Towards Safe Machine Learning Lifecycles with ESG Model Cards

Thomas Bonnier and Benjamin Bosch

Société Générale

Introduction

- The term **Environmental, Social and Governance** (ESG) was coined in 2004 in a United Nations report where several financial institutions developed recommendations on ESG investment [35]. Its three pillars can be seen as measuring performance based on certain factors: (i) Environmental impacts: e.g., climate change and related risks. (ii) Social impacts: e.g., workplace health and safety. (iii) Corporate governance: e.g., accountability and transparency.

- We propose an approach which identifies the environmental, social, and transparency risks and suggest mitigating actions for each aspect of the ML modeling lifecycle. This approach aims to meet the following criteria:

– **Environmental pillar**: Efficient, Environmentally-friendly ML.

– **Social pillar**: Secure, Fair, Unbiased, Robust ML.

– **Governance pillar**: Transparency, Accountability, Auditability, Compliance throughout the ML lifecycle.

- Report these impacts in the ESG model card, along with the actions employed to reach that result. Our main contribution is thus to propose standardization in deploying safe ML by presenting a risk-based approach and a reporting tool considering the ESG impacts.

1

ESG risk identification and mitigation through the ML lifecycle

Data layer

- Fitted data size (E pillar):
  - data ingestion, storage and processing require power draw → carbon emissions
  - collecting data in excess

    → principles of proportionality and minimization
- Protected data area (S pillar):
  - uncontrolled number of external data sources (e.g., pretrained models), data quality

    → Know Your Data principle
- Transparent data flows (G pillar):
  - IP, personal data (GDPR)

    → principles of proportionality, minimization and Know Your Data

ML life cycle	Angle	E pillar	S pillar	G pillar
Data	Risk	Carbon emissions due to excessive storage, processing, and related infrastructure.	Using unchecked external data. Lack of transparency of pre-trained models. Embedded biases.	Inadequate personal and sensitive information retention. Lack of dataset transparency. Illegal collection of data.
	Mitigation	Proportionality rule based on use case. Data minimization. Reduced storage time.	Using reliable certified sources. Data exploration and pre-processing. Reweighing.	Data lineage. Documenting data limitations. Proportionality rule and data minimization.
	Limitation	Reduced availability of data resources for users.	Checking entire dataset is unachievable. Access to sensitive attributes for monitoring.	Detecting data bias in multidimensional settings is complex.

Model design layer

- Design rethinking (E pillar):
  - Structural cost: set of observations, feature space
  - Algorithmic cost: the model architecture, the learning algorithm and the hyperparameter optimization

    → principle of parsimony: reducing the hypothesis space (e.g., transfer learning) ,lightening the model structure (e.g., quantization), speeding up the optimization (e.g., cost-frugal optimization)
- Treatment for model's Achilles' heel (S pillar):
  - lack of representativeness in the modeling data
  - Sensitivity to adversarial attacks (white box, black box attacks)
- Scientific evidence (G pillar)
  - EDA
  - Local and global explainability methods (accuracy, fidelity, stability, sparsity, consistency)

Explaining the data with Prototypes

- Steps:
  - Input: **number of prototypes**
  - Objective: **minimize the discrepancy** between the distributions of the data and selected prototypes
  - Search strategy: find prototypes with simple **greedy search**

- Discrepancy measure: Squared Maximum Mean Discrepancy

$$MMD^2 = \frac{1}{m^2}\sum_{i,j=1}^{m} k(z_i, z_j) - \frac{2}{mn}\sum_{i,j=1}^{m,n} k(z_i, x_j) + \frac{1}{n^2}\sum_{i,j=1}^{n} k(x_i, x_j)$$

m prototypes z

kernel
'similarity' function
e.g.:RBF
$k(x, x') = \exp\left(-\gamma \| x - x' \|^2\right)$
Infinite feature space dimension

n original
data points x

normalized
feature
vectors

ML life cycle	Angle	E pillar	S pillar	G pillar
Design	Risk	Carbon cost due to feature engineering and model optimization, training, and inference.	Model bias in decision-making. Adversarial attacks.	Lack of transparency of model decision process.
	Mitigation	Model compression. Parsimonious feature selection. Cost-frugal optimization. Knowledge transfer.	Diagnostic tools: fault tree analysis, causal graph. Human oversight. Incremental learning. Differential privacy. In-processing techniques.	Model documentation. ESG model card. Auditing. XAI methods.
	Limitation	Bias amplification due to model compression. Lack of transparency of pre-trained models.	Utility vs. privacy. Disagreement between bias detection metrics. Fault tree in multidimensional settings.	Disparate quality of XAI methods across subgroups. Disagreement in XAI. Computational and storage cost of XAI.

Model implementation layer

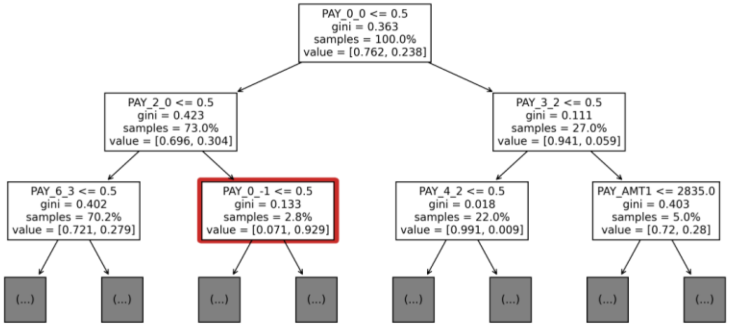
- Low-carbon code (E pillar):
  - Code with redundancies, inadequate data structure and algorithm choice might generate bottlenecks
  - Infrastructure dependencies
- Safe implementation (S pillar):
  - Unsecured model implementation strategy (e.g., treatment of missing values)
  - Third party package dependencies
- Reproducibility at every stage (G pillar):
  - ML pipelines complexity, lack of seed for random number generators

ML life cycle	Angle	E pillar	S pillar	G pillar
Implementation	Risk	Carbon footprint of inefficient coding practices and data centers.	Shadow APIs. Flaws in third-party packages. Data leakage.	Lack of pipeline reproducibility.
	Mitigation	Sharing benchmarks. Code profiling. Optimizing data center location.	Model review, certification, and inventory. Adversarial testing. Reporting security breaches. Checking CVE.	Code documentation. Specifying software and hardware characteristics. Randomness control.
	Limitation	Performance/latency trade-off.	Exhaustive list of edge cases. Hidden vulnerabilities.	External package dependency. Difficult to achieve reproducibility with certain libraries or online learning.

Model use and monitoring layer

- Ongoing monitoring (E pillar):
  - Definition of key performance indicators and related thresholds (eg, carbon footprint at inference)
- Vulnerability monitoring (S pillar):
  - Distribution shift monitoring (e.g., symmetrized KL Divergence)
  - Model uncertainty (e.g., Non-Conformity Analysis)
- Trust but verify (G pillar):
  - degree of decision automation (e.g., human in the loop, human on the loop)

- Uncertainty quantification (CP: LABEL method)
  - Conformity score:
    1. Fit classification model $\hat{p}_y$ to the training
    2. Compute the conformity score for the m data points of the calibration dataset (yi : true label): $s_i = 1 - \hat{p}_{yi}(x_i)$
    3. Compute q = (1-alpha)(m+1)/m quantile of $s_1, \ldots, s_m$, for target coverage 1-alpha (ex. 90%)
    4. Compute the prediction set for each x in Test: C(x) =$\{ y | s_i = 1 - \hat{p}_y(x) \leq q \}$
- Uncertainty explanation



Datasource: Credit (UCI)

ML life cycle	Angle	E pillar	S pillar	G pillar
Use & Monitoring	Risk	Deviation in the expected carbon footprint.	Deviation in bias detection metrics. Hidden vulnerabilities. Incidents. Increase in model uncertainty.	Algorithm aversion. Automation bias. No feedback from model users.
	Mitigation	Continuous monitoring: number of queries, average inference time, data size. Human-on-the-loop.	Human oversight. Monitoring bias detection metrics. Corrective actions. Checking new CVE. Reporting data breaches. Explaining model uncertainty.	Monitoring usage, user feedback, and rationale for model overrides. Training users on limitations. Human comprehensible explanations.
	Limitation	Complexity of carbon cost measurement in decentralized systems.	Disagreement between bias detection metrics. Patch deployment time frame.	Sparsity of explanations in multidimensional settings.

2

Model card

ESG model card

Model Card

- **Model Details**. Basic information about the model.
  - Person or organization developing model
  - Model date
  - Model version
  - Model type
  - Information about training algorithms, parameters, fairness constraints or other applied approaches, and features
  - Paper or other resource for more information
  - Citation details
  - License
  - Where to send questions or comments about the model
- **Intended Use**. Use cases that were envisioned during development.
  - Primary intended uses
  - Primary intended users
  - Out-of-scope use cases
- **Factors**. Factors could include demographic or phenotypic groups, environmental conditions, technical attributes, or others listed in Section 4.3.
  - Relevant factors
  - Evaluation factors
- **Metrics**. Metrics should be chosen to reflect potential real-world impacts of the model.
  - Model performance measures
  - Decision thresholds
  - Variation approaches
- **Evaluation Data**. Details on the dataset(s) used for the quantitative analyses in the card.
  - Datasets
  - Motivation
  - Preprocessing
- **Training Data**. May not be possible to provide in practice. When possible, this section should mirror Evaluation Data. If such detail is not possible, minimal allowable information should be provided here, such as details of the distribution over various factors in the training datasets.
- **Quantitative Analyses**
  - Unitary results
  - Intersectional results
- **Ethical Considerations**
- **Caveats and Recommendations**

Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B.,
Spitzer, E., Raji, I.D., Gebru, T.: Model cards for model reporting. In: Proceedings
of the Conference on Fairness, Accountability, and Transparency, FAT* 2019,
Atlanta, GA, USA, January 29-31, 2019. pp. 220–229. ACM (2019)

- Motivations behind the ESG model card:
  - reporting the ESG net impacts of the ML lifecycle, along with the actions used to reach that outcome
  - help to launch new initiatives and prompt model developers to build frugal, secure, and transparent ML systems

ESG model Card example: Image classification

ESG Model Card – Dog vs Cat Prediction

Model Details

- Developed by researchers at Anonymous Authors.
- CNN binary classifier : either a cat (class 0) or a dog (class 1) on a picture.
- Model owner: Anonymous. Model inventory code: DogvsCat_PRED. June 2022.

Intended Use

- Intended to be used by companies to identify whether there is a cat or a dog on a picture.
- Model exclusions: only works with pictures of cats or dogs.

Key Risk Metrics

- Application stakes: Low.
- Automation level: Medium.
- Adverse impact strength: Medium.

DATA

E Pillar

- *Dog vs cat* dataset size: 2 classes, 16,001 training images, 8,997 validation images.

Storage Scenario	
Training size	515 MB
Validation size	293 MB

S Pillar

- External data sources: Cats and Dogs dataset from Microsoft Research (also on https://www.kaggle.com/c/dogs-vs-cats) , 'imagenet' weights of the pretrained ResNet-152 neural network.
- Protected attribute: none.
- Group representation bias: none, (50% cats and 50% dogs).

G Pillar

- Personal data: no personal information.
- Data owner: Data dpt.
- Data preprocessing: downsizing of the pictures to150x150 pixels, ResNet preprocessing.
- Data augmentation: described in the notebook.

DESIGN

E Pillar

- Input : picture of size 150 x 150 with 3 channels.
- Data augmentation (rotating, width shifting, height shifting, shearing, zooming, horizontal flipping).
- Architecture based on a ResNet-152 (frozen 'imagenet' weights), followed by a Dense layer (1024 units, 'relu' activation) and a Dense layer (1 unit, 'sigmoid' activation), total params: 110,801,793.
- Loss: binary cross-entropy; metric: accuracy ; optimizer: Adam.
- Training: 1600 images per epoch.
- Processor/GPU/Allocated Memory: CPU 2.4GHz/GPU 16GB/32GB.

	Actual	With unfrozen ResNet-152 weights*
Trainable params	52,430,849	110,650,369
Validation accuracy after 5 epochs	96.1%	52.3%
Validation accuracy after 15 epochs	98.3%	Not computed
Modeling carbon emission	0.0004 g	0.0003 g*
Inference carbon emission (for 160 examples)	7.22e-06 g	Not computed

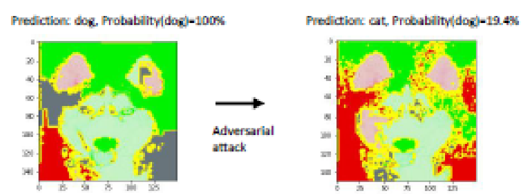
**training was stopped after 5 epochs for the model with unfrozen ResNet-152 weights (He et al. 2016)*

S Pillar

- Adversarial attack testing using Fast Gradient Signed Method (FGSM) :

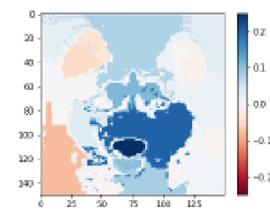
Epsilon (FGSM's perturbation factor)	0.1	0.2	0.5	0.7	1
Adversarial examples with wrong predictions (%)	6	9	21	26	35

- High sensitivity of model prediction to adversarial noise in the image background (Prediction contributions: green for dog and red for cat):

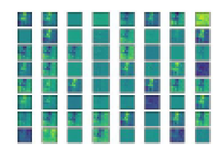


G Pillar

- XAI with Local Interpretable Model-agnostic Explanations (LIME):
  - Size of the neighborhood to learn the linear model: 1000 examples;
  - Heatmap : blue corresponds to a positive contribution and red to a negative contribution to "dog" class.
- Examples of relevant extracted features:
  - Cat: shape of ears;
  - Dog: shape of muzzle.



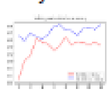
- Analysis of feature maps on different layers:



IMPLEMENT

E Pillar

- Model implementation with Tensorflow (Apache License 2.0).
- Transfer learning strategy to speed up the training.
- Emission tracking with *codecarbon* package (MIT License).
- Execution time at inference (100 examples) : 3.19s.



S Pillar

- Adversarial attacks based on Tensorflow's implementation of FGSM.
- Reliable external packages: Common Vulnerabilities and Exposures checked on MITRE, June 2022. Python 3.8.13, *codecarbon* 2.1.1, *numpy* 1.19.5, *tensorflow* 2.4.1, *lime* 0.2.0.1.

G Pillar

- Explainability with *lime* 0.2.0.1.
- Code owner: SW Engineering dpt.
- Pipeline included in the notebook.

USE & MONITOR

E Pillar

- Monitoring metrics: accuracy, energy mix evolution (g CO2/kWh).
- Metrics refresh rate: monthly.
- Greenhouse gas emission reduction target: at least 55% by 2030 (European Commission Target Plan).

S Pillar

- Monitoring metrics: accuracy, recall, precision.
- Model threshold: 50%.

G Pillar

- Monitoring metrics: XAI stability.
- Model user-level XAI: local explanation with LIME (BSD 2-Clause "Simplified" License).
- Model user training frequency: once a year.
- Model use mode: Human-on-the-loop.

Conclusion

- We presented a risk-based approach to standardize safe ML deployment.
- Several practical principles have been suggested: proportionality, parsimony or continuity.
- ESG model card for fairly reporting the model impacts and remediations across the ML lifecycle.
- Next steps: ESG MLOps tool to scale ESG principles