ML impact

Big data management

Assignment 2

1. Create prediction pipelines; each pipeline includes:
    - preprocessing data
    - a prediction model
2. Experiment with these models
    - try out different parameters and preprocessing methods
    - use MLflow tracking during experimentation to log parameters and metrics
3. Choose the best model, log it
    - serve the model on your local machine to test if it works as intended
4. Deploy it
    - deploy it to a cloud-based machine

MLflow deployment with VMs on Azure

Push a locally developed MLflow project to Github.

Create a VM on Azure and ssh into it.

Install Python, pip, libssl-dev, pyenv. Reload shell.

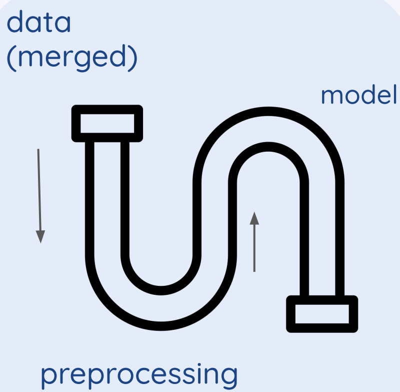
Create a virtual environment, where you install MLflow.

Clone the GitHub repository containing your model.

Run and then serve the model.

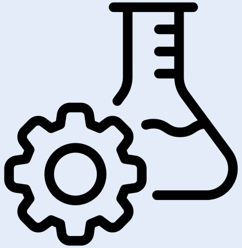
Query the served model from your machine using the VM's public IP address.

Assignment 2



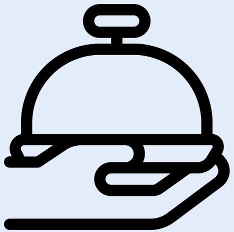
Create pipelines

`sklearn`

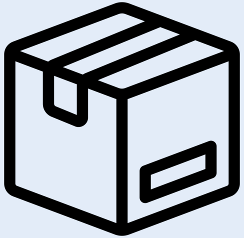


Log metrics

`MLflow tracking`



Serve model
(locally)



Package, deploy

`MLflow Projects`

`Azure`

MLflow overview

MLflow provides a unified platform to navigate model development, deployment, and management.

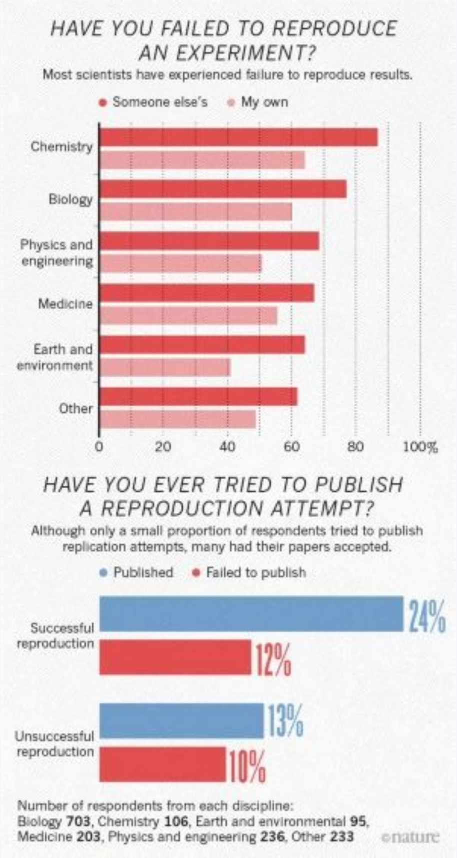
Experiment Management

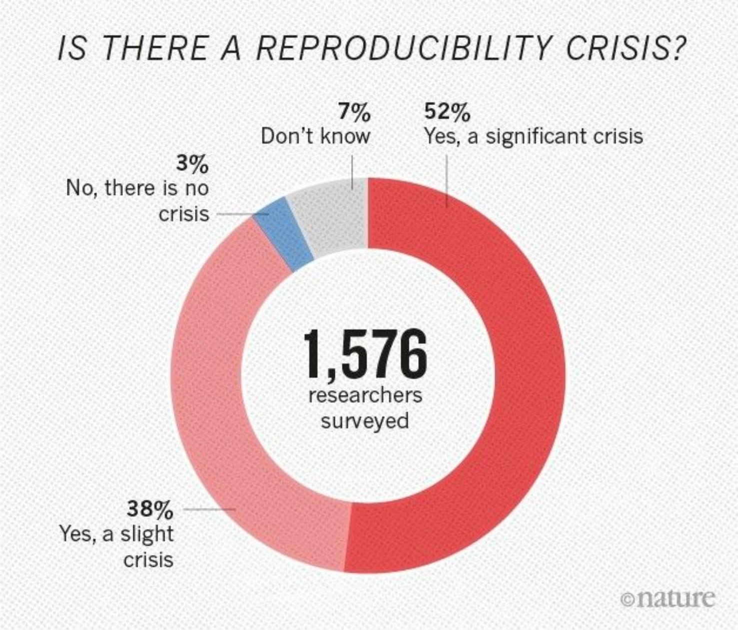
Model Management

Library Agnosticism

Reproducibility

Reproducibility crisis in science





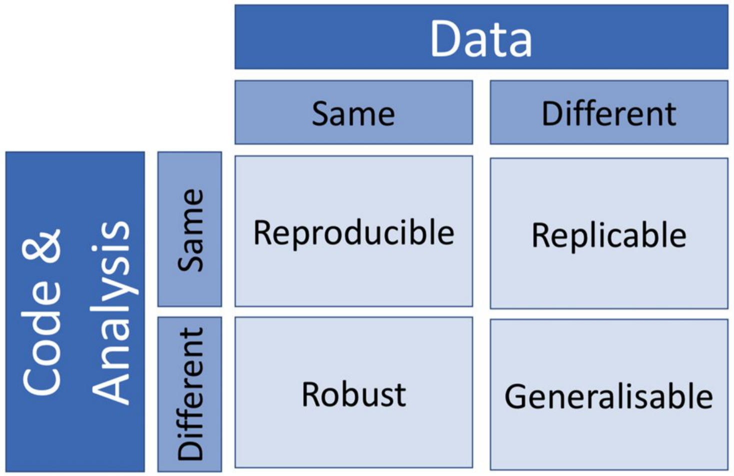
[Link to article]

Reproducibility gap in ML research

- Underspecification of the **metrics** used to report results
- Improper use of **statistics** to analyze results
- **Selective** reporting of results
- Lack of access to the same **training data** / differences in data distribution
- Lack of availability of the **code** necessary to run the experiments, or errors in the code

Reproducibility gap in ML research

- Underspecification of the **metrics** used to report results
- Improper use of **statistics** to analyze results
- **Selective** reporting of results
- Lack of access to the same **training data** / differences in data distribution
- Lack of availability of the **code** necessary to run the experiments, or errors in the code



Training the same model,

with the same hyperparameters

can give different results.

CHALLENGES IN REPRODUCIBLE AI/ML

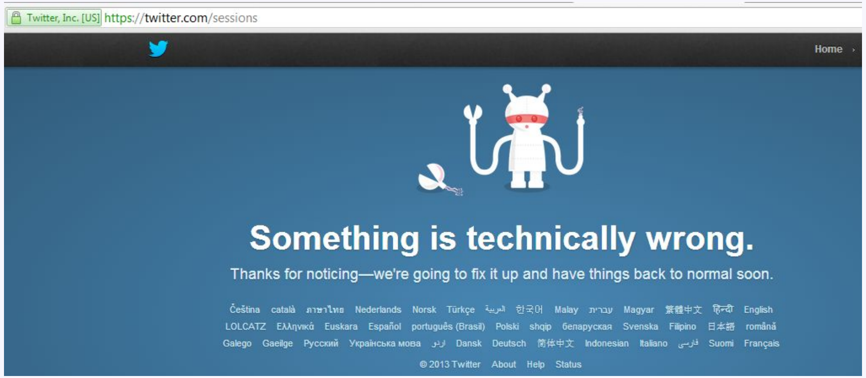
- Hardware
  - Variations from GPU architecture
  - CPU multithreading variations (SESS)
- Software
  - Unintended behaviours due to dependencies
  - Floating Point Precision
- Algorithmic
  - Dropouts
  - Random initializations
  - Shuffles
  - Random Augmentations
- Practices/Process
  - Randomizations
  - Iterative Exploratory
- Data
  - Incomplete Demographics
  - Poisoned data

Paper

Article

ML systems fail silently

Normal software fails



ML systems fails

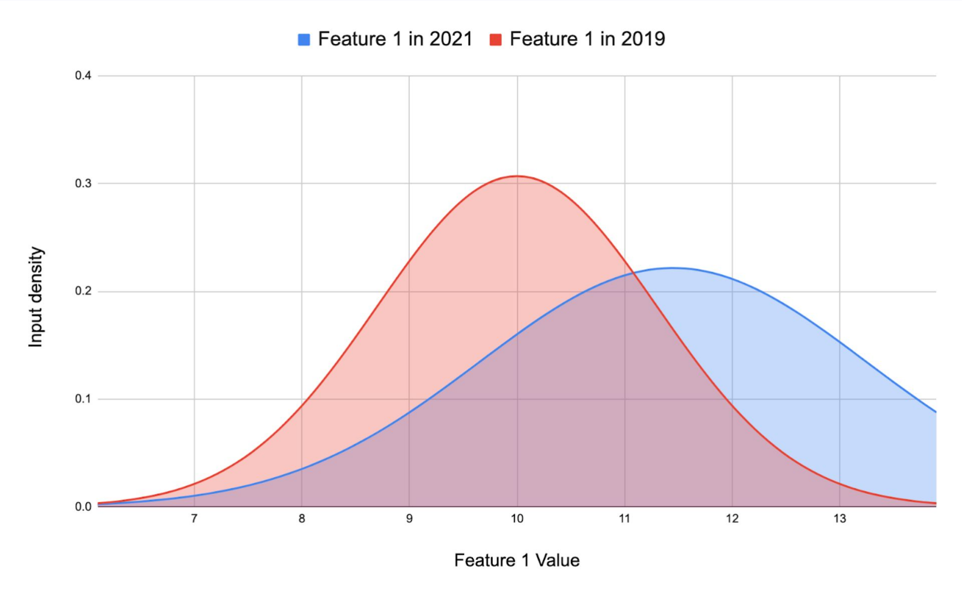
Monitoring ML systems

Monitoring = the act of tracking, measuring, and logging different metrics that can help us determine when something goes wrong.

One source: changes in data distribution

Data distribution shift

= the data a model works with changes over time



Data distribution shift

X = inputs to a model

Y = outputs

In supervised learning, the training data can be viewed as a set of samples from the joint distribution $P(X, Y)$. We model $P(Y|X)$

The joint distribution $P(X, Y)$ can be decomposed in two ways:

$P(X, Y) = P(Y|X)P(X)$

$P(X, Y) = P(X|Y)P(Y)$

Data distribution shift

Covariate shift - $P(X)$ changes, but $P(Y|X)$ remains the same

= the distribution of the input changes, but the conditional probability of a label given an input remains the same

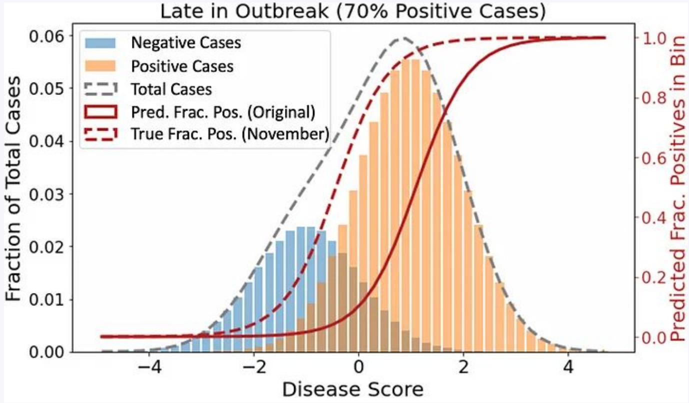
Data distribution shift

Covariate shift - P(X) changes, but P(Y|X) remains the same

= the distribution of the input changes, but the conditional probability of a label given an input remains the same

Label shift (prior shift) - P(Y) changes but P(X|Y) remains the same

= the output distribution changes, but for a given output, the input distribution stays the same



Data distribution shift

Covariate shift - $P(X)$ changes, but $P(Y|X)$ remains the same

= the distribution of the input changes, but the conditional probability of a label given an input remains the same

Label shift (prior shift) - $P(Y)$ changes but $P(X|Y)$ remains the same

= the output distribution changes but for a given output, the input distribution stays the same

Concept drift (posterior shift) - $P(Y|X)$ changes, $P(X)$ remains the same

= same input, different output

Detecting data distribution shifts

How to determine that two distributions are different?

1. Compare statistics: mean, median, variance, quantiles, skewness, …
   - Compute these stats during training and compare these stats in production
   - Not universal: only useful for distributions where these statistics are meaningful
   - Inconclusive: if statistics differ, distributions differ. If statistics are the same, distributions can still differ.

Detecting data distribution shifts

How to determine that two distributions are different?

1. Compare statistics: mean, median, variance, quantiles, skewness, …
    - Compute these stats during training and compare these stats in production
    - Not universal: only useful for distributions where these statistics are meaningful
    - Inconclusive: if statistics differ, distributions differ. If statistics are the same, distributions can still differ.
2. Two-sample hypothesis test
    - Determine whether the difference between two populations is statistically significant
    - If yes, likely from two distinct distributions

E.g.
1. Data from yesterday
2. Data from today

Two-sample test: KS test (Kolmogorov–Smirnov)

- Doesn't make assumptions about distribution
- But only works with one-dimensional data

=> Useful for prediction & label distributions, not so useful for features.

Two-sample test

Drift Detection

Detector	Tabular	Image	Time Series	Text	Categorical Features	Online	Feature Level
Kolmogorov-Smirnov	✓	✓		✓	✓		✓
Maximum Mean Discrepancy	✓	✓		✓	✓	✓	
Learned Kernel MMD	✓	✓		✓	✓		
Least-Squares Density Difference	✓	✓		✓	✓	✓	
Chi-Squared	✓				✓		✓
Mixed-type tabular data	✓				✓		✓
Classifier	✓	✓	✓	✓	✓		
Spot-the-diff	✓	✓	✓	✓	✓		✓
Classifier Uncertainty	✓	✓	✓	✓	✓		
Regressor Uncertainty	✓	✓	✓	✓	✓		

[alibi-detect]

Most tests work better on low-dim data, so dim reduction is recommended beforehand!

ML in production: expectation

1. Collect data
2. Train model
3. Deploy model
4. 

ML in production: reality

1. Choose a metric to optimize
2. Collect data
3. Train model
4. Realize many labels are wrong -> relabel data
5. Train model
6. Model performs poorly on one class -> collect more data for that class
7. Train model
8. Model performs poorly on most recent data -> collect more recent data
9. Train model
10. Deploy model
11. Dream about $$$
12. Wake up at 2am to complaints that model biases against one group -> revert to older version
13. Get more data, train more, do more testing
14. Deploy model
15. Pray
16. Model performs well but revenue decreasing
17. Cry
18. Choose a different metric
19. Start over

Value of data

Integrating ML/AI models in production is difficult.

Survey on a cohort of organizations that began AI pilots before 2019:

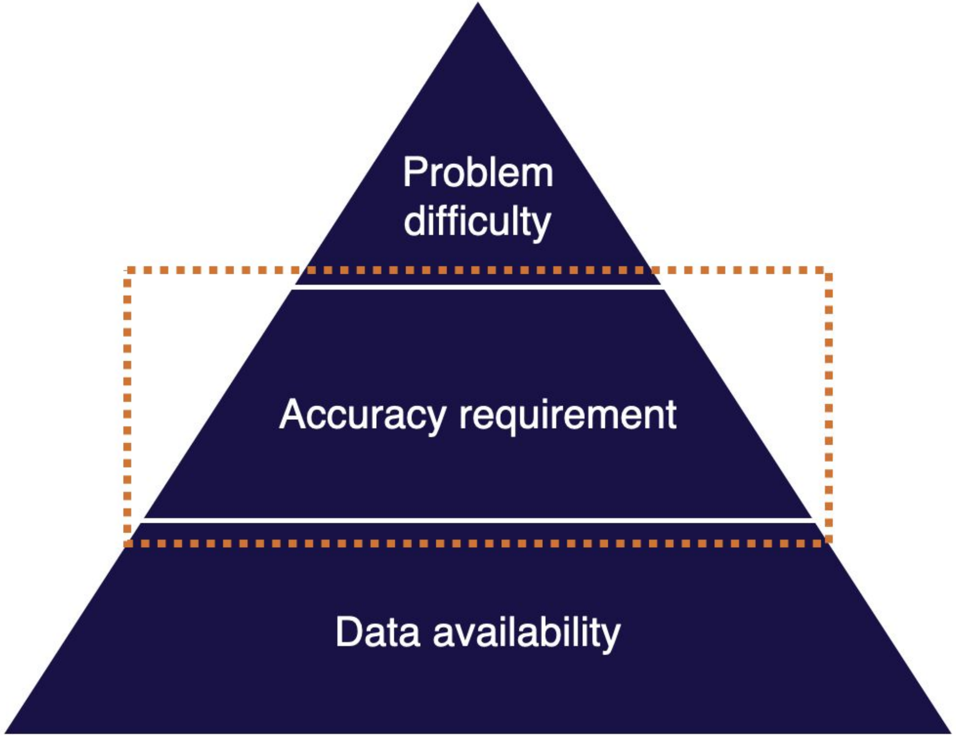
- **only half** of the organizations have moved beyond pilots
- **72%** of organizations have not been able to deploy any application

Challenges:

- a *high degree of manual* and one-off work
- no *reusable or reproducible* components
- difficulties in *handoffs between data scientists and IT*
- challenges in *deployment, scaling, and versioning* efforts

Before designing an ML system:

Cost drivers



Main considerations

- Is the problem well-defined?
- Good published work on similar problems? (newer problems mean more risk & more technical effort)
- Compute requirements?
- Can a human do it?

- How costly are wrong predictions?
- How frequently does the system need to be right to be useful?
- Ethical implications?

- How hard is it to acquire data?
- How expensive is data labeling?
- How much data will be needed?
- How stable is the data?
- Data security requirements?

ML systems

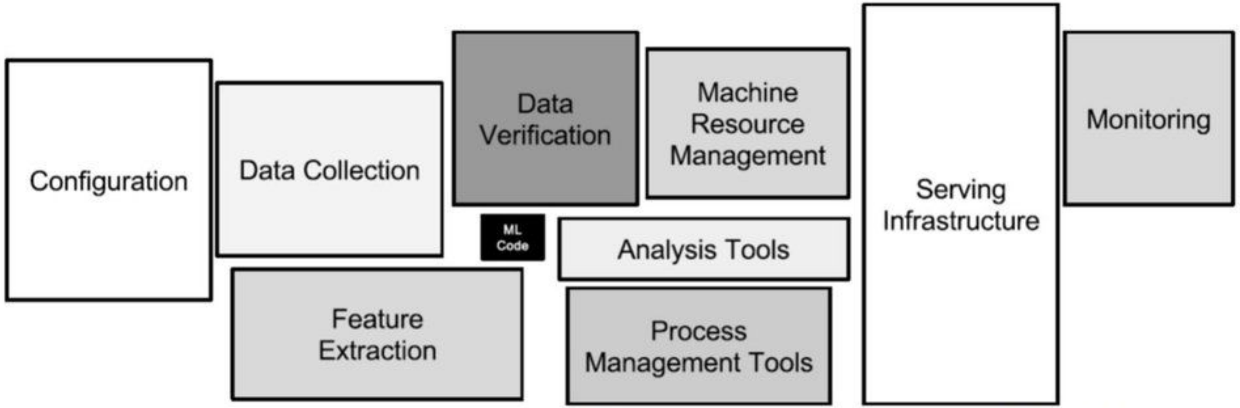
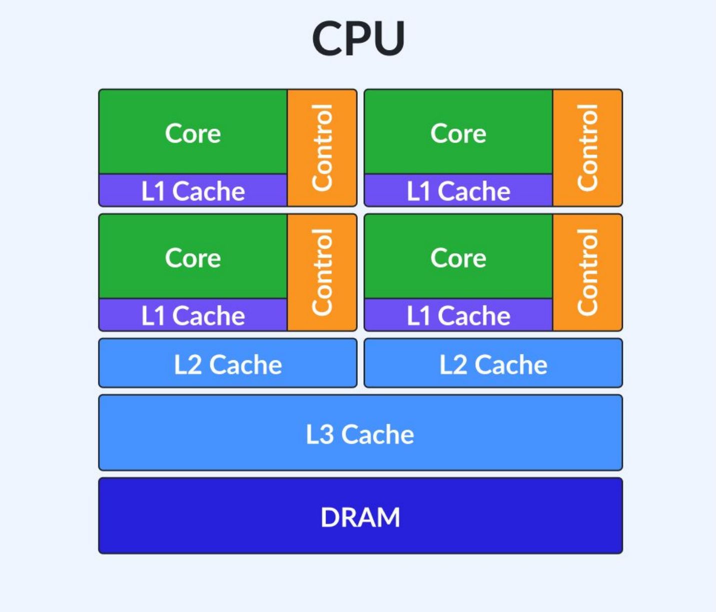


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

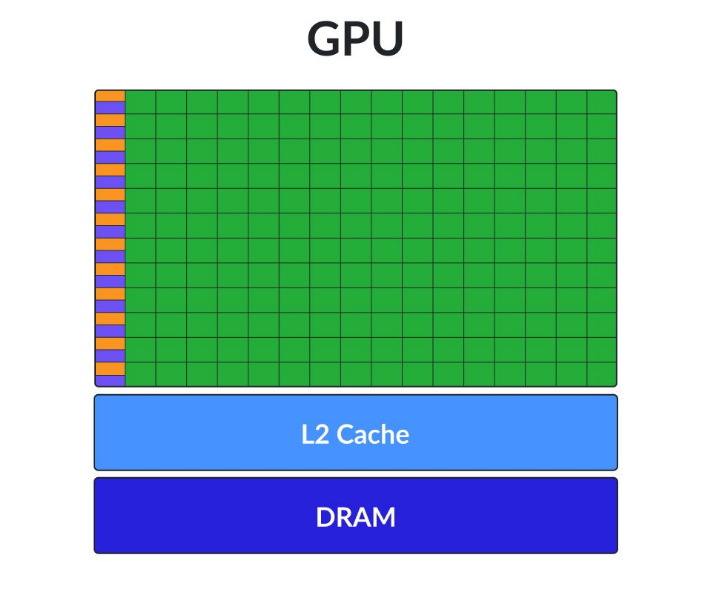
Sculley, David, et al. "Hidden technical debt in machine learning systems." *Advances in neural information processing systems* 28 (2015).

Hardware for data processing and ML

Central Processing Units



Graphics Processing Units



Others: Tensor Processing Units, FPGAs, accelerators etc.

CPUs - Central Processing Units

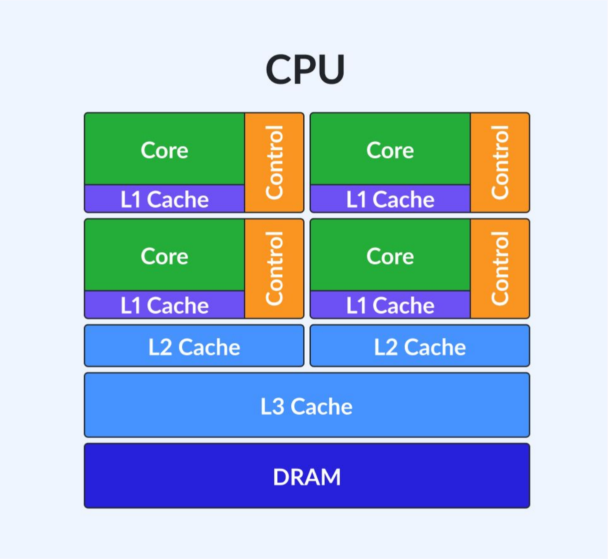
A CPU has relatively few, powerful cores.

Each core :

- is designed for fast sequential processing
- has its own L1 Cache for quick data access
- handles complex instructions
- layers for coordination between cores

Optimized for executing tasks that are:

- **varied**
- **complex**
- **sequential**



GPUs - Graphics Processing Units

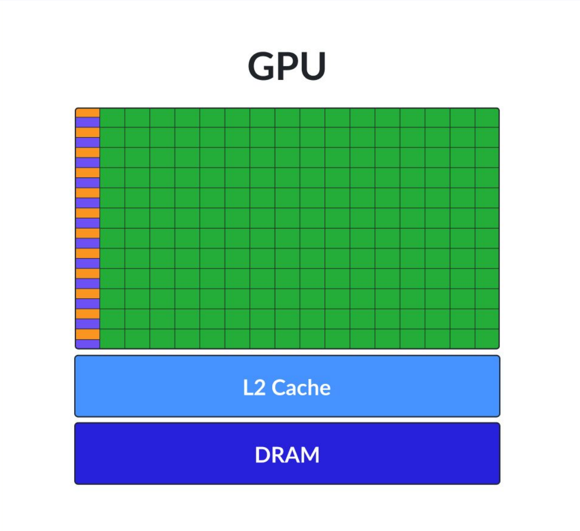
GPU has a large number of small, simple cores.

Each core:

- is individually less powerful than a typical CPU core
- control units manage groups of cores

Designed for massive parallelization:

- doing the **same operation**
- on **many data points**
- **simultaneously**



CPU: "Few workers, each highly skilled"

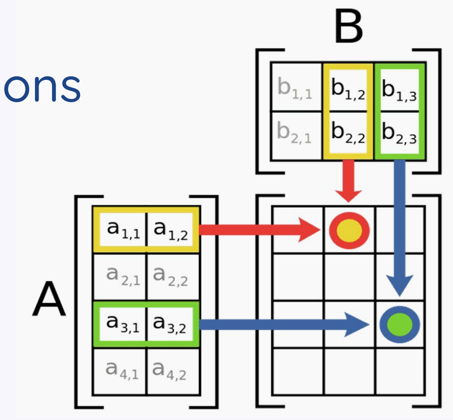
- used for complex, varied tasks

  → data processing, “classical” ML

GPU: "Army of workers doing simple tasks in unison"

- used for repetitive calculations like matrix multiplications

  → deep learning training



video: [NVIDIA GPU vs CPU. 'Mythbusters' stars paint the Mona Lisa in 80 Milliseconds!]

Parallel execution

List Doubler

```
list = [0, 1, 2, 3, 4, 5, ...., 9999] # Initialise list
for 0 .. i .. 9999: # For every element in our list
    list[i] = list[i] * 2 # Double the value
```

The workload is iterative without dependencies.

Cases that are (near-)optimal such as these are called **Embarrassingly Parallel**.

Parallel execution

Cumulative sum

```
list = [0, 1, 2, 3, 4, 5, ...., 9999] # Initialise list
for 0 .. i .. 9999: # For every element in our list
    list[i] = list[i-1] + list[i] # Add the value to the previous total
```

The workload is **completely dependent**.

The algorithm has to be rewritten to be parallelizable.

Performance vs efficiency

Breakthroughs in hardware research have continuously improved performance in data-intensive applications for many decades.

Performance

how fast can I do a task?

Efficiency

how many resources do I need for a task?

Performance vs efficiency: car analogy

Transport 200kg cargo over some distance D

Option 1

  truck, 1 trip, 80km/h

Option 2

  super car, 3 trips (100kg, go back, 100kg), speed S

If S = 240km/h - we can choose either

If S = 400km/h - much faster to use the super car

Germany to investigate suspected 417 km/h Autobahn racer

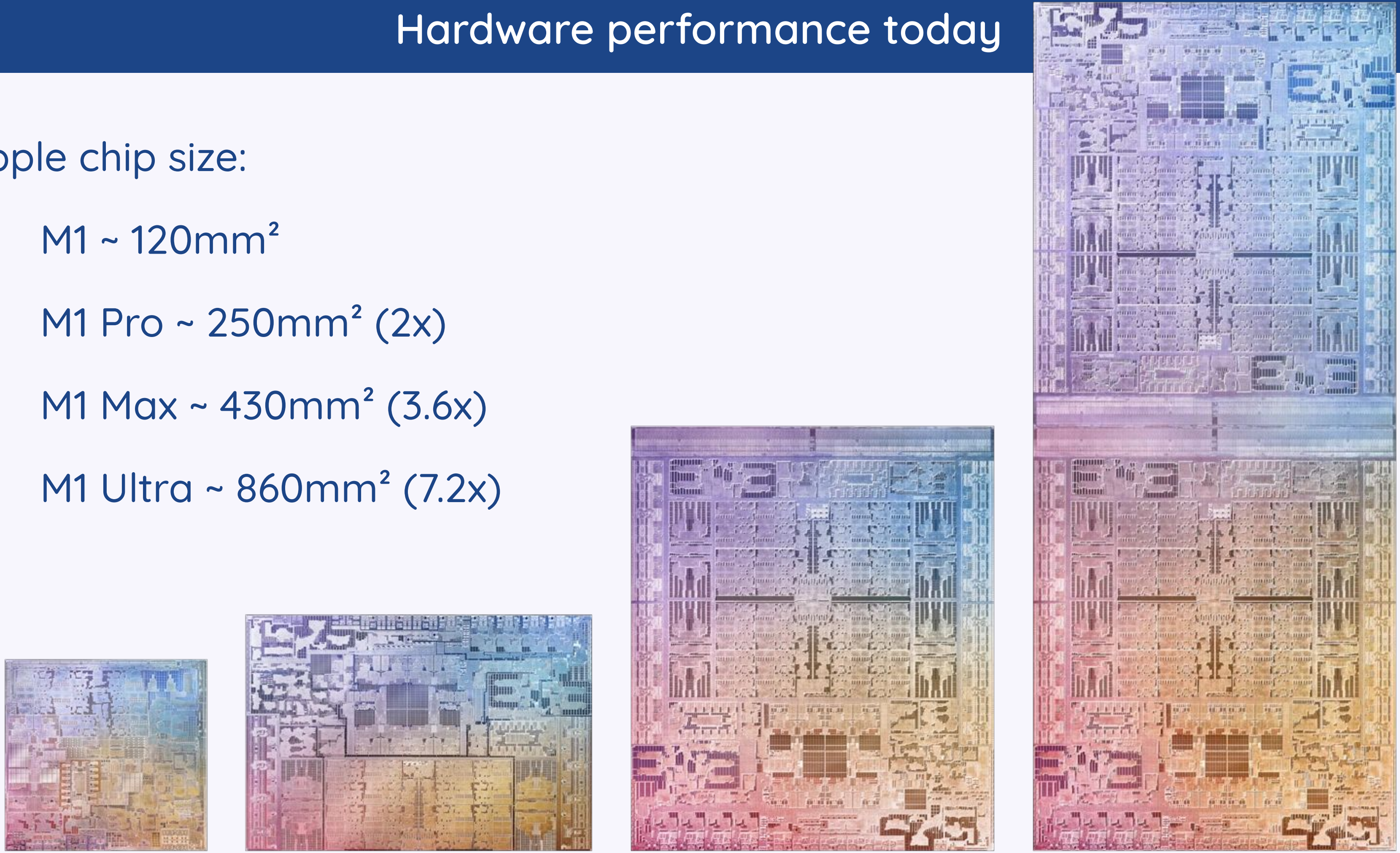
AFP - news@thelocal.de
Published: 7 Feb, 2022 CET. Updated: Mon 7 Feb 2022 17:36 CET

A racing car speeds down the Autobahn. Photo: picture alliance/dpa/Maserati | Lorenzo Marcinno

Hardware performance today

Apple chip size:

- M1 ~ 120mm^2
- M1 Pro ~ 250mm^2 (2x)
- M1 Max ~ 430mm^2 (3.6x)
- M1 Ultra ~ 860mm^2 (7.2x)



M1 M1 Pro M1 Max M1 Ultra

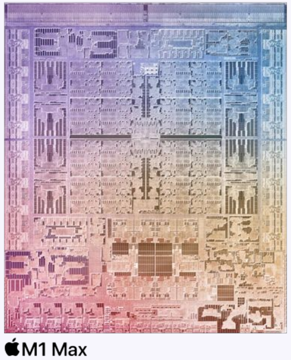
Thermal design power (TDP)

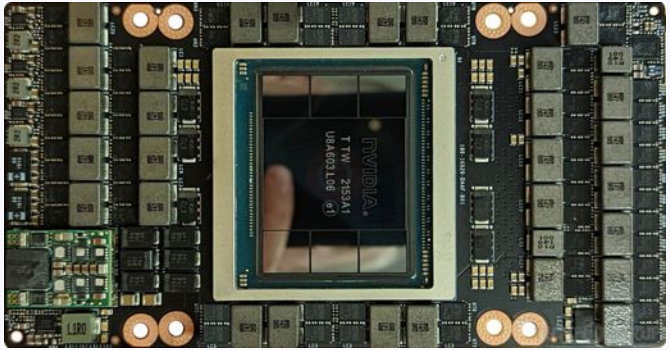
What resources are needed for compute?

Thermal design power = power needed under a compute intensive workload.

Intel: the maximum power consumption under a sustained workload

AMD: maximum cooling power required





M1 Max

- 92W
- 430mm²
- TDP/mm² = 0.21

NVIDA H100 SMX

- 700W
- 800mm²
- TDP/mm² = 0.86

Hot plate

- 1520W
- 25400mm²
- TDP/mm² = 0.06

Performance of Intel CPUs

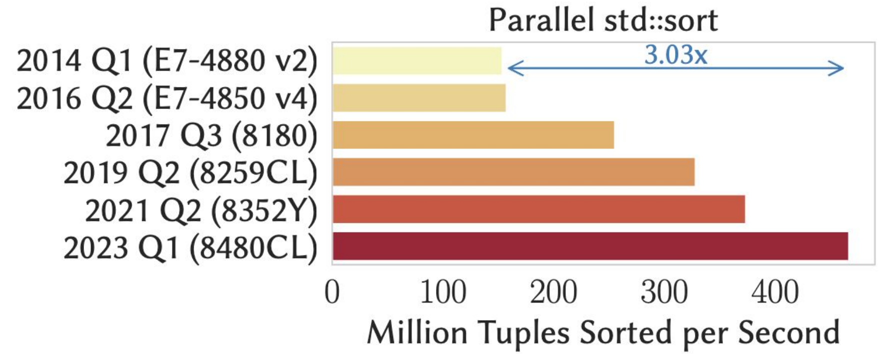
Performance for **sorting**

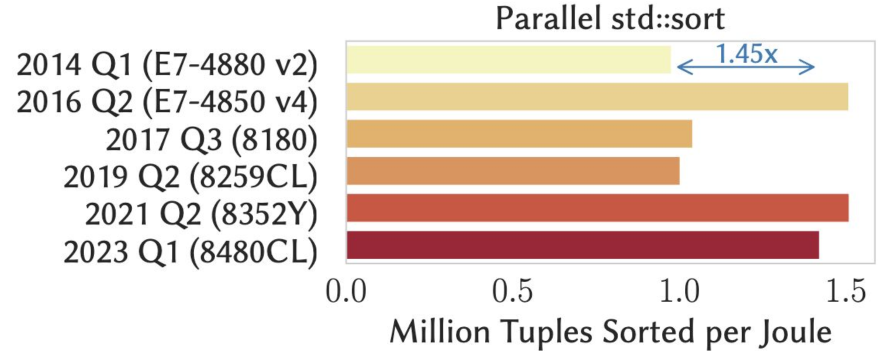
(hard to parallelise,

used in many data systems)

Improvement in **performance**: 3x

Improvement in **energy efficiency**:
1.45x in about 10 years





Carbon emissions

Carbon emissions for a compute system can be estimated using:

- the carbon intensity of the local energy grid
- the energy consumption of the system.

Example:

Run computation on an Intel Xeon Platinum 8480+ with **TDP = 0.37kW**

placed in Denmark when the **carbon intensity ≃ 156g CO_2eq/kWh**:

$$0.37\text{kW} \times 156\text{g } CO_2\text{eq/kWh} \simeq \mathbf{58g\ CO_2eq/h}$$

Carbon intensity

Carbon intensity = a measure of the greenhouse gas emissions associated with producing electricity

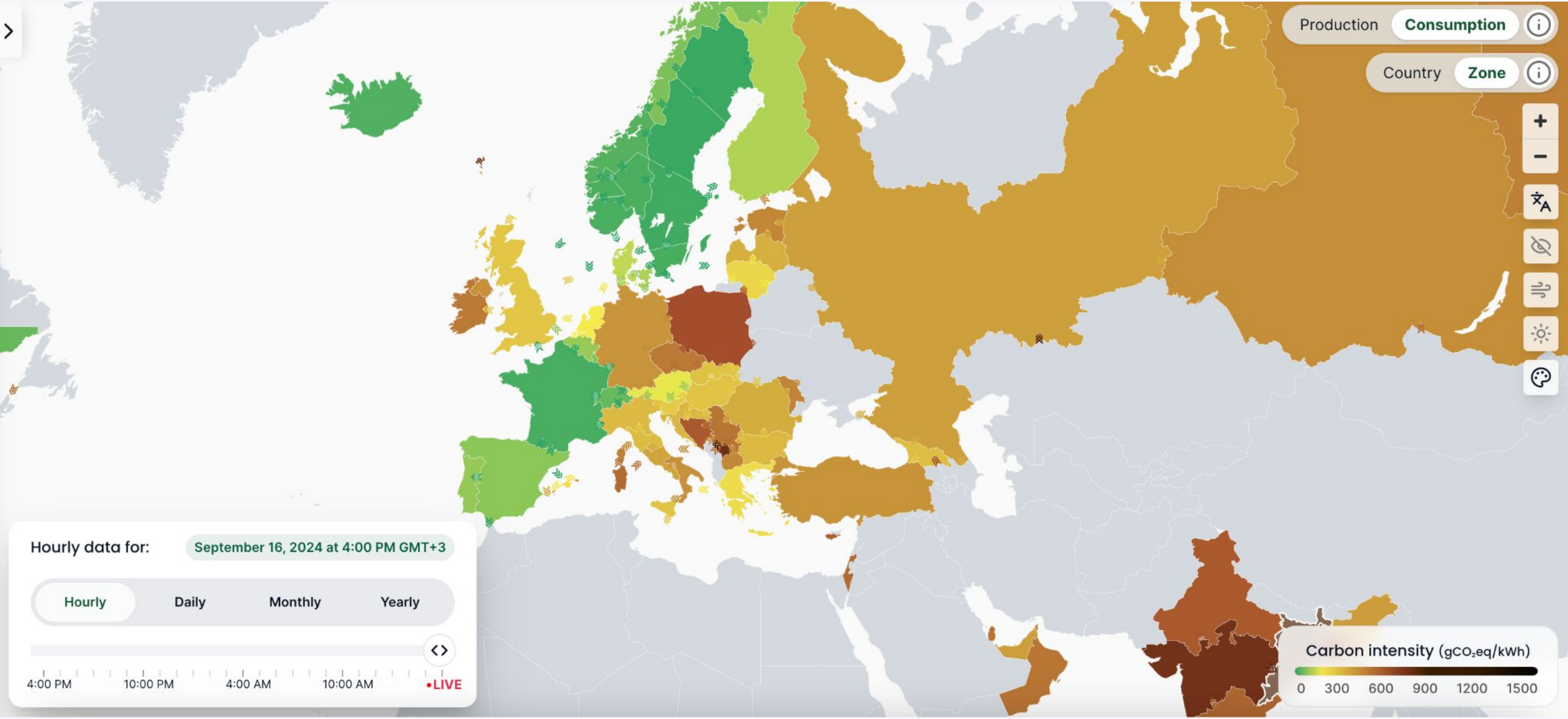
grams of carbon dioxide (CO_2) released in the atmosphere

for each

kilowatt hour (kWh) of electricity consumed

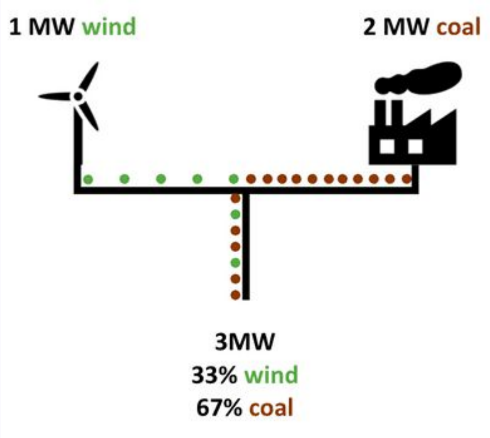
Each type of greenhouse gas can be converted to its *CO_2 equivalent* in terms of global warming potential for 100 years.

Carbon intensity - electricitymaps.com



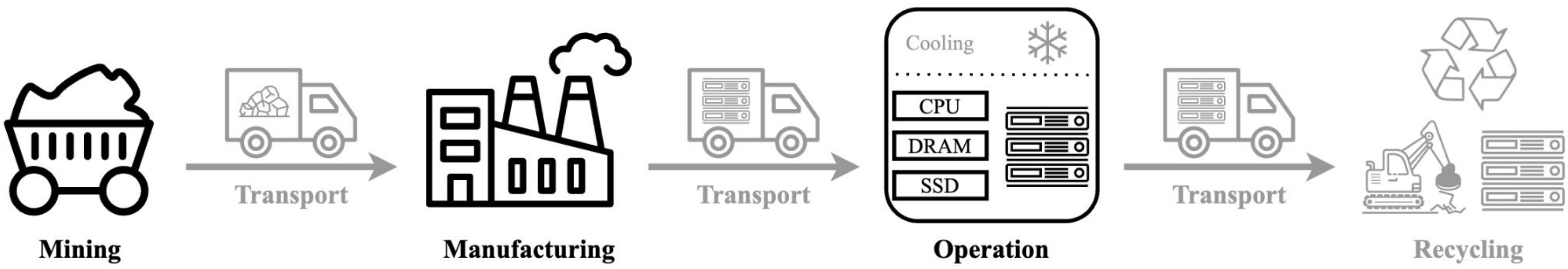
Emission factors

= factors for the whole lifecycle of consumed fuels and used power plants



Mode	Life-cycle Emission factor (gCO2eq/kWh)	Operational Emission factor (gCO2eq/kWh)	Category	Source
battery discharge	301	TBD	Renewable (default)	World average intensity by Electricity Maps
biomass	230	0	Renewable	IPCC 2014
coal	820	760	Fossil	IPCC 2014
gas	490	370	Fossil	IPCC 2014
geothermal	38	0	Renewable	IPCC 2014
hydro	24	0	Renewable	IPCC 2014
hydro discharge	301	TBD	Renewable (default)	World average intensity by Electricity Maps
nuclear	12	0	Low-carbon	IPCC 2014
oil	650	406	Fossil	UK Parliamentary Office of Science and Technology
solar	45	0	Renewable	IPCC 2014
unknown	700	575	Fossil	Assumes thermal (coal, gas, oil)
wind	11	0	Renewable	IPCC 2014

Lifecycle of a cloud server



Server footprint = embodied + operational

- Embodied = the footprint of producing a piece of hardware
- Operational = the footprint of using the hardware
  - (CPU x utilization + DRAM + SSD)x Carbon Intensity

Energy reporting in ML

Towards the Systematic Reporting of the Energy and Carbon Footprints of Machine Learning

Henderson et al.

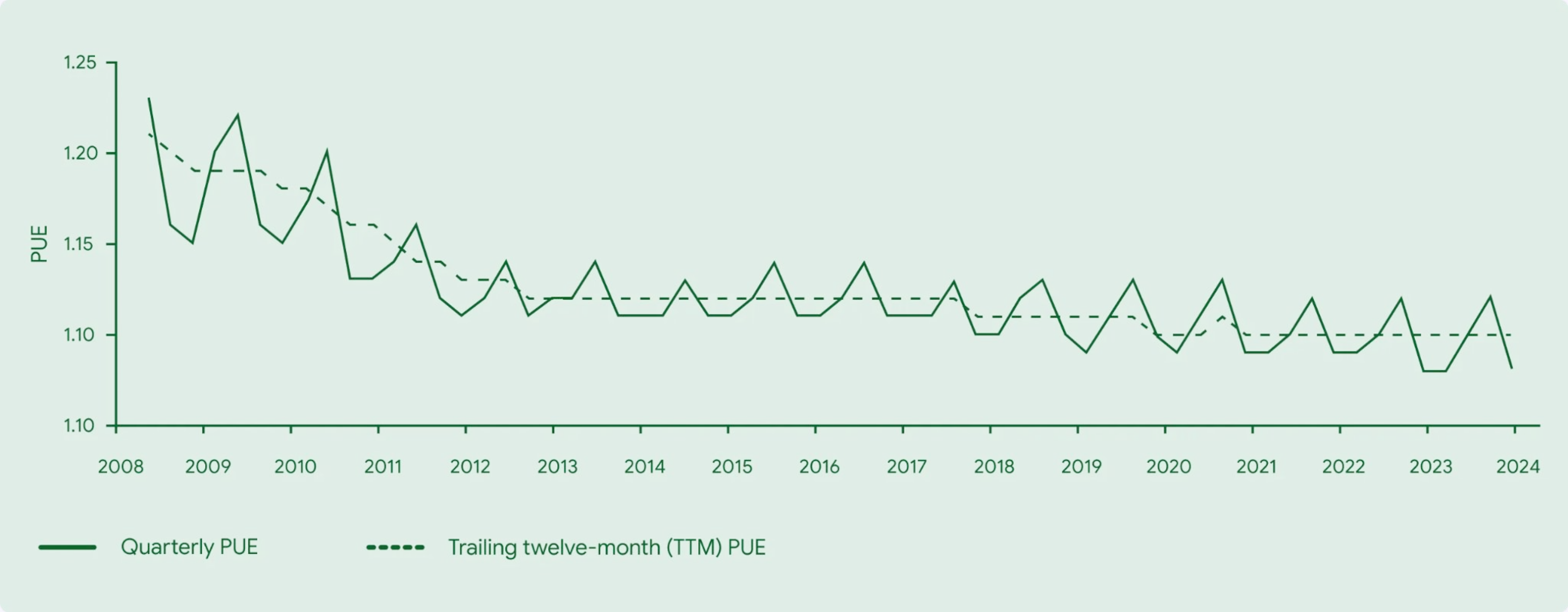
Typically energy consumption reporting is constrained to that of CPUs, GPUs, RAM

Power usage effectiveness (PUE) factor - accounts for the additional energy required to support the compute infrastructure

$$PUE = \frac{\text{Total Facility Energy}}{\text{IT Equipment Energy}} = 1 + \frac{\text{Non IT Total Facility Energy}}{\text{IT Equipment Energy}}$$

Data center trends

Data centers are becoming more efficient in terms of cooling and operation

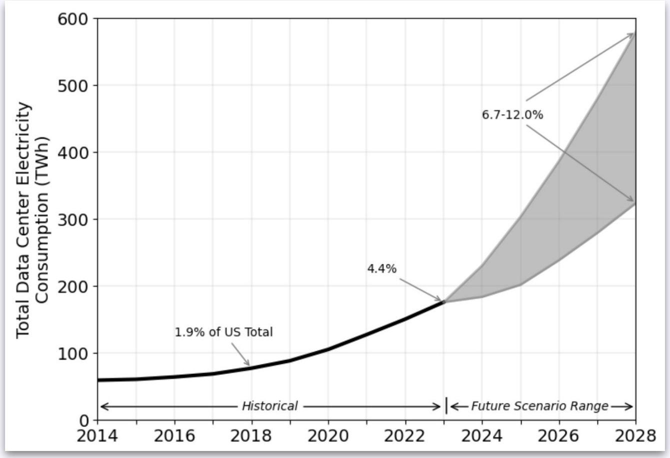


Data center trends

> *The Cloud now has a greater carbon footprint than the airline industry. A single data center can consume the equivalent electricity of 50,000 homes.*

Currently, all major cloud providers are missing carbon reduction targets and increase their footprints

Major factor: new hardware deployment



Compute-related impacts for AI/ML

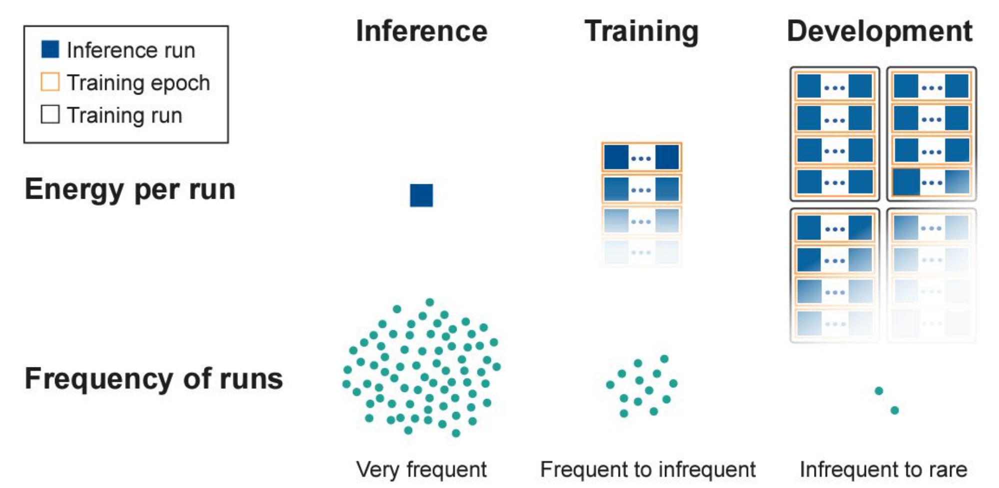
Energy needed for ML computations + embodied emissions associated with hardware

ML compute:

Amount of computing power varies dramatically between both applications and users

Lifecycle of an ML model:

- inference
- training
- development and tuning



Kaack, Lynn H., et al. "**Aligning artificial intelligence with climate change mitigation.**" *Nature Climate Change* 12.6 (2022): 518-527.

Carbon footprint of ML models

Towards the Systematic Reporting of the Energy and Carbon Footprints of Machine Learning

Peter Henderson PHEND@CS.STANFORD.EDU
Stanford University, Stanford, CA, USA

Jieru Hu JIERU@FB.COM
Facebook, Menlo Park, CA, USA

Joshua Romoff JOSHUA.ROMOFF@MAIL.MCGILL.CA
Mila, McGill University, Montreal, QC, Canada

Emma Brunskill EBRUN@CS.STANFORD.EDU
Stanford University, Stanford, CA, USA

Dan Jurafsky JURAFSKY@STANFORD.EDU
Stanford University, Stanford, CA, USA

Joelle Pineau JPINEAU@CS.MCGILL.CA
Facebook AI Research, Mila, McGill University, Montreal, QC, Canada

Hyperparameter Power Impact in Transformer Language Model Training

Lucas Høyberg Puvis de Chavannes
IT University of Copenhagen
M47 Labs
lucas.puvis@m47labs.com

Mads Kongsbak
IT University of Copenhagen
mkon@itu.dk

Timmie Mikkel Rantzau Lagermann
IT University of Copenhagen
timn@itu.dk

Leon Derczynski
IT University of Copenhagen
ld@itu.dk

Carbontracker: Tracking and Predicting the Carbon Footprint of Training Deep Learning Models

Lasse F. Wolff Anthony*[1] **Benjamin Kanding*[1]** **Raghavendra Selvan[1]**

Efficient Methods for Natural Language Processing: A Survey

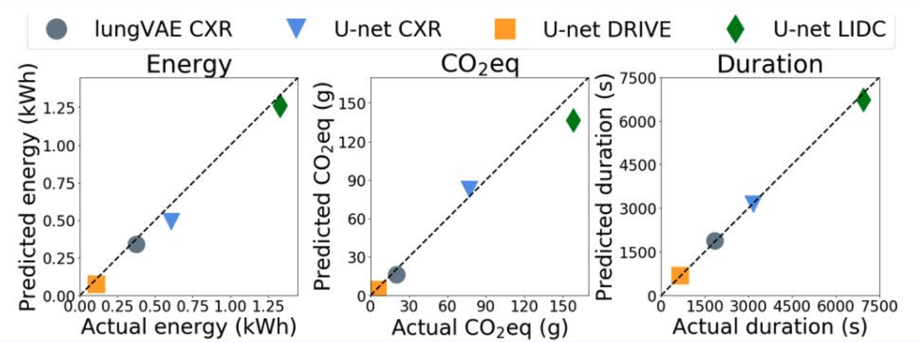
Marcos Treviso[10]*, Tianchu Ji[3]*, Ji-Ung Lee[7]*, Betty van Aken[8], Qingqing Cao[2], Manuel R. Ciosici[9], Michael Hassid[1], Kenneth Heafield[13], Sara Hooker[5], Pedro H. Martins[10], André F. T. Martins[10], Peter Milder[3], Colin Raffel[6], Edwin Simpson[4], Noam Slonim[12], Niranjan Balasubramanian[3], Leon Derczynski[11], Roy Schwartz[1]

[1]The Hebrew University of Jerusalem, [2]University of Washington, [3]Stony Brook University, [4]University of Bristol, [5]Cohere For AI, [6]University of North Carolina at Chapel Hill, [7]Technical University of Darmstadt, [8]Berliner Hochschule für Technik, [9]University of Southern California, [10]IST/University of Lisbon & Instituto de Telecomunicações, [11]IT University of Copenhagen, [12]IBM Research, [13]University of Edinburgh

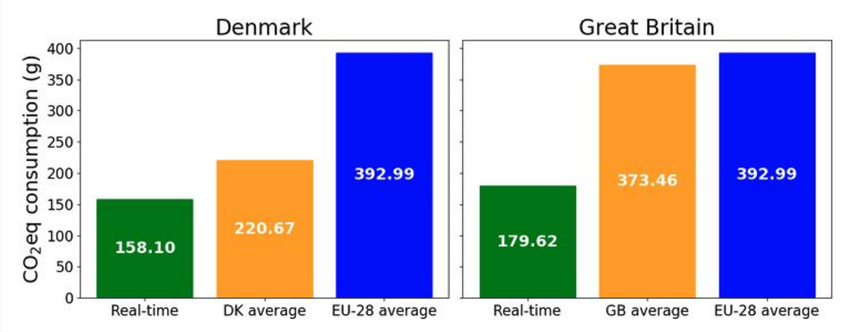
Carbontracker: Tracking and Predicting the Carbon Footprint of Training Deep Learning Models

Lasse F. Wolff Anthony* [1] Benjamin Kanding* [1] Raghavendra Selvan [1]

A tool for tracking and predicting the energy and carbon footprint of training models



Comparison of predicted and measured values



Carbon emissions of training in Denmark and Great Britain based on different carbon intensity estimation methods

Large increase in emissions

Generative AI is reportedly tripling carbon dioxide emissions from data centers

News By Ellen Jennings-Trace published September 9, 2024

Research suggest data centers will emit 2.5 billion tons of greenhouse gas by 2030 [Link]

Google says its emissions have grown nearly 50% due to AI data center boom — and here's what it plans to do about it

News By Craig Hale published July 3, 2024

Will Google meet its emissions target? [Link]

Water cooling in data centers

AI push from US tech giants leads to skyrocketing water consumption figures

News By Ellen Jennings-Trace published August 19, 2024

Millions of gallons being consumed each month in water-scarce areas

Link

Water is primarily used in data centres to cool hardware, using air mechanisms to dissipate heat, causing up to nine liters of water to evaporate per kWh of energy used.

Water cooling in data centers

AI push from US tech giants leads to skyrocketing water consumption figures

News By Ellen Jennings-Trace published August 19, 2024

Millions of gallons being consumed each month in water-scarce areas

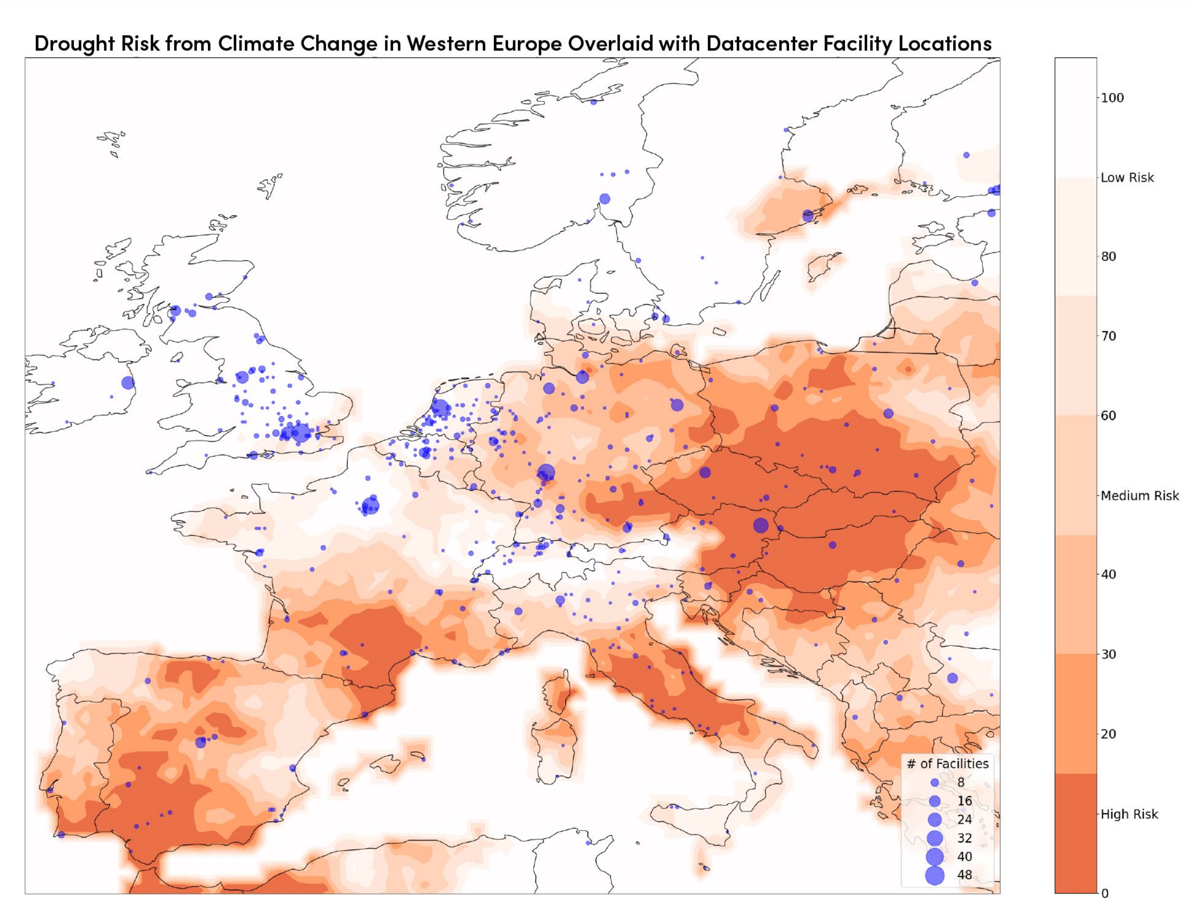
Link

Water is primarily used in data centres to **cool hardware**, using air mechanisms to dissipate heat, causing up to nine liters of water to evaporate per kWh of energy used.

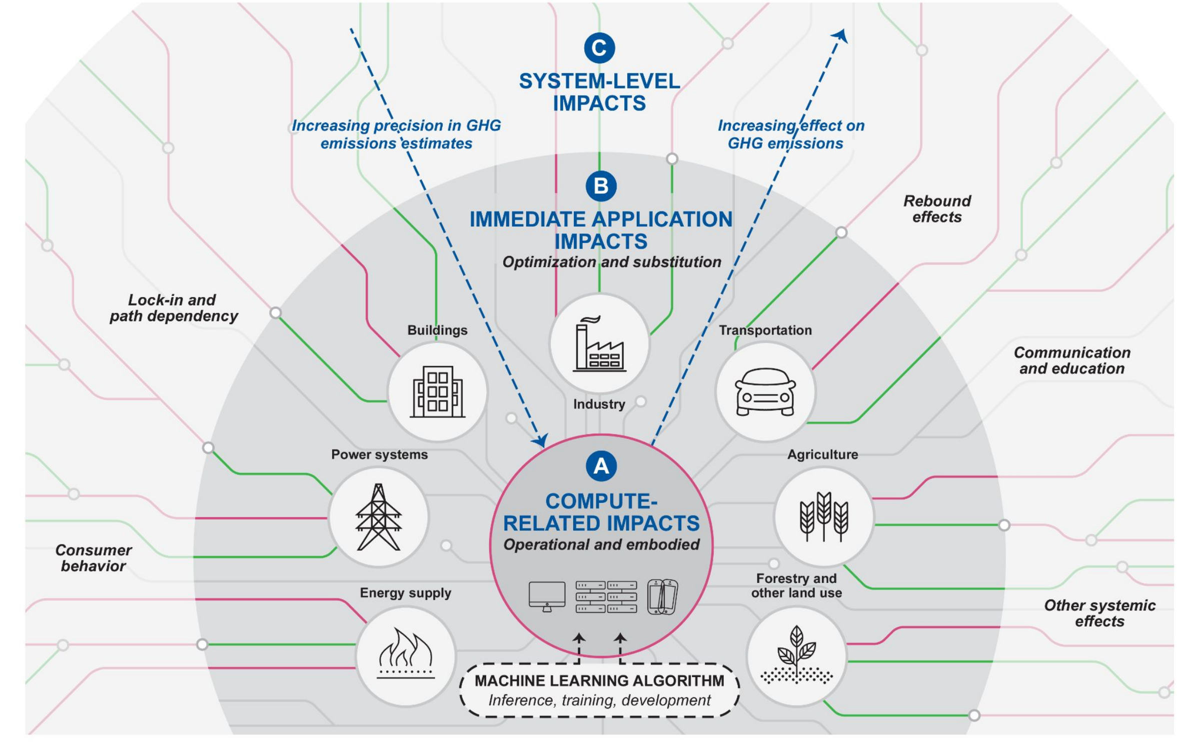
A month before OpenAI finished training GPT-4, a data centre cluster in West Des Moines, Iowa, consumed 6% of the district's water, according to a lawsuit filed by its residents.

Link

Drought risk and data centers locations



A framework for assessing greenhouse gas (GHG) emissions impacts of Machine Learning



Kaack, Lynn H., et al. "**Aligning artificial intelligence with climate change mitigation.**" *Nature Climate Change* 12.6 (2022): 518-527.

Impact of AI/ML - negative effects

Effects associated with computation

- energy (both of running software and maintaining infrastructure)
- indirect emissions (producing hardware)

Facilitation of activities associated with high emissions

- oil and gas exploration and extraction
- advertising industry - consumption

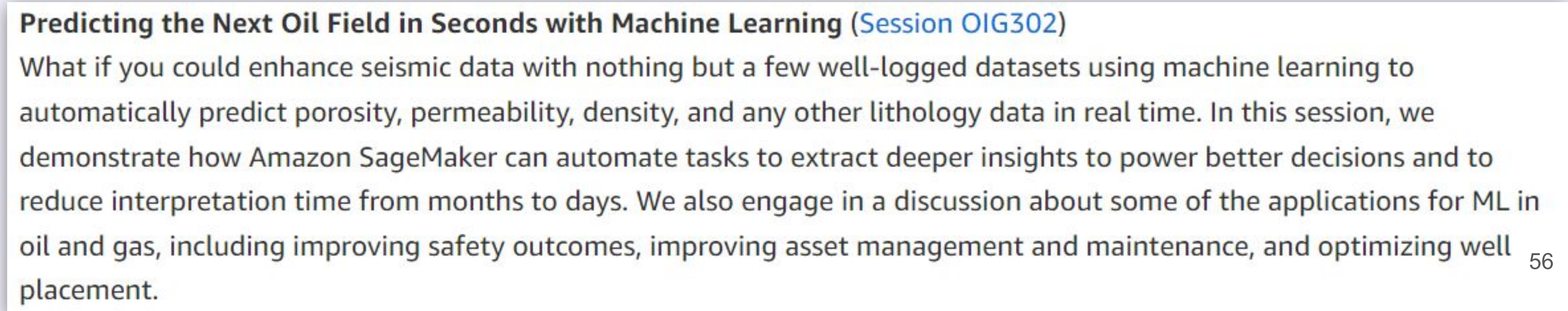
Impact of AI/ML - negative effects

Effects associated with computation

- energy (both of running software and maintaining infrastructure)
- indirect emissions (producing hardware)

Facilitation of activities associated with high emissions

- oil and gas exploration and extraction
- advertising industry - consumption



Predicting the Next Oil Field in Seconds with Machine Learning (Session OIG302)

What if you could enhance seismic data with nothing but a few well-logged datasets using machine learning to automatically predict porosity, permeability, density, and any other lithology data in real time. In this session, we demonstrate how Amazon SageMaker can automate tasks to extract deeper insights to power better decisions and to reduce interpretation time from months to days. We also engage in a discussion about some of the applications for ML in oil and gas, including improving safety outcomes, improving asset management and maintenance, and optimizing well placement.

Impact of AI/ML - negative effects

Effects associated with computation

- energy (both of running software and maintaining infrastructure)
- indirect emissions (producing hardware)

Facilitation of activities associated with high emissions

- oil and gas exploration and extraction
- advertising industry - consumption

Wider societal impact

- inequality, discrimination can be perpetuated and amplified through AI systems, privacy

Wider impact - system level effects

Structural rebound effects:

increase individual efficiency => increase overall energy use

In the case of autonomous driving:

- longer and more frequent trips
- living further in the suburbs
- higher quality of assistance systems

Wider societal impact - loss of privacy

We will discuss privacy in terms of uniqueness

How easy is it to de-anonymize a big dataset?

Living in a surveillance economy

Personal data is:

- being collected en masse
- sold to 3rd parties
- anonymised, but how secure is it?

Sci Rep. 2021; 11: 3861. PMCID: PMC7887199
Published online 2021 Feb 16. doi: 10.1038/s41598-021-82294-1 PMID: 33594096

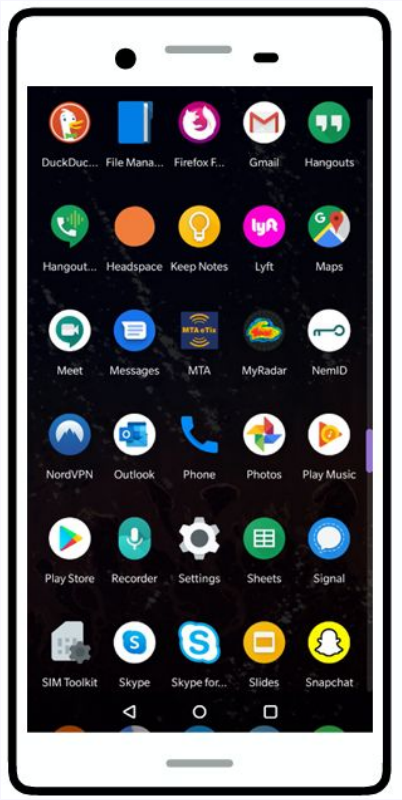
Temporal and cultural limits of privacy in smartphone app usage

Vedran Sekara,[1,2] Laura Alessandretti,[1,3,4] Enys Mones,[1,3] and Håkan Jonsson[1,5]

Dataset:

- 3.5 million people
- 1 years of usage
- 1.1 million apps (~30% of all apps)
- approved by appropriate ethics boards
- only data from users that consented

Temporal and cultural limits of privacy in smartphone app usage - the dataset

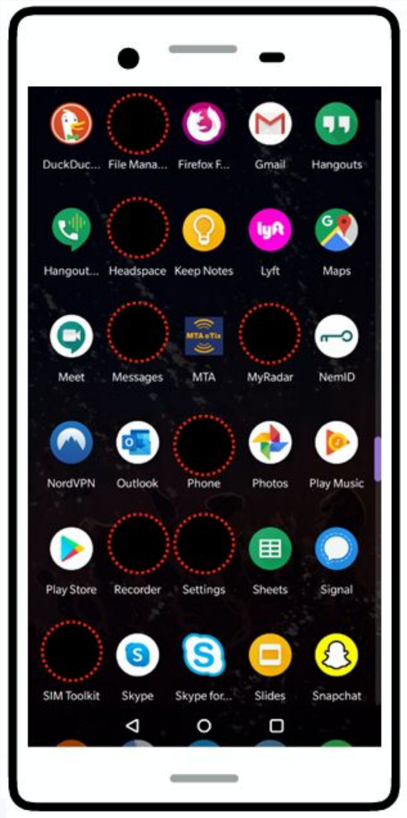


- binary labels (y/n = used/not used) for every month

Estimating unicity

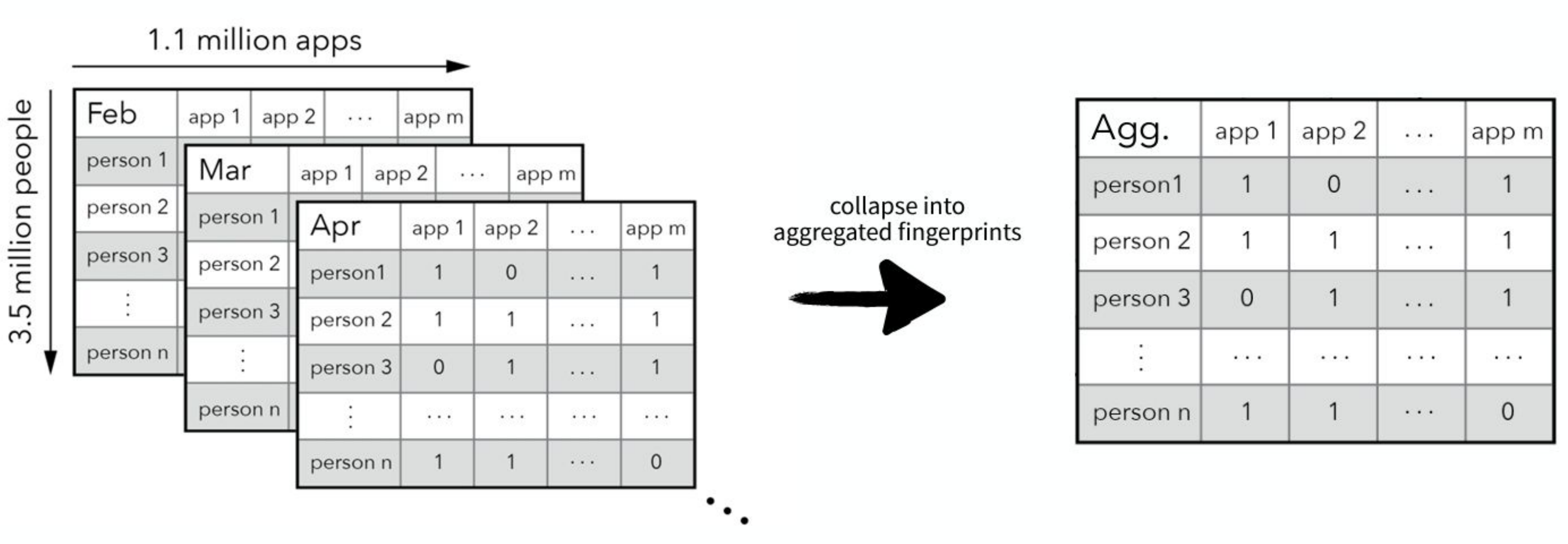
- randomly choose an individual
- sample *n* apps (using any heuristic)
- check if a person is the only one with those apps
- count the number of “re-identifications”

Temporal and cultural limits of privacy in smartphone app usage - the dataset

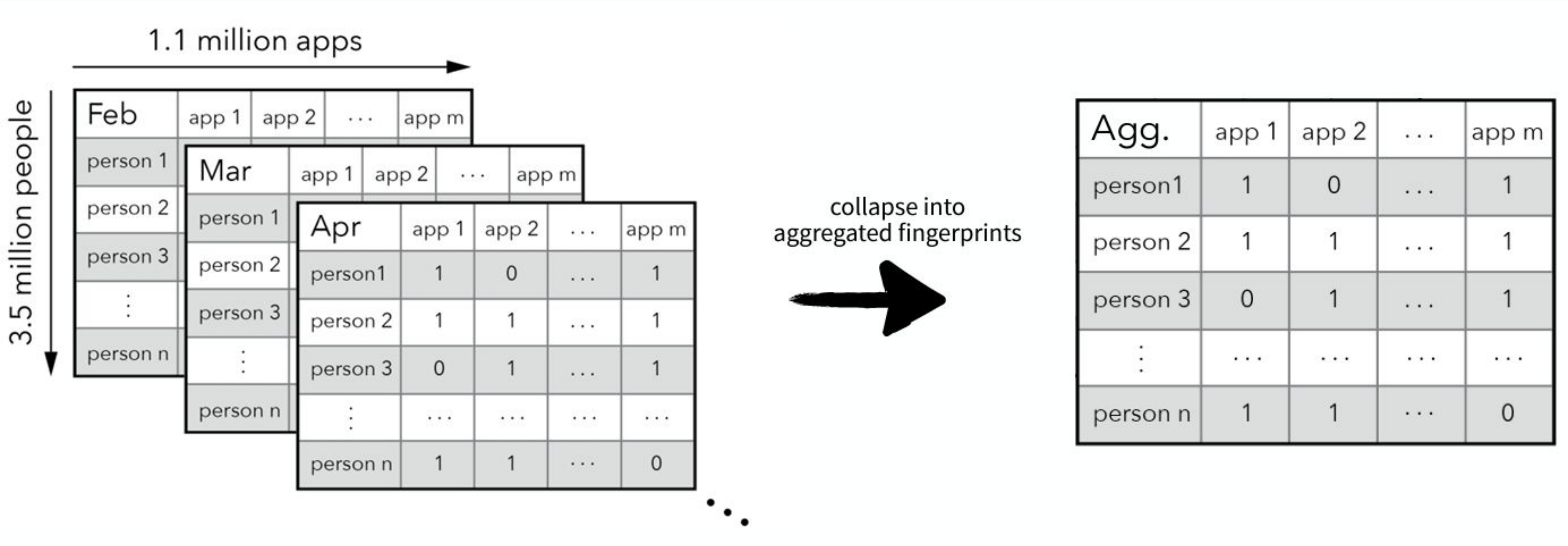


- binary labels (y/n = used/not used) for every month
- disregard temporal information
- removed vendor specific apps
- only apps from Google Play
- only used apps (not installed)

Temporal and cultural limits of privacy in smartphone app usage - the dataset



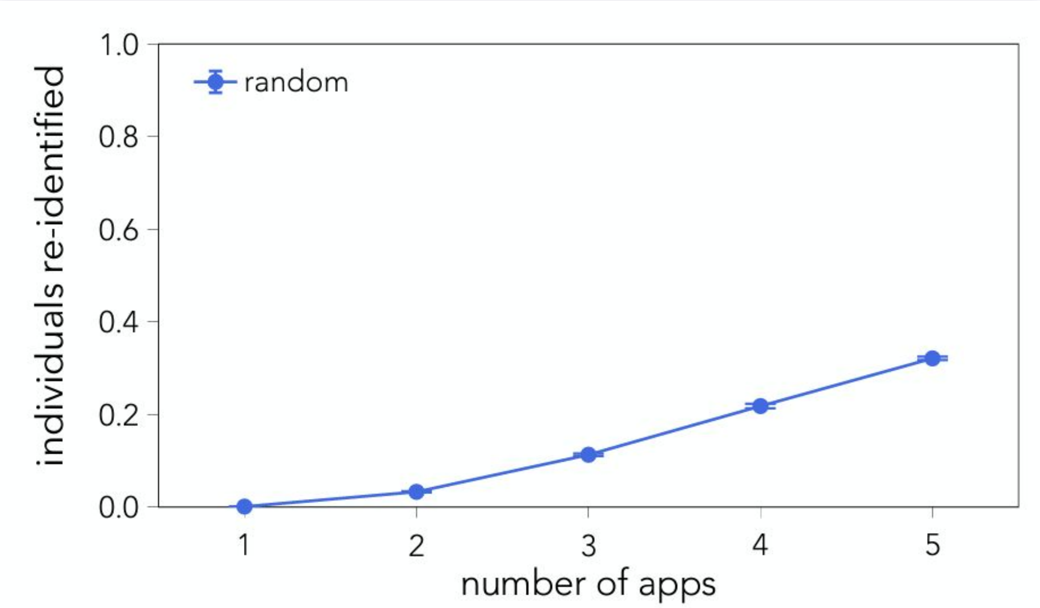
Temporal and cultural limits of privacy in smartphone app usage - the dataset



Look for *n* data points in the data that uniquely identify an individual

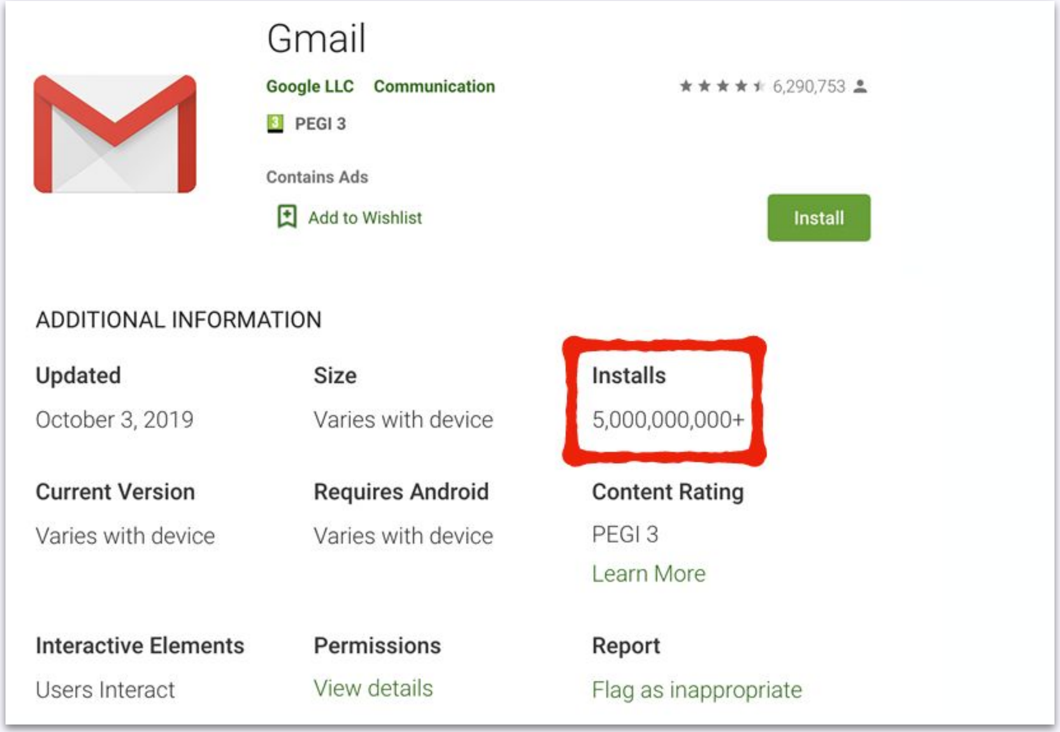
Estimating unicity

Under the random attack scheme, the assumption is that an attacker has information about *n* apps selected at random from the pool of an individual's mobile phone apps.



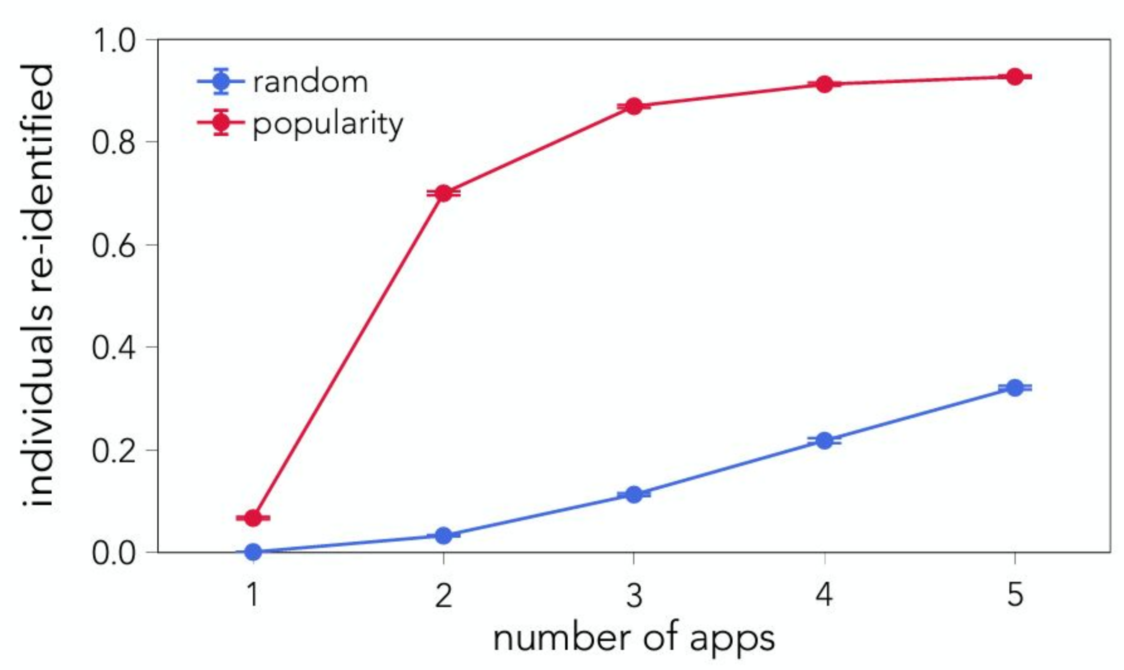
Estimating unicity

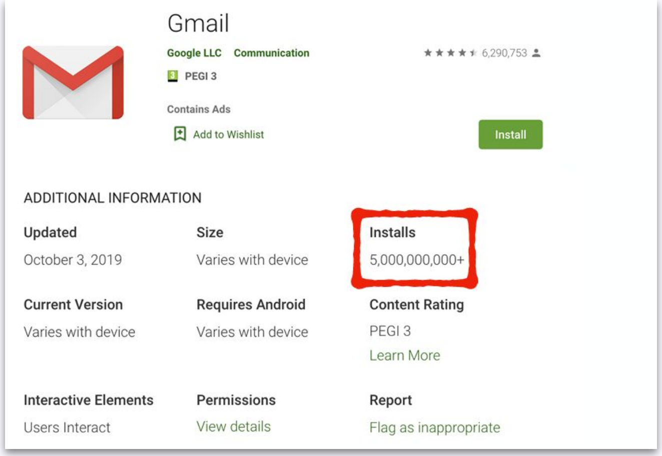
Under the popularity attack scheme, an attacker would know the n least popular app used by a person.



Estimating unicity

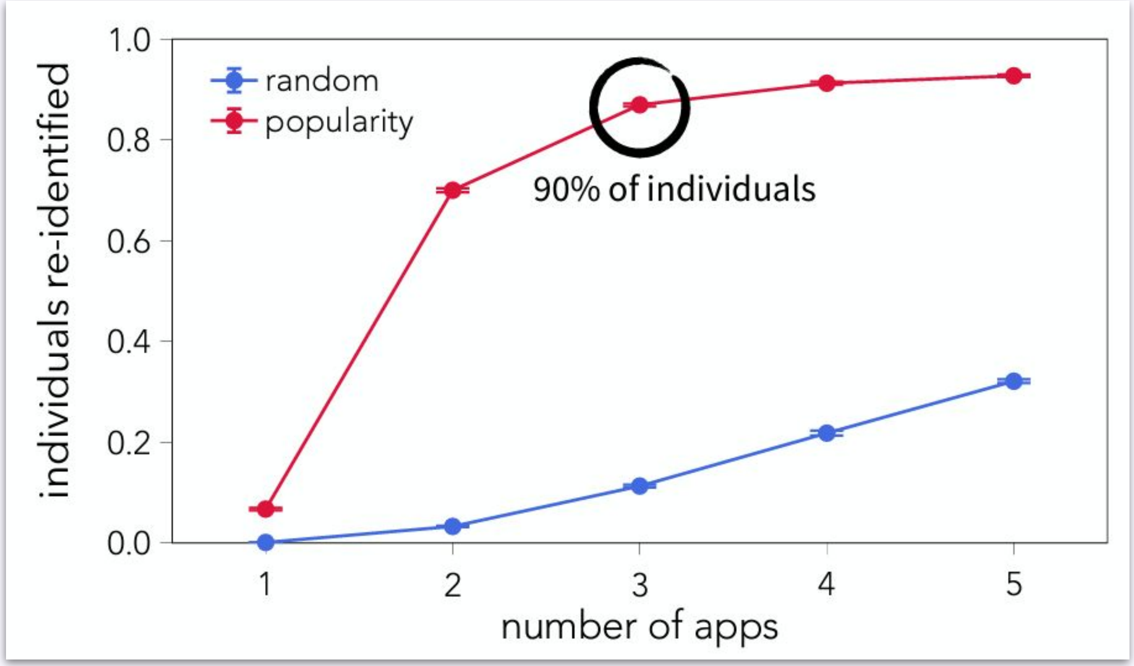
additional data - app popularity

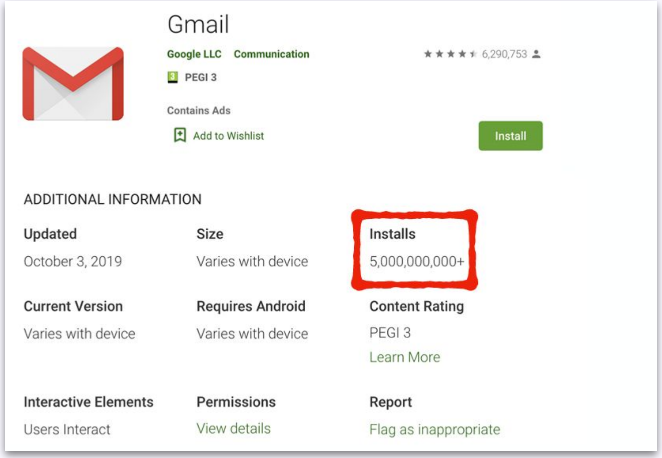




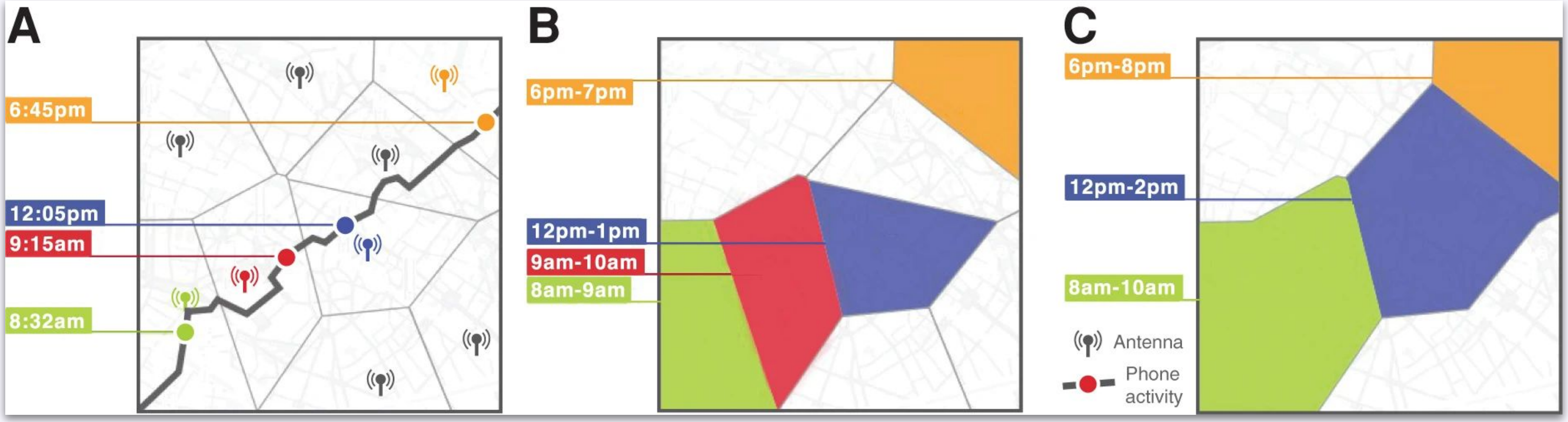
Estimating unicity

additional data - app popularity



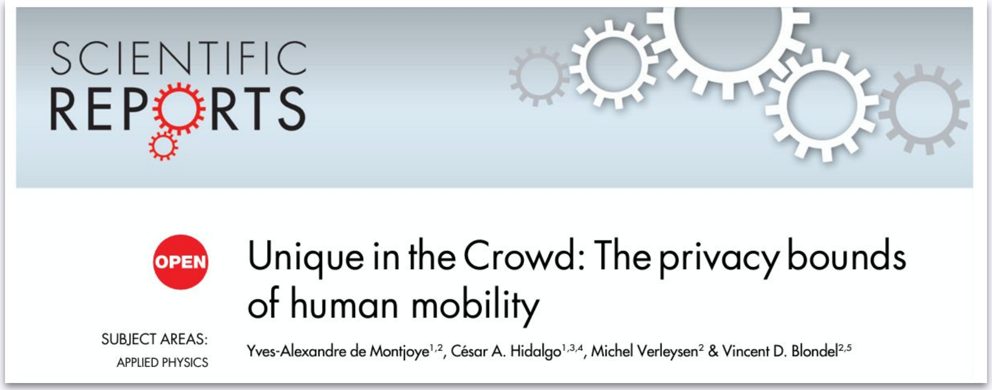


Similarly: mobility patterns



Location data from

1.5 million people


SCIENTIFIC REPORTS

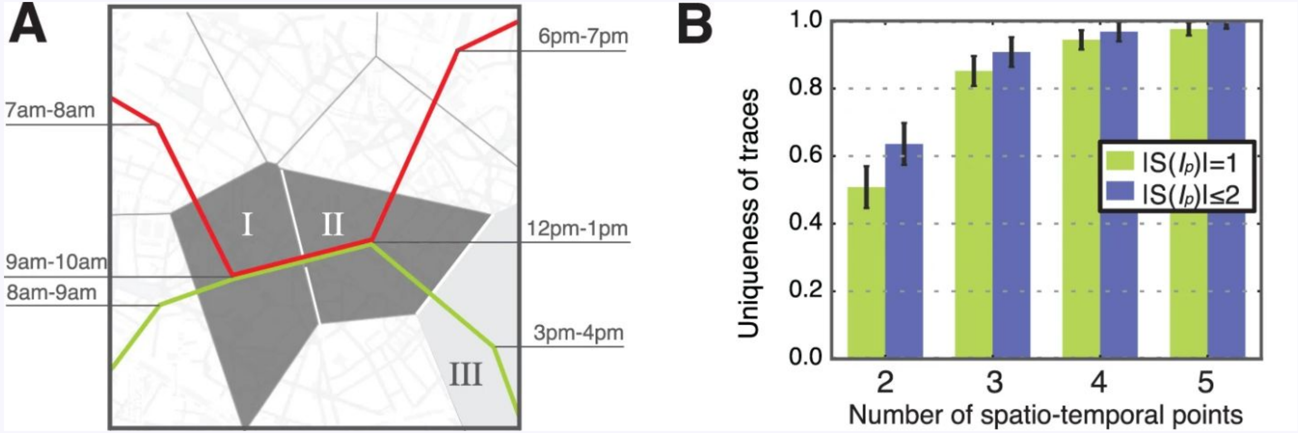
OPEN

SUBJECT AREAS:
APPLIED PHYSICS

Unique in the Crowd: The privacy bounds of human mobility

Yves-Alexandre de Montjoye[1,2], César A. Hidalgo[1,3,4], Michel Verleysen[2] & Vincent D. Blondel[2,5]

Similarly: mobility patterns



Knowing as few as four spatio-temporal points taken at random (Ip = 4) is enough to uniquely characterize 95% of the traces amongst 1.5 M users.

If you'd like to read more

Sci Rep. 2021; 11: 3861. PMCID: PMC7887199
Published online 2021 Feb 16. doi: 10.1038/s41598-021-82294-1 PMID: 33594096

Temporal and cultural limits of privacy in smartphone app usage

Vedran Sekara,[1,2] Laura Alessandretti,[1,3,4] Enys Mones,[1,3] and Håkan Jonsson[1,5]

SCIENTIFIC REPORTS

OPEN

SUBJECT AREAS:
APPLIED PHYSICS

Unique in the Crowd: The privacy bounds of human mobility

Yves-Alexandre de Montjoye[1,2], César A. Hidalgo[1,3,4], Michel Verleysen[2] & Vincent D. Blondel[2,5]

IDENTITY AND PRIVACY

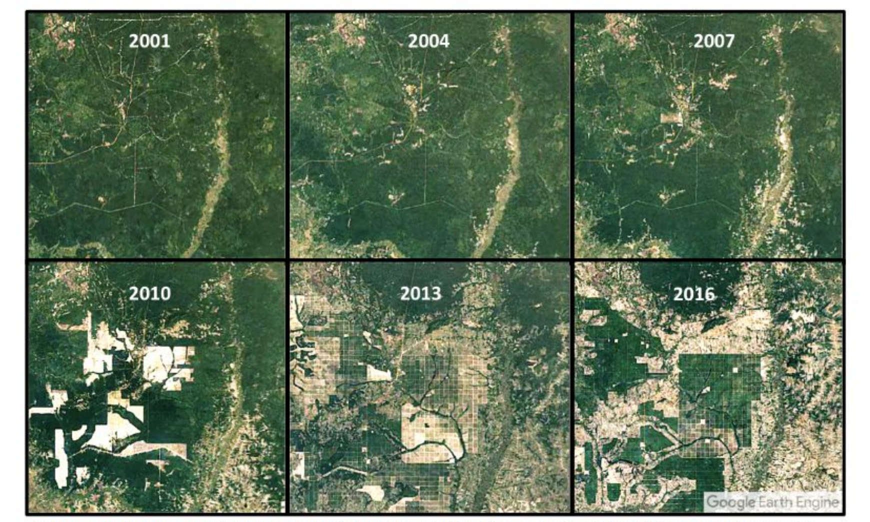
Unique in the shopping mall: On the reidentifiability of credit card metadata

Yves-Alexandre de Montjoye,[1*] Laura Radaelli,[2] Vivek Kumar Singh,[1,3] Alex "Sandy" Pentland[1]

Impact of AI/ML - positive effects

Distilling raw data into actionable information

- scaling up annotations that humans provide
  - example: analyze satellite imagery in order to pinpoint deforestation



Cambodia: Satellite imagery from Google Earth show rubber plantations and associated deforestation ramping up over the past six years.

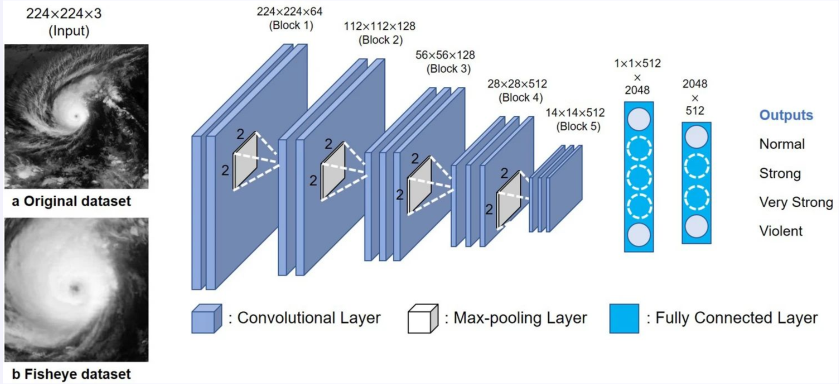
Impact of AI/ML - positive effects

Distilling raw data into actionable information

- scaling up annotations that humans provide
    - example: analyze satellite imagery in order to pinpoint deforestation

Improving forecasting

- provide forecasts of quantities such as wind power production, transportation demand, and extreme events by analyzing patterns in historical data



Domain knowledge integration into deep learning for typhoon intensity classification

Impact of AI/ML - positive effects

Distilling raw data into actionable information

- scaling up annotations that humans provide
  - example: analyze satellite imagery in order to pinpoint deforestation

Improving forecasting

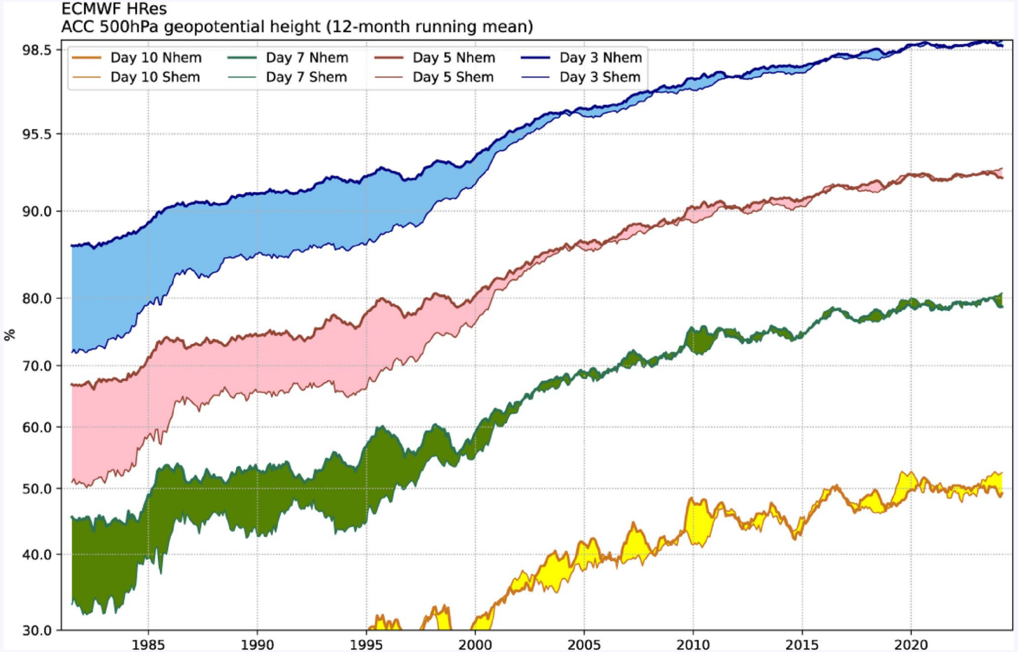
- provide forecasts of quantities such as wind power production, transportation demand, and extreme events by analyzing patterns in historical data

Optimizing complex systems

- optimizing for a specific objective given a complicated system with many variables
  - example: optimize freight transportation schedules

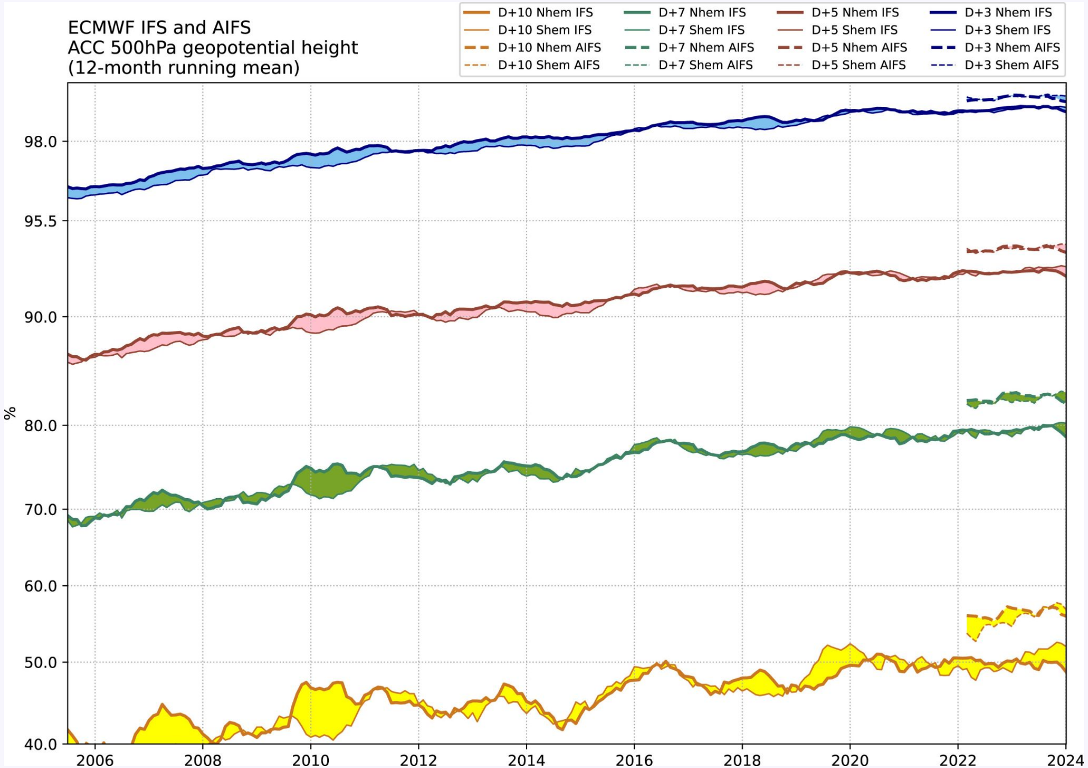
The quiet revolution of numerical weather forecasting

Forecast skill has improved ~ 1 forecast day per decade, partly due to advances in computational performance.



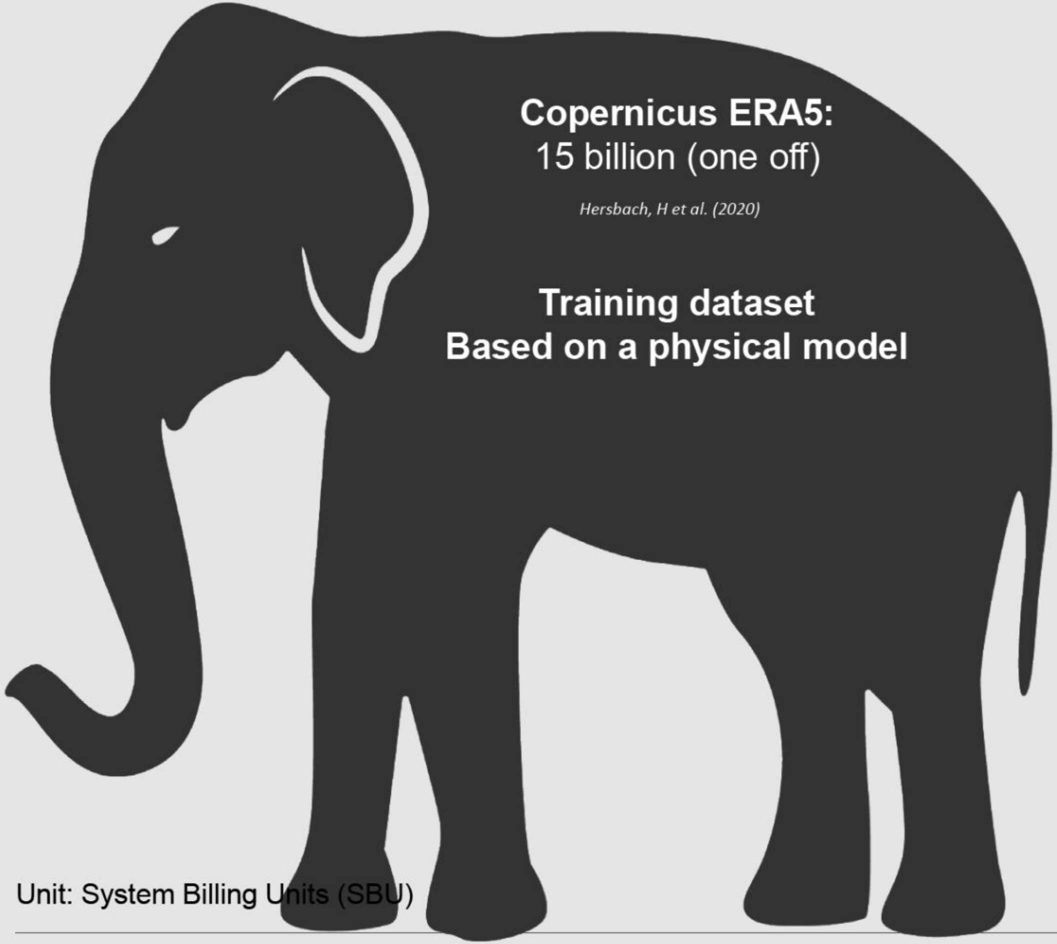
graph from: Göber, Martin, et al. "Enhancing the value of weather and climate services in society: identified gaps and needs as outcomes of the first WMO WWRP/SERA weather and society conference." *Bulletin of the American Meteorological Society* 104.3 (2023): E645-E651.

The ML revolution of numerical weather prediction



AI forecasting systems are years ahead in terms of forecast scores for deterministic and ensemble predictions.

Bauer, Peter. "What if? numerical weather prediction at the crossroads." *Journal of the European Meteorological Society* 1 (2024): 100002.



Copernicus ERA5:
15 billion (one off)

Hersbach, H et al. (2020)

Training dataset
Based on a physical model

Physics-based Forecast:
180 000
per forecast

AI Model:
0.3
per forecast

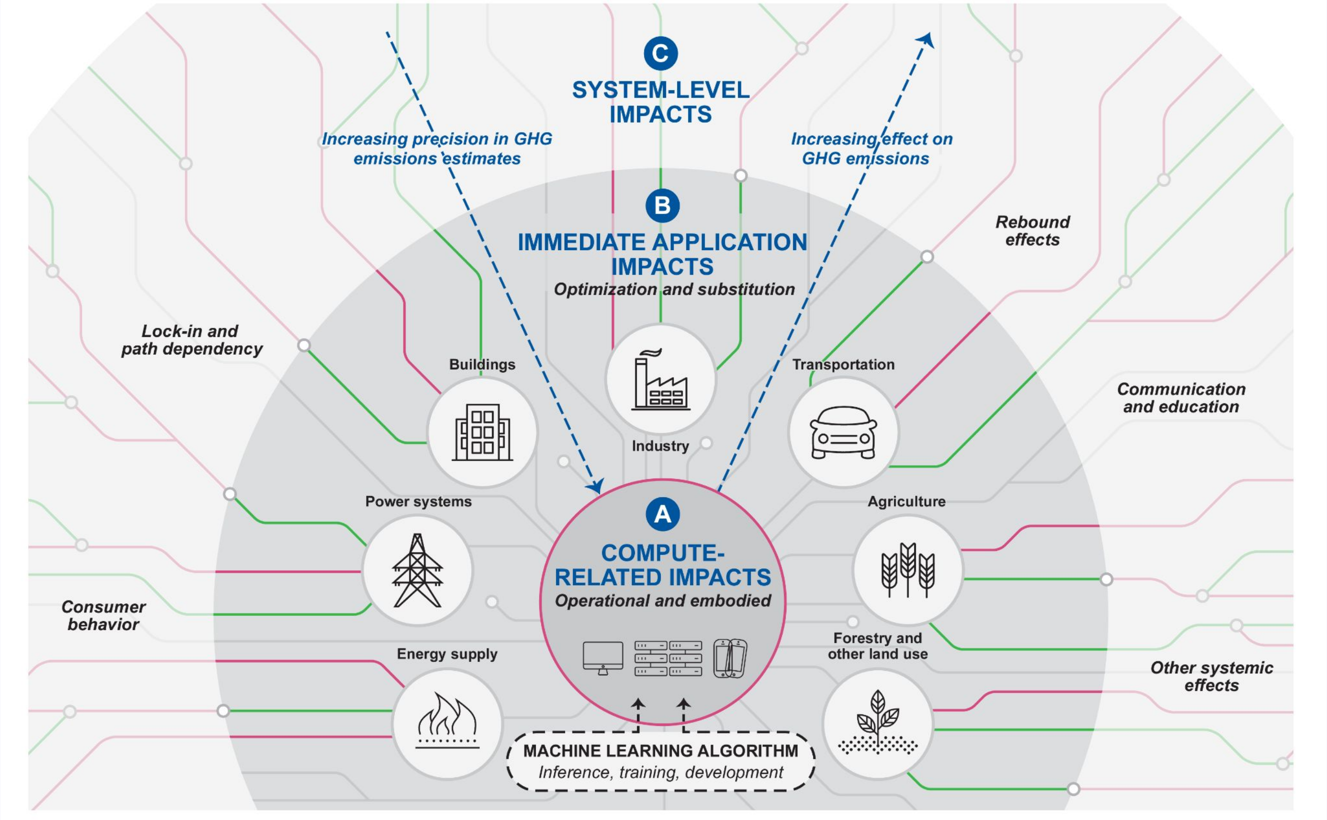
~1000x
reduction in
energy
Forecast time
reduced from
~30minutes to
~3minutes

Unit: System Billing Units (SBU)

Tackling Climate Change with Machine Learning

Paper

		Causal inference	Computer vision	Interpretable models	NLP	RL & Control	Time-series analysis	Transfer learning	Uncertainty quantification	Unsupervised learning
Mitigation	Electricity systems									
	Enabling low-carbon electricity		•	•		•	•		•	•
	Reducing current-system impacts		•				•		•	•
	Ensuring global impact		•					•		•
	Transportation									
	Reducing transport activity		•				•		•	•
	Improving vehicle efficiency		•			•				
	Alternative fuels & electrification					•				•
	Modal shift	•	•				•			
	Buildings and cities									
	Optimizing buildings	•				•	•			
	Urban planning		•				•			
	The future of cities				•					
	Industry									
	Optimizing supply chains		•			•	•			
	Improving materials									
	Production & energy		•	•		•				
	Farms & forests									
	Remote sensing of emissions		•							
	Precision agriculture		•			•	•			
	Monitoring peatlands		•							
	Managing forests		•			•	•			
	Carbon dioxide removal									
	Direct air capture									
	Sequestering CO_2		•							
Adaptation	Climate prediction									
	Uniting data, ML & climate science		•	•			•		•	
	Forecasting extreme events		•	•			•		•	
	Societal impacts									
	Ecology		•					•		
	Infrastructure					•	•		•	
	Social systems		•				•			•
	Crisis		•		•					
	Solar geoengineering									
	Understanding & improving aerosols						•		•	
	Engineering a control system					•			•	
	Modeling impacts						•		•	
Tools for Action	Individual action									
	Understanding personal footprint	•			•	•	•			
	Facilitating behavior change				•					•
	Collective decisions									
	Modeling social interactions			•		•				
	Informing policy	•	•		•				•	•
	Designing markets					•	•			•
	Education				•	•				
	Finance				•		•		•	


C
SYSTEM-LEVEL IMPACTS
Increasing precision in GHG emissions estimates
Increasing effect on GHG emissions
B
IMMEDIATE APPLICATION IMPACTS
Optimization and substitution
Rebound effects
Lock-in and path dependency
Buildings
Industry
Transportation
Communication and education
Power systems
Agriculture
A
COMPUTE-RELATED IMPACTS
Operational and embodied
Consumer behavior
Energy supply
Forestry and other land use
Other systemic effects
MACHINE LEARNING ALGORITHM
Inference, training, development