convertion to other metrics	
Energy model	**Hardware taken into account**	CPUs, GPUs, Memory	CPUs, GPUs (less hardwares than GA)	Whole machine	CPUs, GPUs, Memory	CPUs, GPUs, Memory	CPUs, GPUs, Memory	CPUs, GPUs, Memory
	Model used	maximal usage power	maximal usage power	Measure	RAPL files - NVIDIA SMI	RAPL files - NVIDIA SMI	RAPL files - NVIDIA SMI	RAPL files - NVIDIA SMI
	Estimation or Measure	Estimation from maximal usage power	Estimation from maximal usage power	Measure	Estimation from performance counters	Estimation from performance counters	Estimation from performance counters	Estimation from performance counters
	Real time	No	No	Yes	Yes	No	Per epoch	Yes

	Build-In	CLI	CSV export	Software	Middleware	Online	Carbon emissions	Compatibility	Extension
RAPL files	✓	✓			✓			Intel CPU	
Nvidia-smi	✓	✓						Nvidia GPU	
sudometrics	✓	✓						MacOS	
Power Gadget		✓	✓	✓	✓			Intel CPU	Containers
PowDroid		✓	✓					Android	
PowerJoular		✓	✓	✓				Intel CPU	
Greenspector				✓				Intel CPU	
Scaphandre		✓	✓					Intel CPU	Containers
Power API		✓						Intel CPU	Containers
Energy scope				✓				Intel CPU, Nvidia GPU	
CodeCarbon			✓		Python		✓	Intel CPU, Nvidia GPU	Cloud
CarbonTracker			✓		Python		✓	Intel CPU, Nvidia GPU	
Experiment-impact-tracker					Python		✓	Intel CPU, Nvidia GPU	
JoularJX					Java			Intel CPU	
Carbonanlyser						✓	✓	Firefox	
Green algorithms						✓	✓	Most CPU/GPU	Cloud
ML CO2 impact						✓	✓	Most CPU/GPU	Cloud
Wattmetres		✓						✓	