INTERNATIONAL TELECOMMUNICATION UNION

TELECOMMUNICATION STANDARDIZATION SECTOR

STUDY PERIOD 2017-2020

FG-AI4H-I-035

ITU-T Focus Group on AI for Health

Original: English

WG(s):	Plenary	E-meeting, 7-8 May 2020

DOCUMENT

Source:	Editors
Title:	DEL7.3: Data and artificial intelligence assessment methods (DAISAM) reference
Purpose:	Discussion

Contact: Luis Oala
Fraunhofer Institut, HHI
Germany

Email: luis.oala@hhi.fraunhofer.de

Abstract: This document, *Data and artificial intelligence assessment methods (DAISAM) reference*, is the reference collection of WG-DAISAM for assessment methods of data and artificial intelligence quality evaluation. This document also constitutes subsection 7.3 of the FG-AI4H deliverable 7.

Keywords

Artificial Intelligence for Health, Data, Quality Assessment, Measures, Metrics, Evaluation

Change Log

This document contains Version 1 of the ITU-T Technical Paper on "*Data and Artificial Intelligence Assessment Methods (DAISAM) Reference*" approved at the ITU-T Focus Group Artificial Intelligence for Health (FG-AI4H) meeting held virtually, 07-08 May 2020.

Authors (in alphabetical order)

Contact:	Balachandran,Pradeep TechnicalConsultant India	Email: pbn.tvm@gmail.com
Contact:	Cabitza, Federico Università degli Studi di Milano-Bicocca Italy	Email: federico.cabitza@unimib.it
Contact:	Calderon Ramirez, Saul Costa Rica Institute of Technology/ De Montfort University Costa Rica/ UK	Email: sacalderon@itcr.ac.cr
Contact:	Chiavegatto Filho, Alexandre University of São Paulo Brazil	Email: alexdiasporto@usp.br
Contact:	Eitel, Fabian Charité - Universitätsmedizin Berlin Germany	Email: fabian.eitel@charite.de
Contact:	Extermann, Jérôme HEPIA Switzerland	E-mail: missing
Contact:	Fehr, Jana Digital Health Center, Hasso-Plattner-Institute Germany	Email: jana.fehr@hpi.de
Contact:	Ghozzi, Stephane Robert Koch Institut Germany	Email: ghozzis@rki.de
Contact:	Gilli, Luca ClearBox AI Italy	Email: luca@clearbox.ai
Contact:	Jaramillo-Gutierrez, Giovanna Milan and Associates SPRL Belgium	Email: gjgutierrez@protonmail.com
Contact:	Kester, Quist-Aphetsi CRITAC | Ghana Technology University College Ghana	Email: kesterphysics@yahoo.co.uk
Contact:	Kurapati, Shalini Clearbox AI Solutions Italy	Email: shalini@clearbox.ai
Contact:	Konigorski, Stefan Digital Health Center, Hasso-Plattner-Institute Germany	E-mail: missing

Contact:	Krois, Joachim Charité - Universitätsmedizin Berlin Germany	Email: joachim.krois@charite.de
Contact:	Lippert, Christoph Digital Health Center, Hasso-Plattner-Institute Germany	E-mail: missing
Contact:	Martin, Jörg Physikalisch-Technische Bundesanstalt Germany	Email: joerg.martin@ptb.de
Contact:	Merola, Alberto AICURA medical GmbH Germany	Email: alberto.merola@aicura-medical.com
Contact:	Murchison, Andrew University of Oxford UK	Email: agmurchison@gmail.com
Contact:	Niehaus, Sebastian AICURA medical GmbH Germany	Email: sebastian.niehaus@aicura-medical.com
Contact:	Oala, Luis Fraunhofer Institut, HHI Germany	Email: luis.oala@hhi.fraunhofer.de
Contact:	Ritter, Kerstin Charité - Universitätsmedizin Berlin Germany	Email: kerstin.ritter@charite.de
Contact:	Samek, Wojciech Fraunhofer HHI Germany	Email: wojciech.samek@hhi.fraunhofer.de
Contact:	Sanguinetti, Bruno Dotphoton AG Switzerland	Email: bruno.sanguinetti@dotphoton.com
Contact:	Schwerk, Anne Technical Consultant Germany	Email: schwerkler@googlemail.com
Contact:	Srinivasan, Vignesh Fraunhofer HHI Germany	Email: vignesh.srinivasan@hhi.fraunhofer.de

CONTENTS

List of Tables

List of Figures

FG-AI4H Deliverable 7.3

Data and artificial intelligence assessment methods (DAISAM) reference

Summary

This document, *Data and artificial intelligence assessment methods (DAISAM) reference*, is the reference collection of WG-DAISAM for assessment methods of data and artificial intelligence quality evaluation. This document also constitutes subsection 7.3 of the FG-AI4H deliverable 7.

1 Scope

TBD

2 References

Please refer to bibliography lists in the individual sections.

3 Terms and definitions

3.1 Terms defined elsewhere

This Technical Paper uses the following terms defined elsewhere:

3.1.1 term [reference]: TBD

3.2 Terms defined here

This Technical Paper defines the following terms:

3.2.1 term [reference]: TBD

4 Abbreviations

TBD	TBD

5 Data Quality – Measures, Metrics, and Methods

5.1 Bias and Fairness

Authors (alphabetically): Balachandran, Pradeep; Fehr, Jana; Jaramillo-Gutierrez Giovanna; Konigorski, Stefan; Lippert, Christoph; Murchison, Andrew; Niehaus, Sebastian

Introduction and problem definition

AI-algorithms supporting clinical diagnostic, prognostic and triage and prevention decision-making may unintentionally learn hidden features such as clinical context variables to improve prediction performance (Badgeley et al. 2019). Learning hidden features (also called confounders) can result in algorithmic bias which may yield unreliable predictions when the algorithm is applied on an external testset sourced from a population with different distributions of hidden features. Algorithmic bias can further be present when an algorithm that was trained with data that only represents a subset of the real-world data it is intended for. This can potentially lead to prediction results that are harmful for people and unintended by the model creators (Chen et al. 2019). For example, AI systems could perpetuate racial bias since the biases already exist in historical data. This may reflect differences in biological vulnerability to disease as well as differences in social resources. Identifying algorithmic bias is a non-trivial task that requires domain expertise about the targeted use-case scenario as well as expertise about methods to identify and mitigate algorithmic bias. Not only in healthcare but also in other AI application fields, it is of critical importance to identify learnt hidden features, especially sensitive social factors, to assure fairness, avoiding discrimination and unreliable predictions (Holstein et al. 2019). The use of machine learning algorithms for clinical decision-making should focus on demonstrating clinically important improvement in patient outcomes rather than solely performance metrics such as area under the curve and accuracy. It is critically important to ensure that all genders, ethnicities, age groups are adequately represented, if the AI-based product is then applied to a wide range of patients. Statistical accuracy does not necessarily equal clinical accuracy. To address these challenges, tech practitioners need to address the limitations in the machine learning algorithms and ensure quality control of their application in various clinical settings and patient population and document and state their limitations.

Aim

Here we provide a summary of how to understand and identify algorithmic bias at different stages of the AI-based product that may have critical implications when the algorithm is applied in a real-world clinical setting. The aim is to train the most accurate model for each group without harming any minority group of patients. Furthermore, methods to mitigate bias according to the problem at hand are provided. These guidelines aim to provide a framework for technologists that build health related AI based products to investigate the presence of algorithmic bias.

Bias definition

Bias can be considered a systematic deviation in a result compared to the true estimate. This has the potential to arise in AI models if the training database is substantially different to the target population (defined by the intended use), and may arise in the assessment of the model accuracy if the test database is inadequate. As a result, the algorithms may not offer benefit to for example people whose data are not represented in the data set.

Potential sources of bias

Knowledge of the intended context (domain expertise) and uses of a model should inform the identification of bias sources. Potential sources of bias in healthcare algorithms can arise at the pre-processing stage (data collection, data preparation) and post-processing stage (model deployment and evaluation).

Pre-processing stage:

- Representation bias: If the targeted user group and the targeted patient group differ between the data used for the development process and the data is intended for.
- Learning hospital-specific features: Departments in hospitals often have a well-defined area of responsibility and rarely treat cases outside this focus. Different departments also often use different medical devices. Learning features such as hospital-specific parameters can bias predictions.
- A ‘case-control’ database design could over-inflate measures of accuracy- i.e. if the data from a group of patients known to have a given condition is combined with the data from a group who do not, cases where there may be uncertainty and in which the model may not perform as well are excluded. The test database should ideally comprise a non-selected group of individuals reflecting the intended use population as closely as possible.
- Measurement bias: If variables were measured with different methods of different accuracies. I.e. a positive result of a disease is more likely to be truly positive when it was measured with test A, rather than test B.
- Label bias: Annotation and label bias can arise when data was labelled by different practitioners with different levels of experience
- Assigning ground truth- if the ground truth (reference standard) in the test set is established by raters who have knowledge of the outcome of the AI model in the test group, this could inflate measures of accuracy.
- ‘Over-curation’ of the data- e.g. if poor-quality MRI scans are excluded, measures of accuracy may not reflect the real-world application where noise or artefacts may be common. Similarly, if cases with missing data are excluded from the population, the accuracy of the model in the real-world setting may be lower than in the test setting.
- Issues related to data integrity & data quality: Improper procedures on data inclusion and exclusion, input and output variable selection, pre-processing methods (data encoding-decoding formats, data compression and encryption, outlier and missing value treatment).
- Lack of standardized protocols and tools for data reproducibility (Who, When, Where, How, etc.), lack of interoperable data interfaces to collect and integrate diverse data types

Post-processing stage:

- Historical bias: An algorithm might be biased by social factors especially when training data was collected through services, surveys, or social media that are predominantly used by a certain social group (defined by ethnicity, religion, gender, …).
- Representation bias: An underrepresentation of minority or marginalized social groups in the training data can lead to unreliable predictions on underrepresented social groups. In this case, algorithmic fairness is not guaranteed.
- Algorithmic tuning: When business heuristics are applied to model outputs e.g. differential tuning of performance parameters in order to optimize for chosen business logic (e.g. differential diagnosis based on age, gender, ethnicity, etc.)
- Aggregation bias: arises during model building. If there are two or more distinct populations that are inappropriately combined. In that case, the population of interest is heterogeneous and a single model is unlikely to suit all minority groups.

- Evaluation bias: occurs during model iteration and evaluation. It can arise when the testing or external benchmark populations do not equally represent the various parts of the population it is applied on. Evaluation bias can also arise from the use of performance metrics that are not appropriate for the way in which the model will be used.
- Deployment bias: occurs after model deployment, when a system is used or interpreted in inappropriate ways.

Bias detection and mitigation methods

The following sections list qualitative and quantitative methods that can be used for detecting bias in AI algorithms.

Qualitative detection methods

Directed acyclic graphs (DAGs)

Causal diagrams are one strategy to systematically identify hidden features (such as demographic or hospital variables) that may be indirectly learnt by the algorithm to make predictions. The diagram incorporates a directed acyclic graph (DAG) to visualize the interaction and dependence of variables, where variables are depicted as nodes and the direction of their influence (from cause to effect) is depicted with arrows (Pearl, 1995). Such DAGs can be either constructed based on a priori expert knowledge, or also learned from the data using causal discovery algorithms (Peters, et al. 2017). When a DAG is drawn to assess the risk of bias of an algorithm, the following variable categories and their dependencies on the respective disease and algorithmic output should be considered: Patient (age, gender, ethnicity), disease (early onset, late-stage, mild, severe), hospital (department, practitioner, device model) (motivated by Badgeley et al., 2019). This list only serves as a starting point and can be extended.

Matched case-control test datasets

One approach to assess the presence of bias is to create a testset where one or more potential bias variables (that were for example identified with a DAG) are equally distributed between the case and control group (matched case-control). A risk of bias is present if the prediction power decreases with increased controlling of biasing variables. Badgeley et al. (2019) used this approach to determine whether the algorithm detected fractures from learned clinically meaningful image features or from indirectly learned confounding variables.

QUADAS-2 and PROBAST: Tools for risk of bias assessment

QUADAS-2 (Whiting et al., 2011) and PROBAST (Moons et al. 2019) are two example tools that assess the risk of bias by systematically reviewing prediction models. QUADAS-2 uses signalling questions for a systematic bias assessment in diagnostic accuracy studies. PROBAST is a tool to assess the risk of bias of diagnostic and prognostic prediction models, and their applicability for the intended population and context using reviewing questions. In the last step of the systematic review, PROBAST guides the user to make an overall judgement about the risk of bias in the prediction model as ‘low concern’, ‘high concern’ or ‘unclear’.

These tools may require modification to suit the particular requirements of assessing bias in the setting of healthcare AI, but provide a good framework for thinking about how bias may affect measures of accuracy. It is also worth noting that these approaches generally rely on judgement to assess the risk of bias in certain categories, rather than applying specific metrics.

Quantitative detection methods

DeLong test

One approach to assess the risk of bias is to control for potential confounders in the testset. This can be done by creating a testset that is stratified by the biasing variable (i.e. gender). Subsequently, the algorithmic performance, i.e. measured in sensitivity, specificity and others, can be compared between these strata. For example, Oakden-Rayner, 2020 stratified the performance of a pneumonia classifier between chest x-rays with and without a chest drain. The unpaired DeLong statistical test (DeLong et al., 1988) can be applied to test if the area under the receiver operating curves (AUC) of the strata are the same. A rejection of this null hypothesis indicates bias.

Unsupervised k-means clustering

Unsupervised clustering can be used to detect hidden stratification between subclasses with different algorithmic performance. As a concrete example, Oakden-Rayner et al. (2019) applied k-means clustering on the test dataset's pre-softmax feature vector using k ∈ {2, 3, 4, 5}. For each k, a high and a low error cluster was identified with the largest distance, measured by the Euclidean distance of their centroids. It can then be investigated whether underlying parameters are causing the performance differences between those clusters. One challenge of this approach can be that it is not always possible to achieve a meaningful separation of clusters.

Regression

Regression models can be used to quantify the effect of biasing variables on the algorithmic outcome. For a continuous or dichotomous or algorithmic output linear and logistic regression modelling can be applied respectively, using potential biasing variables (i.e. age and gender) as input. As additional covariate, the gold standard test to verify the disease is added in order to control the "true" part of the variance that confounders might cover (equation (1)). The parameter estimates of □ serve as a first assessment to quantify the influence of the biasing variable on the algorithmic output. Then, hypothesis tests of the regression coefficients of the confounders can be used to evaluate their association with the score.

$$Algorithmic\ output = \square 0 + \square 1 * age + \square 2 * gender + \square 3 * disease + \epsilon \qquad (1)$$

One challenge in the interpretation of regression models is when two covariates are highly correlated and if it is of interest to separate the effects of the predictors. This challenge can be tackled by removing highly correlated variables, computing principal components of the predictors in a first step and using these in a second step in the analysis, or by using approaches as implemented in the 'FairML' package (Adebayo, 2016).

FairML

'FairML' is a python toolbox that comprises four methods to quantify the dependence of input variables on algorithmic predictions. (Adebayo, 2016):

- Random forest
- minimum redundancy, maximum relevance feature selection (mRMR)
- Least absolute shrinkage and selection operator (LASSO)
- Iterative orthogonal feature projection (IOFP)

Linear dependencies between input variables are addressed by the IOFP and LASSO algorithms, while non-linear relationships are addressed by Random Forest and mRMR.

Phi (Φ)-Representativeness metric

Phi (Φ)-Representativeness (Cabitza et al.) measures the similarity between two datasets. This metric can be used to understand to which extent the sampled testset represents the reference dataset (i.e. the real-world clinical dataset). Low values of Pi-representative indicate selection bias. It is an alternative way of comparing data set distributions equality tests like Kolmogorov-Smirnov test.

H-accuracy

H-accuracy (Ha) is an alternative to the regular accuracy (Cabitza & Campagner, 2019). It overcomes biased accuracy measures, also called the accuracy paradox of highly imbalanced datasets (Valverde-Albacete & Peláez-Moreno, 2014). H-accuracy accounts for the importance of rightly detecting one class at the expense of the others, the chance of obtaining the correct prediction by chance as well as the complexity of cases correctly identified. By considering these factors, H-accuracy is considered safe and helps to curb the bias factor of model drift towards 'over diagnosis’. The other advantage of H-accuracy is that it can be tailored to a specific diagnostic task by tuning its parameters to make it more suitable to the preferences of the domain experts. The parameter configuration can be made local (e.g. hospital setting) or for a specialist community, scientific society or association and hence this measure can be considered as a parametric version of accuracy (Cabitza & Campagner, 2019).

GANs

Conditional generative adversarial networks (GANs) allow to learn and generate class-wise data representations. These representations can be used, for example, to test a classification model and not only to identify a bias, but also to characterize it. For this purpose, different representations with different characteristics are generated for each class. If certain characteristics reduce the probability of classification, this characteristic is explored in various combinations and intensities using a bayesian optimization. This enables to identify sets in the data, where the classifier is not working and to describe them exactly.

Bias error mitigation methods

Federated learning

To prevent a data bias and thus also a model bias, the training data set should contain a broad range of characteristics, which is difficult in the healthcare, because often data sets from different institutions cannot be merged. Federated Learning is a method in machine learning that allows the training of models on decentralized data pools. The setup includes several local nodes and a global node, whereby gradients are calculated locally on the local nodes and these are combined into a global model. Thus, no data sets have to be merged for medical applications. This allows the acquisition and use of training data sets, which usually could not be shared due to data privacy concerns. Larger and more comprehensive data sets can be used to train models that, for example, have no geographical bias.

Re-sampling technique (k-fold cross validation)

K-fold cross validation is a gold standard re-sampling technique used to estimate the model accuracy on unseen data. This method looks at splitting the dataset into *k* parts (e.g. *k=10*). Out of *k* parts, the model is trained on *k-1* parts and tested on the one part maintained separately. This procedure is repeated for each part maintained as test data. The result is a more reliable and accurate estimate of the model. In k-fold cross-validation the value of *k* is a design consideration. In error estimation, low values of *k* results in high bias and low variance and high values of k results in low bias and high variance. Typical values of *k* are 3, 5, 10, etc.

ConceptNet method

Using the principle of 'ConceptNet', which is a knowledge base of word meanings, domain-specific semantic models are produced from smaller sets of “trusted data” than deep learning normally requires. In the ConceptNet based bias mitigation method, models learn from general-domain background knowledge base i.e. (by bootstrapping with previously created, general-domain data points) in conjunction with domain-specific data. Through this models tend to induce less bias compared to that learned from the training corpus alone e.g. this involves starting off deep learning models with millions of “common sense” facts, instead of starting from nothing, which can offset the bias otherwise introduced by a domain-specific training corpus. This method is also faster than the method of “trusted humans” writing rules to maintain the system.

Commented [TSB1]: Please confirm, i.e. [=that is] or e.g. [=for example].

Adversarial training

Usually, adversarial training addresses the robustness of models, but biases can also be reduced. Similar to bias detection and characterisation, conditional generative adversarial networks (GANs) can be used for this purpose. However, in this case further model updates are trained on the additional generated representations of the respective class. Thus, adversarial training specifically addresses the ability of the predictor to predict underrepresented classes more accurately and consequently to eliminate class-wise bias (Zhang, 2018).

Treating imbalanced datasets

A standard technique used to address the problem of imbalanced datasets is that of balancing the skewed classes (e.g. when one class is over-represented in the data set) in the training data. Here the objective is to attain approximately equal number of data samples for both the minority and majority classes by balancing their class frequencies. This can be achieved by re-sampling techniques namely Synthetic Minority oversampling technique (SMOTE) which attempts to balance the data set by creating synthetic data samples. Another robust strategy to deal with imbalanced datasets is that of ensemble techniques, where performance of single classifiers is improved by constructing several two stage classifiers from the original data and then combining their predictions. State-of-the-art ensemble techniques include Adaptive Boosting, Gradient Boosting techniques, etc.

AI Fairness 360 - a bias detection and mitigation toolkit

AI Fairness 360 is a comprehensive open-source toolkit of metrics to check for unwanted bias in datasets and machine learning models, and algorithms to mitigate such bias throughout the AI application lifecycle. It contains over 30 fairness metrics and 9 algorithms that aim to deal with bias. It enables practitioners to incorporate the most appropriate tool for their problem into their work products (Bellamy, 2018).

The 9 algorithms are:

1. Optimized Pre-processing
2. Disparate Impact Remover
3. Equalized Odds Postprocessing
4. Reweighting
5. Reject Option Classification
6. Prejudice Remover Regularizer

7. Calibrated Equalized Odds Postprocessing
8. Learning Fair Representations
9. Adversarial Debiasing

References

Commented [TSB2]: All references should be consolidated in a references section of the document. Normally, clause 2 (as per the deliverables template). Also note the style is "[label]".

Adebayo, Julius A. "FairML : ToolBox for Diagnosing Bias in Predictive Modeling." Thesis, Massachusetts Institute of Technology, 2016. https://dspace.mit.edu/handle/1721.1/108212.

Badgeley, Marcus A., John R. Zech, Luke Oakden-Rayner, Benjamin S. Glicksberg, Manway Liu, William Gale, Michael V. McConnell, Bethany Percha, Thomas M. Snyder, and Joel T. Dudley. "Deep Learning Predicts Hip Fracture Using Confounding Patient and Healthcare Variables." *Npj Digital Medicine* 2, no. 1 (April 30, 2019): 1–10. https://doi.org/10.1038/s41746-019-0105-1.

Bellamy, R. K. E., Mojsilovic, A., Nagar, S., Ramamurthy, K. N., Richards, J., Saha, D., … Mehta, S. (2019). AI Fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM Journal of Research and Development*, *63*(4/5).

https://doi.org/10.7326/M18-1377.

Cabitza, Federico, and Andrea Campagner. "Who Wants Accurate Models? Arguing for a Different Metrics to Take Classification Models Seriously." *ArXiv:1910.09246 [Cs, Stat]*, October 22, 2019. http://arxiv.org/abs/1910.09246.

Cabitza, Federico, Andrea Campagner, and L Sconfienza. "As If Sand Were Stone. New Concepts and Metrics to Probe the Ground on Which to Build Trustable AI. (Submitted). Submitted to BMC Medical Informatics and Decision Making. Cabitza, F., Campagner, A., & Sconfienza, L.." *Submitted to BMC Medical Informatics and Decision Making*.

Chen, Irene, Fredrik D. Johansson, and David Sontag. "Why Is My Classifier Discriminatory?" *ArXiv:1805.12002 [Cs, Stat]*, December 10, 2018. http://arxiv.org/abs/1805.12002.

Hajian, S., Bonchi, F., y Castillo, C. (2016). Algorithmic bias: From discrimination discovery to fairness-aware data mining, *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*: 2125-2126.

Holstein, K., Wortman Vaughan, J., Daumé III, H., Dudik, M., & Wallach, H. (2019, April). Improving fairness in machine learning systems: What do industry practitioners need?. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (p. 600). ACM.

Moons, Karel G.M., Robert F. Wolff, Richard D. Riley, Penny F. Whiting, Marie Westwood, Gary S. Collins, Johannes B. Reitsma, Jos Kleijnen, and Sue Mallett. "PROBAST: A Tool to Assess Risk of Bias and Applicability of Prediction Model Studies: Explanation and Elaboration." *Annals of Internal Medicine* 170, no. 1 (January 1, 2019): W1. https://doi.org/10.7326/M18-1377.

Oakden-Rayner, Luke. "Exploring Large-Scale Public Medical Image Datasets." *Academic Radiology* 27, no. 1 (January 2020): 106–12. https://doi.org/10.1016/j.acra.2019.10.006.

Pearl, Judea. "Causal Diagrams for Empirical Research." *Biometrika* 82, no. 4 (1995): 669–88. https://doi.org/10.2307/2337329.

Peters, Jonas, Janzing, Dominik, and Schoelkopf Bernhard. *Elements of Causal Inference*. The MIT Press. Accessed March 31, 2020. https://mitpress.mit.edu/books/elements-causal-inference.

Valverde-Albacete, Francisco J., and Carmen Peláez-Moreno. "100% Classification Accuracy Considered Harmful: The Normalized Information Transfer Factor Explains the Accuracy Paradox." *PloS One* 9, no. 1 (2014): e84217. https://doi.org/10.1371/journal.pone.0084217.

Whiting, Penny F. “QUADAS-2: A Revised Tool for the Quality Assessment of Diagnostic Accuracy Studies.” *Annals of Internal Medicine* 155, no. 8 (October 18, 2011): 529. https://doi.org/10.7326/0003-4819-155-8-201110180-00009.

Zhang, Brian Hu, Blake Lemoine, and Margaret Mitchell. "Mitigating Unwanted Biases with Adversarial Learning." arXiv:1801.07593 [cs.LG]. https://arxiv.org/abs/1801.07593.

5.2 Perturbations

Authors: Bruno Sanguinetti, Jérôme Extermann, Enrico

Reviewers: Tobias, Andrew, Quist-Aphetsi Kester

Summary

The reliability of AI depends on the quality of training and inferencing data, as well as on the statistical consistency of those two datasets. Here we give recommendations on managing data quality so that AI systems are reliable at scale. From the physical sample to final analysis result (e.g. diagnostic) data undergoes a number of transformations, accumulating “perturbations”. We first describe the typical data acquisition and processing pipeline, splitting it into two logical parts: (a) pre-archival processing, where perturbations should be minimised to preserve the statistical and metrological properties of the data, and (b) post-archival processing where stronger perturbations may be acceptable as soon as these are applied consistently to each data element of a dataset before being fed to an AI system.

We also discuss how including metrological data relating to the expected uncertainty and statistics with respect to “ground truth” enables normalization across different data sources, data augmentation and uncertainty propagation through monte-carlo methods. Finally, we discuss how the importance of data authentication to verify that no undocumented perturbations have occurred, malicious or not.

Commented [TSB3]: ??? What does this line mean?

Scaling AI systems from a research environment to worldwide clinical use presents several challenges with respect to data quality. AI reliability depends on the quality of training and inferencing data, as well as on the statistical consistency of those two sets of data.

Quality and consistency depend on the entire data acquisition storage and processing pipelines. A number of actors are involved in developing, validating and maintaining each stage of this pipeline. These stages and activities must be coordinated according to clear interfaces to ensure the reliability of AI systems. These aspects become increasingly important as AI systems move from the lab to worldwide clinical use, i.e. from an environment where high-end equipment and end-to-end expertise are present, to an environment where it is important to select the most affordable components that guarantee accuracy and where expertise is clinical.

In this section we give an overview of a typical data acquisition and processing pipeline, then we discuss the overall logic allowing to manage the “distortions” that may occur in this pipeline. We then give measures that help evaluating the impact of these distortions, as well as specific examples.

Data acquisition and processing pipeline

For the AI-developer to achieve reliable and repeatable results, across a number of systems, it is important that he is aware of the full data acquisition and processing pipeline:

1. A physical sample is prepared (e.g. histopathology slide)
2. The physical sample is digitized by the acquisition instrument (e.g. slide scanner). This instrument typically lightly pre-processes the digital data, and the output of this instrument is called “raw data”. The target of this pre-processing is typically to correct errors in the data, or

to compute a relevant physical property. This pre-processing should be simple, generic, and well documented. Raw data is archived for future use.

3. Raw data is then retrieved to be used by a post-processing algorithm, typically this is a complex pipeline, including a large number of steps. An example is: de-bayering, denoising, geometric corrections, normalization, stitching, de-convolution, segmentation.

4. The post-processed data is then the input to an AI algorithm, for training or inference.

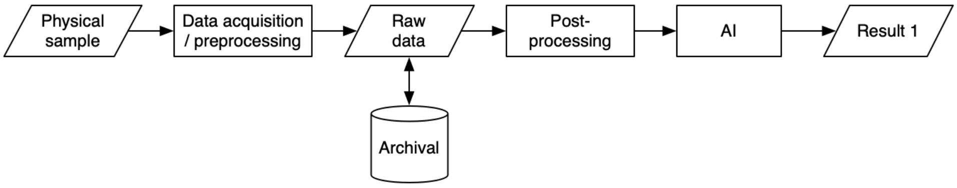


Figure x – Caption missing

Distortion

Distortion is any process that may irreversibly alter, statistically speaking, the final results of post-processing. Distortions are accumulated in the data acquisition and processing pipeline.

Effect of distortions in machine learning

Commented [TSB4]: This should be a Heading 4 style, I suppose (as well as a couple of the next headings).

Different distortions, or different distortion parameters, may add degrees of freedom to the sample space associated with the measurement, and affect their probability space. Training data should span this space, and therefore the amount of training data will grow with the number of possible post-processing paths.

The figure below illustrates the effect of having a single data processing pipeline, vs multiple data processing pipelines. In the upper illustration, the data is always affected by the same distortions, resulting in a small sample space, which can therefore be densely sampled by training data and is expected to yield repeatable and reliable results.

In the lower illustration, data may undergo different types of distortions, with different parameters and potentially in a different sequence, resulting in a much larger sample space, which will be more sparsely sampled and yield less reliable results.

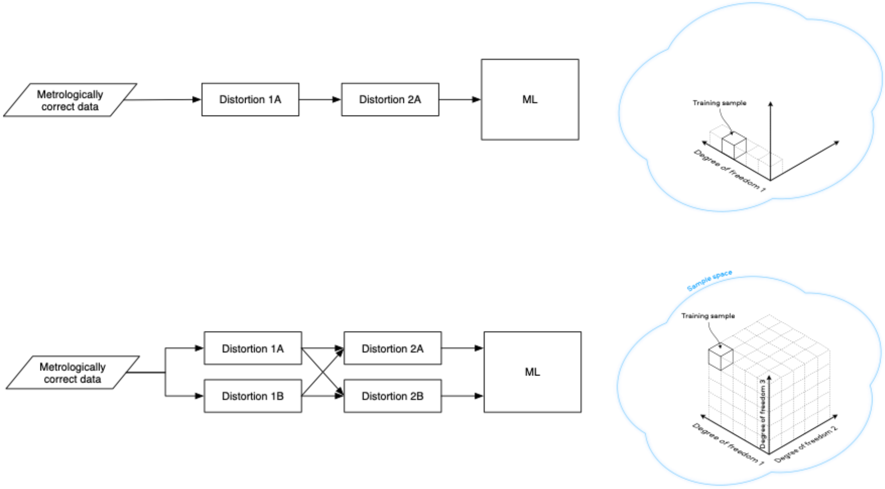


Figure x – Caption missing

Deep data processing pipelines are common, as is distortion. Processing pipelines, and associated distortions are susceptible to evolve across different systems and versions of the same system. It is therefore recommended that the sample data is kept in a metrologically correct format, and that data is processed through a single post-processing pipeline before being used for Machine Learning applications. In situations where this is not possible, each of the different pipelines should be represented in the training data.

Managing distortions during data acquisition and pre-processing.

Data acquisition and pre-processing targets the generation of metrologically correct raw data. This means that data should represent the “ground truth”, i.e. the measured physical property within the specified uncertainty, and according to a specified statistical model.

It is important that only simple, explainable, reproducible and quantifiable distortions are introduced in raw data.

Simple distortions may be divided in: random errors and systematic errors. Any distortion happening on the raw data should specify how these two types of errors are affected.

Measuring distortions using statistics with a phantom or synthetic sample

These distortions can be measured using standard metrological techniques, i.e. by analysing the statistics of the measurement of a known, stable synthetic sample (or phantom).

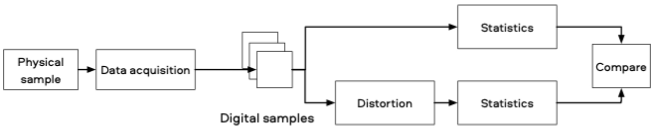


Figure x – Caption missing

Random errors can be specified in terms of Signal-to-Noise ratio, where (assuming a linear system) the signal is the mean of the measured values, minus their offset, and noise is the standard deviation. The signal-to-noise ratio is the ratio of those two quantities.

A distortion may decrease the Signal-to-Noise ratio, and this decrease may be specified in dB.

The below figure gives an example, from image compression, where a reduction is SNR is measured for each potential pixel value.

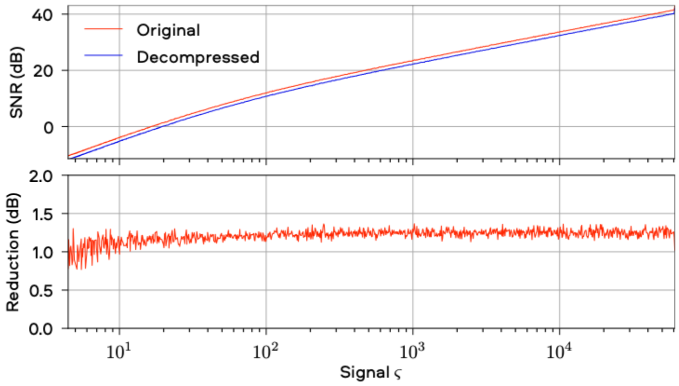


Figure x – Caption missing

Systematic errors may be specified in terms of Bias, which gives a measure of how an average value change (e.g. mean or median) with the distortion.

The below figure gives an example (from image compression) showing how for each potential pixel value, the bias has been measured.

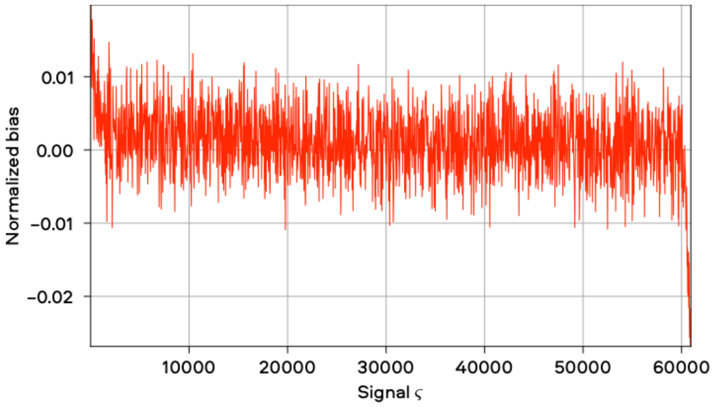


Figure x – Caption missing

Measuring distortions in the pre-analysis processing

Distortions in the pre-analysis processing can be measured via a statistical method relying on the comparison of the processing outcomes after distortion with respect to the statistical dispersion provided by raw data. Outcomes variability from raw data can be obtained by feeding the AI algorithm with synthetic data, obtained by simulating the raw statistical distribution.

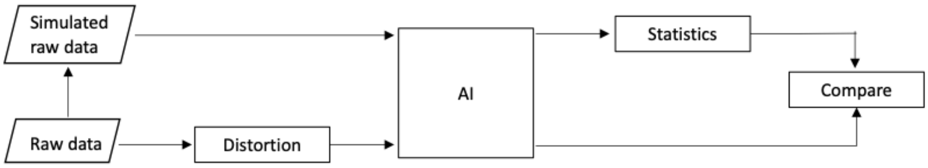


Figure x – Caption missing

This method has been applied to a trainable cell segmentation task. The area of single segmented cells in a raw image with and without compression is measured. Synthetic raw images, obtained by simulating noise in the image after accurate calibration of the acquiring camera, are used to find the uncertainty of the values obtained from the uncompressed data. The dispersion of the results obtained with compression, estimated over all segmented objects, can be compared to that obtained from synthetic raw data. This measure can be used to validate a specific distortion in the pre-analysis processing. [ref to our paper]

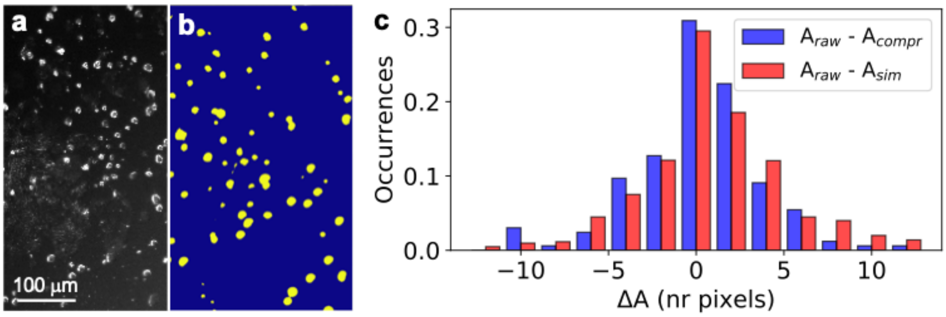


Figure x – Caption missing

Figure … a, b) Phase-contrast micrograph of cells and segmentation mask. c) Distribution of the single-cell area difference obtained with raw and compressed image (blue), as well as with raw and simulated images (red).

Normalization

As systems scale, data from different sources (e.g. instruments from different manufacturers, or different versions of the same instrument, different sample preparations etc.) will be used in the context of the same AI model. Each of these systems will have applied to the data different perturbations. However, if data is available in a raw format, and its metrological properties are known, it may be normalized, and validated before being fed to post-processing and AI. As an example if a microscopy sample was taken with an effective pixel size of 500nm and a point-spread function of 750nm and another with a pixel size of 400nm, and a point-spread function of 700nm, they may be resampled to have the same effective pixel size and point-spread function, increasing the reliability of AI results.

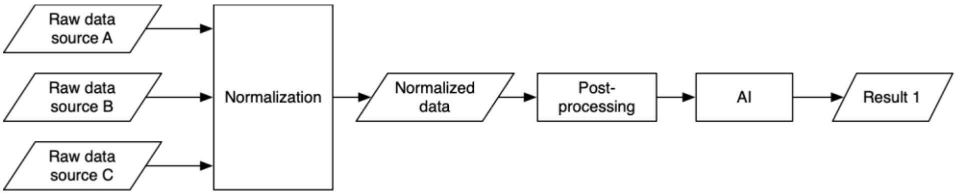


Figure x – Caption missing

Data authentication

Perturbations may occur at any time, intentionally, unintentionally or maliciously (e.g. through a cyberattack). Therefore, before being used in AI systems, data must be authenticated. This may be achieved by the equipment manufacturer providing a mechanism to validate such data, such as a cryptographically signed checksum of the data.

References

[1] Mirsky, Yisroel, Tom Mahler, Ilan Shelef, and Yuval Elovici. “CT-GAN: Malicious Tampering of 3D Medical Imagery Using Deep Learning.” *ArXiv:1901.03597 [Cs]*, June 6, 2019. http://arxiv.org/abs/1901.03597.

[2] Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. “Explaining and Harnessing Adversarial Examples.” *ArXiv:1412.6572 [Cs, Stat]*, March 20, 2015. http://arxiv.org/abs/1412.6572.

[3] Kurakin, Alexey, Ian Goodfellow, and Samy Bengio. “Adversarial Examples in the Physical World.” *ArXiv:1607.02533 [Cs, Stat]*, February 10, 2017. http://arxiv.org/abs/1607.02533.

5.3 Summary statistics for data quality

Author: Kherif, Ferath

5.4 Data set splitting

Authors: Stephane Ghozzi, Joachim Krois, Alberto Merola

Why data splitting?

Evaluating a model entails gauging its generalisability: After it was trained and implemented, how should one expect it to perform on new data? To mimic this situation, the evaluation should not happen on data seen for learning the internal model parameters: The data set has to be split. Specifically, one otherwise risks *overfitting* the model on the data seen, following them too closely and reproducing meaningless noise.

One usually considers three independent sets:

- the *training set* for fitting internal parameters;
- the *validation set* for finding the best settings (hyperparameters as well as which features are included);
- the *test set* for the final evaluation.

As described below and in the next section, crucial properties of the set-up in which the model is applied as well as the quantity of data available should be considered for how to split. With those constraints, data points are then attributed at random to each set, thus reflecting (apparently random) factors that one cannot account for or shouldn’t play a role.

In particular, each set should be a *sample* representative of the whole *population*, meaning all values or ranges coming up in the whole data set should also be present in each set. For example, in a binary classification task, each of both classes should be present in at least a few exemplars in each set, otherwise evaluation scores cannot be computed! This is true also for the features. Imbalanced data sets, where some classes are rare, thus can prove difficult for the evaluation (as well as for the learning itself).

Lastly, especially on smaller data sets, the evaluation scores will depend to some extent on the particular sampling applied to generate the different sets. Thus, as always when working with random variables, one should generate many different realizations of set splitting and consider the *statistics* of evaluation scores. This is crucial for sensibly comparing different models: Is one *significantly* better than the others? How much and how reliably so?

See e.g. [1,2] for more details.

Note that after models have been evaluated, the best one(s) should be trained on all available data before being put in production!

Splitting strategies: Time series

For data taking the form of time series, time implicitly plays a crucial role, meaning the fact that the new data come chronologically after available data is important. In that case, the (validation and) test set(s) have to include data that succeed the training set. This means that the (validation and) test set(s) are necessarily contiguous in time. In cross-validation and to gauge the stability of the evaluation, the size of the training set will vary with each splitting: Smaller for earlier test sets, larger for the later ones. See e.g. section 3.1.2.5.1. of [2].

Bibliography

[1] Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning. Springer-Verlag*. Pages 219-260. Retrieved from https://web.stanford.edu/~hastie/ElemStatLearn/

[2] https://scikit-learn.org/stable/modules/cross_validation.html

[3] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830. Retrieved from http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html

Cross validation

Cross-validation (or CV) is an approach for model validation used to assess how the results of a statistical analysis will generalize to an independent unknown data set, that is how the model will perform in practice. Cross-validation combines prediction precision measurements to derive a more accurate estimate of the model’s predictive performance (Grossman et al., 2010).

Cross-validation is typically an iterative process, in which at each iteration a sample of data is divided into two complementary subsets: one used for the model fitting (the training set), and the other for validating the analysis (the validation or testing set). For a more robust estimate of the model's predictive performance, multiple iterations are performed using different data set partitions, and the results are combined (for example averaged).

Every model training process involves fitting the parameters or weights of the model to the training data set as well as possible. If the trained model is then used to predict an independent sample of validation data, it will generally perform worse. While a difference in performance is to be expected, high discrepancies are symptomatic of problems in the training of the model and might be due, among others, to the amount of data available. In particular, discrepancies in performance are likely to be large when the size of the training data set is small, either in absolute terms, or relatively to the number of parameters in the model. The latter case typically results in overfitting (Tetko et al., 1995). Cross-validation helps to estimate the size of this effect.

Cross-validation has been shown to be a nearly unbiased estimator of the model performance (Christensen, 2015). On the other hand, due to the large variance of the estimates, if two models are compared based on the results of cross-validation, the one with the better estimated performance may not actually be the best performing.

Cross-validation methods can be categorized into two groups: exhaustive and non-exhaustive. While the former methods iterate on all possible ways to divide the original sample into training and validation sets, the latter do not. The most exemplary cross validation method is an exhaustive one: the leave-one-out (LOO) cross-validation.

Leave-one-out cross-validation

Leave-one-out cross-validation (LOO CV) involves using 1 data sample as the validation set and the remaining observations (N-1, where N is the data set size) as the training set. This is repeated on all permutations of the original data set with a validation set of 1 sample and a training set of N-1 (Celisse, 2014).

LOO CV is a specific case of leave-p-out cross-validation (LpO CV), where at each iteration sets of p and N-p samples are considered. Notably, LpO CV with p=2 has been shown to be an accurate method for estimating the area under ROC curve of binary classifiers (Airola et al., 2011).

Class Imbalance

A dataset is said to show class imbalance when its observations present a disproportionate distribution among the classes (or categories) that constitute the given dataset. In other words, some classes are overrepresented compared to others. The former are commonly referred to as majority classes, while the second as minority classes. This might be a problem in classification tasks, because most machine learning algorithms assume that data is equally distributed among classes. In the presence of class imbalance this condition is violated and the classification tends to be biased towards the majority class.

Ideally class imbalance in the dataset is to be avoided, although this might not be possible, either because there is no control over the data acquisition or because of the very nature of the data distribution. Notably, many data types in the physics and social sciences domains are characterised by a naturally imbalanced data distribution and follow a power law probability distribution (aka Zipf's law, see Wikipedia: https://en.wikipedia.org/wiki/Zipf%27s_law).

Multiple strategies are available to tackle class imbalance and improve classification performance and depend, among others, on the specific predictive task and on the specific metric used to evaluate the algorithm. Some of them act at the algorithmic level and therefore won't be touched upon here.

The strategies that focus on the data set and its splitting approaches can be grouped in 3 different categories (Branco et al, 2015):

1. Up-sampling of minority classes: randomly duplicating observations from the minority classes, in order to balance the distribution. This comes with the caveat that no new information is introduced in the dataset.
2. Down-sampling of majority classes: removing random observations from majority classes. This might be problematic in case of data scarcity, although .
3. Generating synthetic samples: new synthetic samples are generated from the original data sample. Typical approaches include creating new samples based on the distances between the point and its nearest neighbours or based on the distances for the minority samples near the decision boundary, either in a fully automated or semi-automated fashion. Similarly, trained AI models like generative adversarial networks (or GANs, (Goodfellow et al., 2014)) can be used to generate data sets with a balance class distribution.

Bibliography

Airola, A.; Pahikkala, T.; Waegeman, W.; De Baets, Bernard; Salakoski, T. (2011-04-01). "An experimental comparison of cross-validation techniques for estimating the area under the ROC curve". Computational Statistics & Data Analysis. 55 (4): 1828–1844. doi:10.1016/j.csda.2010.11.018

Branco, P., Torgo, L., Ribeiro, R. P. (2015). A Survey of Predictive Modelling under Imbalanced Distributions. *arXiv:1505.01658v2.*

Celisse, A. (2014). "Optimal cross-validation in density estimation with the L^2-loss". The Annals of Statistics. 42 (5): 1879–1910. arXiv:0811.0802. doi:10.1214/14-AOS1240. ISSN 0090-5364.

Christensen, Ronald (May 21, 2015). "Thoughts on prediction and cross-validation". Department of Mathematics and Statistics University of New Mexico. Retrieved May 31, 2017

Goodfellow, Ian; Pouget-Abadie, Jean; Mirza, Mehdi; Xu, Bing; Warde-Farley, David; Ozair, Sherjil; Courville, Aaron; Bengio, Yoshua (2014). Generative Adversarial Networks. *Proceedings of the International Conference on Neural Information Processing Systems,* pp. 2672–2680.

Grossman, Robert; Seni, Giovanni; Elder, John; Agarwal, Nitin; Liu, Huan (2010). "Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions". Synthesis Lectures on Data Mining and Knowledge Discovery. Morgan & Claypool. 2: 1–126.

Tetko, I. V.; Livingstone, D. J.; Luik, A. I. (1995). "Neural network studies. 1. Comparison of Overfitting and Overtraining" . Journal of Chemical Information and Modeling. 35 (5): 826–833

Timnit Gebru, T., Morgenstern, J., Vecchione, B., Wortman Vaughan, J., Wallach, H., Daumé, H. and Crawford, K. (2020). Datasheets for Datasets. *arXiv*:1803.09010v5

6 AI solution quality – Measures, metrics, and methods

6.1 Explainability

Authors (alphabetically): Fabian Eitel, Luca Gilli, Shalini Kurapati, Kerstin Ritter, Anne Schwerk

Explainable AI Considerations

Explainable AI (XAI) is as crucial as AI models themselves, as XAI enables not only trust but also control, which is crucial if we lack a visible model. Broadly speaking, XAI fulfills four primary needs (see Figure X):

1) XAI enables the necessary **control** for counteracting bias, such as measurement errors or wrongly labeled data. This deep error and model understanding allows us to detect possible weaknesses and thereby speed up training and development times, as mistakes are readily detected and improved .

2) XAI allows for **liability**, which is crucial in establishing accountability and a necessary prerequisite for GPs, which have to execute those systems and hence need to understand the outputs and its consequences – including possible errors.

3) **Fairness** is also crucially enabled by XAI, by providing insights into the models, the data, and its learned predictions and thus to prevent discrimination towards subgroups, such as females or ethnic subgroups. Unfortunately, most algorithms are purely optimized towards efficiency and are completely neglecting fairness

4) Finally, the most important aspect for scientists, model insights and mechanistic actions can also only be discovered when those are transparent, which then allows **generalization** and **transfer** to be established – thereby improving replicability and allowing for certain degrees of causality.

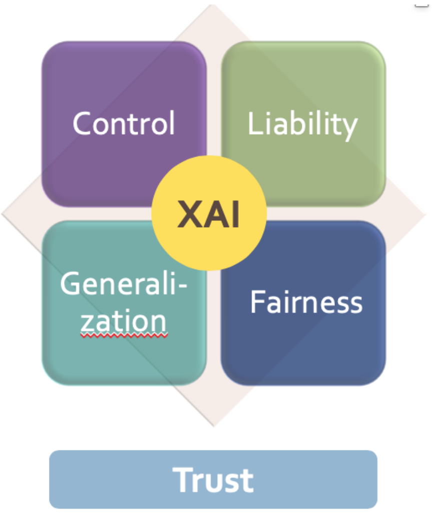


Figure X – Caption missing

Challenges in different transparent AI Pipeline stages

Given that AI is produced over a multi-step pipeline, where each step can be prone to possible bias and errors, one should consider the entire development process for the later to be deployed models. In the following and also in Figure XI, the different steps of the pipeline are considered to ensure transparency and retraceability.

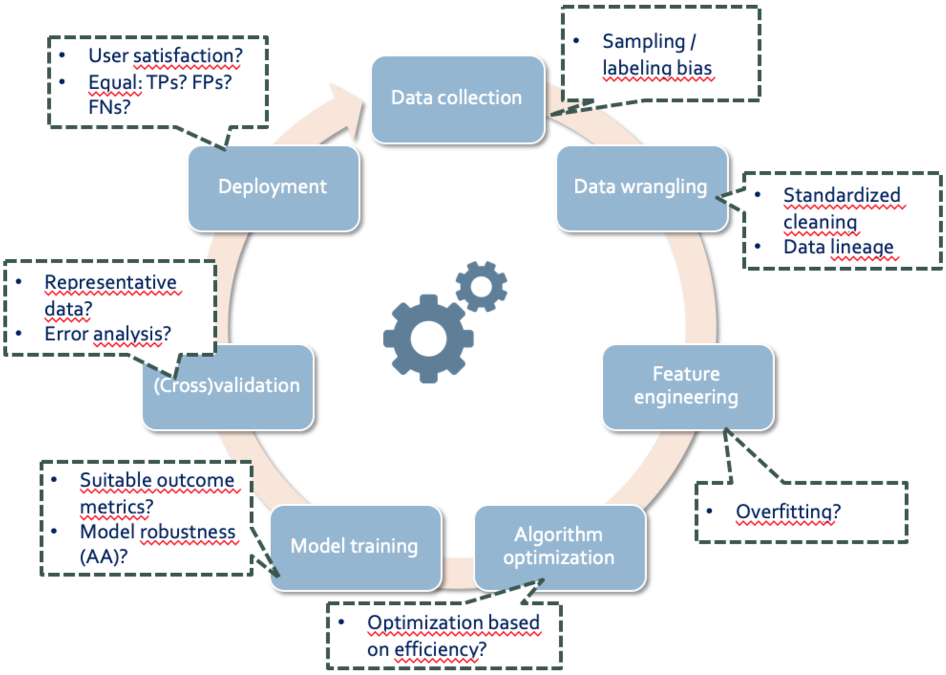


Figure XI – Caption missing

1. Data collection

The data collection process can cause many bias sources, and as AI will extrapolate these, the control of correctly sampled data is crucial. A very common bias during data collection is the so-called sampling bias, which is due to non-representative data samples. This phenomenon is currently seen during the COVID-19 analyses, which are based on data that is not representative, due to limited testing of solely symptomatic patients. Thus data samples need to reflect true populations and the actual diversity and distribution in order to allow for proper and generalized deployment. Another common bias, is the labelling bias, which is due to biased data or biased labels and can be circumvented in some scenarios by including intra/inter rater reliability measurements that identify errors within one labelor and between two independent labellors respectively.

2. Data Wrangling

As data scientists spend most of their time pre-processing data and only 20% in the actual modelling phase, those pre-processing procedures should be very transparent and traceable to allow for control and replicability. For example, when addressing missing values, those may be missing for protected groups in a non-random fashion which makes accurate predictions hard to render. Also, it is not clear often, which data can be considered true outliers or how to handle true outliers. For example, mean imputations can strongly bias towards the mean. Thus, a way to ensure standardized data cleaning would be highly needed, similarly all transformations should be automatically captured to ensure data lineage.

Another crucial pre-processing step is the process of feature engineering, which allows to isolate key information, highlight patterns, and bring in domain expertise. Yet this step is prone to subjective decisions, including the inclusion of redundant features which can then cause overfitting or the deselection of correlated features, which can be subjective and introduce bias. Therefore, a good way would be to also include internal or external cross-validation of the feature engineering and selection process and to take the sum, difference, product, or quotient of multiple features.

3. Model training and optimization

The absence of human oversight and involvement during training of DL models should be avoided or controlled by post-hoc oversight and insights, as there are many possible error sources and most algorithms are purely optimized towards efficiency and accuracy as opposed to human needs, e.g. weighting for the most fastest treatment and not one that is tailored to quality of life of the patient.

Yet, there are handy ways to control ethical and fair decision making of the models. One useful implementation to ensure control during model training can be *regularization*, which consists of adding a penalty to the different parameters of the model to reduce the freedom of the model. Hence, the model will be less likely to fit the noise of the training data and also less likely to be overfitting.

Another necessary control assumption that should be met is *demographic parity*, which implies that the classifier should make positive predictions on a protected population group at the same rate as the entire population. Similarly, the assumption of *equal opportunity* implies that a classifier should have equal true positive rates and also False-positive/negative rates on a protected population as those of the entire population. In addition, algorithm augmentation (e.g. Lagrangian approach) can incorporate fairness into the training algorithm itself, by penalizing the impact of biased samples, e.g. a mathematical technique called Lagrange multipliers uses fairness constraints (e.g. handicapped people should be hired at the same rate as non-handicapped people) to influence the loss in the training algorithm. Even though this is challenging to implement, as it adds considerable complexity to the training process, those technologies can have significant meaning for self-learning systems.

In certain scenarios, a randomization of algorithm optimization is needed, as these allow that the algorithms randomly change between different rules, to prevent the domination of certain rules.

4. Biased outcome metrics

Biased outcome parameters are often used in medical analyses, as this sector is highly complex and affords lots of domain knowledge to ensure proper analytics. For example, the variable of healthcare cost can disguise unequal access of different populations, when used for approximating healthcare needs, as less budget is spent on certain subgroups of patients who have the same level of need.

Though AUC and AP are proven to be effective metrics for measuring the performances in imbalanced datasets, a high score does not necessarily mean that a model makes a clear distinction between patients.

5. Cross-Validation

One of the very crucial steps, as this ensures real-world applicability, yet this is often not addressed, as actual ‘test’ data sets are mostly only available when the systems are deployed.

6. Deployment

When the AI solution is finally deployed, the system should entirely be re-evaluated, as the context and new data can adjust many previously met assumptions, such as equal true positives or false positives or false negatives.

Need for Explainable AI (XAI) Standardization

1. No ground truth for Post-hoc XAI
2. No overlapping Post-hoc XAI methods
3. Unconsidered AI Pipeline
4. Unaddressed need for user-centred XAI (adaptive, interactive)

Degrees of Good AI Explanation Systems

It is not very clear to define a good explanation. Basically, a good explanation implies that is is understood by the user, which implies a complex interaction between: (1) the offered explanation, (2) the receiver's knowledge and beliefs, (3) the context of the situation, and (4) the receiver’s goals. Thus to ensure human-level understanding is already a complex process, and it has to consider different end-user contexts, and hence, in order to consider all of these aspects, precise experiments with different users would be required, such as the patients, their relatives, clinciscians, scientists, and developers.

Nonetheless, some general features can be defined, that are crucial for ensuring understanding:

- Causal relationships and counterfactual faithfulness are easy to grasp concept, as are explanations based on examples and comparisons
- It is useful to show limitations of predictions, in order to not disappoint and create mistrust
- Humans can best understand low dimensional spaces and logical arguments that are clear, precise and complete

As people differ in their satisfaction threshold, with some being satisfied with simple or superficial explanations that reference fewer causes, and others need very granular explanations, ideal explanations are interactive and user-centered.

Evaluation Criteria for Explainable AI (XAI)

1. Confidence measures to training examples
2. Empirical evidence
3. Theoretical guarantees
4. Standards such as IEEE -P7001-Transparency of Autonomous Systems
5. Robustness, Reliability measures, generalizability
6. Consistency of XAI models
7. Enforce experimentation to ensure validity
8. Ensure human-level understanding

Explainable AI Interpretability Classification

- Local XAI Vs Global XAI
- Local XAI : Ability to understand individual decisions for a particular case or feature
- Global XAI : Tries to explain-How does the model reason?

Methods of Posthoc Explainable AI

a. Input perturbation LIME, SHAP method, occlusion, etc

b. Signal method-input method based on activated neurons (Activated Max.)

c. Proxy mechanism: simplifying ANNs, e.g. DeepRed

Challenges of Posthoc Explainable AI

a. Interaction effects

b. Computationally intensive

c. Slow

d. Lack of overlap between methods

e. Lack of Ground Truth

Methods of Antehoc Explainable AI

a. Verbal decision path

b. Heuristic input attribution (Saabas)

Challenges of Antehoc Explainable AI

a. Insufficient for multiple trees

b. Tree-depth bias for feature relevance

c. Slow and Sampling variability

Measures of Explanation Effectiveness for the DARPA XAI Psychology Explainable AI Program

1. User Satisfaction

   a. clarity of the user explanation (user rating)

   b. utility of the user explanation (user rating)

2. Mental Model

   a. Understanding individual decisions

   b. Understanding overall model

c. Strength /Weakness assessment

d. 'What will it do' prediction

e. ‘How do I intervene' prediction

3. Task Performance

a. Does the explanation improve user's decision,task performance

b. Artificial decision tasks introduced to diagnose the user's understanding

4. Truth Assessment

a. Appropriate future use and trust

5. Correctability

a. Identifying errors

b. Correcting errors, continuous training

References

Arrieta, A.B., D'iaz-Rodr'iguez, N., Ser, J.D., Bennetot, A., Tabik, S., Barbado, A., Garc'ia, S., Gil-L'opez, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2019). Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI. ArXiv, abs/1910.10045.

Zweig et al. (2019). Algorithmische Transparenz. Analysen und Argumente. Digitale Gesellschaft. Konrad Adenauer Stiftung. Nr. 338.

Lapuschkin, S., Wäldchen, S., Binder, A. et al. Unmasking Clever Hans predictors and assessing what machines really learn. Nat Commun 10, 1096 (2019). https://doi.org/10.1038/s41467-019-08987-4

Spinner, T., Schlegel, U., Schäfer, H., & El-Assady, M. (2019). explAIner: A Visual Analytics Framework for Interactive and Explainable Machine Learning. https://doi.org/10.1109/TVCG.2019.2934629

Mueller, S. T., Hoffman, R. R., Clancey, W., Emrey, A., & Klein, G. (2019). Explanation in Human-AI Systems: A Literature Meta-Review, Synopsis of Key Ideas and Publications, and Bibliography for Explainable AI. Retrieved from http://arxiv.org/abs/1902.01876

Mueller, S. T., Hoffman, R. R., Clancey, W., Emrey, A., & Klein, G. (2019). Explanation in Human-AI Systems: A Literature Meta-Review, Synopsis of Key Ideas and Publications, and Bibliography for Explainable AI. Retrieved from http://arxiv.org/abs/1902.01876

Zicari CSIG, 2019; World Economic Forum: white paper

Bornstein, 2016.; Bologna & Hayashi, 2017

Holzinger, A, Langs, G, Denk, H, Zatloukal, K, Müller, H. Causability and explainability of artificial intelligence in medicine. WIREs Data Mining Knowl Discov. 2019; 9:e1312. https://doi.org/10.1002/widm.1312

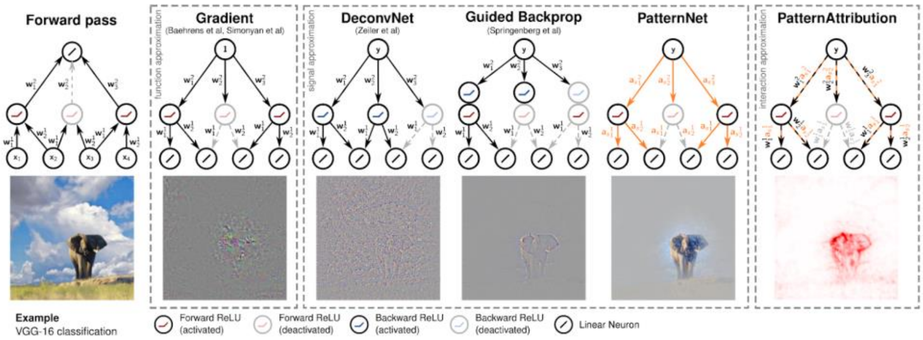
Gilpin et al. (2019). Explaining Explanations: An Overview of Interpretability of Machine Learning. arXiv:1806.00069

Zachary C. Lipton. 2016. The Mythos of Model Interpretability. arXiv:1606.03490

An Overview of Attribution Methods

1. Saliency analysis (Gradient)
    - Backpropagating the network's gradient into the input space. I.e. taking the partial derivative with respect to the input, rather than the weights as in backpropagation updates.
2. Gradient*Input
    - Multiplying the output of the saliency analysis with the original input. Allows to scale the attribution to the features, removing the effects of different feature scales.
3. DeconvNet
    - Adaption of saliency analysis which checks for activation of the model's ReLU nodes on the backpropagation rather than the forward pass.
4. Guided Backprop
    - Adaption of saliency analysis which checks for activation of both the forward and backward passes. Only backprogates when both are active.
5. Layer-wise Relevance Propagation (LRP)
    - Based on taylor decomposition, uses the score rather than the gradient, conserves total relevance in between layers, multiplies with the activation at each layer.
6. PatternNet & PatternAttribution
    - Adaptations of LRP, finding a better suitable reference point than LRP which uses the origin.
7. ...

2. Quantification methods:
    1. We have used *regional quantification* based on brain atlases to judge the quality of attribution methods. Brain MRIs are strongly registered and atlases are available in the common MNI space. By computing the attribution per brain region one can determine if the model finds clinically relevant regions to be of importance or whether artifacts influence the decision. See Böhle et. al and Eitel et. al.
3. Others:
    1. Words of caution see Adebayo et. al and Kindermans (2019).



Taken from Kindermans et. al 2017. Take the author's preference into consideration.

References

Kindermans, P. J., Schütt, K. T., Alber, M., Müller, K. R., Erhan, D., Kim, B., & Dähne, S. (2017). Learning how to explain neural networks: Patternnet and patternattribution. *arXiv preprint arXiv:1705.05598.*

Shrikumar, A., Greenside, P., & Kundaje, A. (2017, August). Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (pp. 3145-3153). JMLR. org.

Simonyan, K., Vedaldi, A., & Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In *European conference on computer vision* (pp. 818-833). Springer, Cham.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K. R., & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, *10*(7).

Böhle, M., Eitel, F., Weygandt, M., & Ritter, K. (2019). Layer-wise relevance propagation for explaining deep neural network decisions in mri-based alzheimer's disease classification. *Frontiers in aging neuroscience*, *11*, 194.

Eitel, F., Soehler, E., Bellmann-Strobl, J., Brandt, A. U., Ruprecht, K., Giess, R. M., ... & Scheel, M. (2019). Uncovering convolutional neural network decisions for diagnosing multiple sclerosis on conventional MRI using layer-wise relevance propagation. *NeuroImage: Clinical*, *24*, 102003.

Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., & Kim, B. (2018). Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems* (pp. 9505-9515).
Kindermans, P. J., Hooker, S., Adebayo, J., Alber, M., Schütt, K. T., Dähne, S., ... & Kim, B. (2019). The (un) reliability of saliency methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning* (pp. 267-280). Springer, Cham.

6.2 Bias and fairness

Please refer to Section 5.1 ("Data Quality - Bias and Fairness")

6.3 Robustness

Authors: Federico Cabitza, Saul Calderon, Jörg Martin

Reference Summary

Overview

Modern artificial intelligence solutions are mostly data-driven approaches, relying on large training datasets. Such training data need to be *representative* enough of the problem at hand, improving model accuracy with unseen testing data, namely model generalization. Within this context, robustness refers to the ability of a model to retain its accuracy performance to significant data changes from training to test data, or at least, make the user aware of the abnormal operation conditions (see explainability section).

The following are major sources of data instability (difference between training and testing data):

1. Outliers: Isolated extreme values in the test set not seen in the training data might considerably perturbate the model output **[4]**.
2. Data distribution mismatch: The distribution of the training data might be significantly different to the distribution of the test data **[5]**.
3. Adversarial attacks: In different applications, external attackers might be interested to deceive the model, by hijacking it with artificial adversarial inputs [1].

Measuring robustness can be done from different perspectives or a combination of them:

1. In the input space, using both training and test data samples. The distance/dissimilarity or distribution mismatch can be measured as advised in [2]. Measuring training and data discrepancy can suggest how challenging the scenario is for a model in terms of retaining its accuracy or explaining its accuracy decrease. Also data representativity (sparseness or heterogeneity **[3]**) can be measured in both the training and test datasets.
    a. A specific robustness scenario is noise in the data labels. A model can be considered robust if its accuracy does not change much when trained with high degree of noisy labels.
2. In the feature space: Modern AI solutions make extensive use of deep learning architectures. These architectures often learn a feature space from the data, making its usage more feasible for measuring training and test data similarity **[3]**. .
3. Model output: Model output perturbations to significant training and test data mismatch. The sensitivity of the model to an increasing training to test data/features to discrepancy is a common approach to define model robustness [62,66].

Therefore, we can define robustness as the rate between model output perturbation and training-test data dissimilarity. Robustness assessment is of key importance in medical applications, as the input test data perturbation is a frequent challenge in real-world medical AI systems.

Measures and Recommendations

Dataset distances and mismatch

Dataset representativity assessment

Outlier testing (generative methods [52,53], Bayesian uncertainty [62,66])

Uncertainty quantification (Gaussian Processes [62], aleatoric uncertainty for deep learning AI systems [67], epistemic uncertainty for deep learning AI systems [66])

Robustness validation

See Uncertainty robustness validation approaches

Alarm systems

Outlier tests (generative methods [52,53], Bayesian uncertainty [62,66])

Attribution methods (gradient*input [54], integrated gradients [55], layerwise relevance propagation [56]/deep Taylor decomposition [57], perturbation-based attribution [58])

Uncertainty quantification (Gaussian Processes [62], aleatoric uncertainty for deep learning AI systems [67], epistemic uncertainty for deep learning AI systems [66])

References

Introduction

Modern deep learning solutions are labelled data hungry. They need high quality labelled and representative enough data. Test data might have a different distribution, either with isolated (rare) cases, known as individual outliers, or in a more challenging scenario, might consist of collective outliers, leading to distribution miss-match of the training and test dataset **[1,2]**. Also, test data can include what is known as adversarial attacks, launched by external malicious agents aiming to deceive the model **[3]**. Also, an usual perturbation is data imbalance in either the training or test data **[9]**. Aside from perturbations coming from the test data, the training dataset can also include perturbations. Noisy inputs **[5]** and labels **[4]** are a frequent short-coming faced in real-world machine learning applications. Robustness can be defined as the ratio between the accuracy of the model variation and the amount of perturbation applied either to the training/test data or the labels of the training data.

$$\frac{\mathrm{a}_{no\_perturbation} - \mathrm{a}_{perturbation}}{\Delta_x}$$

where Δ_{X} refers to the perturbation applied to the training/test data.

1. **Training and test data distances**

A simple robustness measure based in the distance between the train and test dataset is defined as follows **[7,8]**. The main idea is to try to match each data point in the test set with those in the reference population: after exchanging the matched objects we then assess if there has been any changes in the topology.

More precisely, the method to compute the Φ is the following one:

1) Compute the distance distribution in the larger group (e.g., training set)

2) Match each object in the smaller sample (e.g. the test set) with the most similar object in the larger sample.

3) Then substitute the objects in the larger dataset and check if there is a significant difference between the pre- and post-substitution distance distributions with an equality test.

4) The value of Φ is equal to the p-value.

Then the robustness of the model is defined as:

$$R = \frac{H}{1 + \Phi}$$

Thus the robustness of the model has its maximum value (equal to the value of H) when Φ is equal to 0 (so the training set and the test are maximally different) and minimum value (equal to H/2) when Φ is equal to 1 (when the test set is essentially a subset of the training set).

Out of distribution measures

The distribution of the test data, making it different from the distribution of the training data. This is known as out of distribution problem. In the literature a wide range of out-of-distribution detectors have been developed, implicitly or explicitly manipulating out of distribution measures [2].

Simple measures can be implemented, such as the entropy of the softmax from the output layer in a neural network, or the maximum value of the output layer REF.

Data heterogeneity as perturbation

The heterogeneity, sparseness or “representativeness” of the training data can also be an indirect measure of the training data quality, hence also the robustness to variations of this aspect can be measured [7].

Distribution mismatch or training to test data distance can alternatively be measured in the feature space. Common deep architectures learn the feature space from data. The feature space has a lower dimensionality, easing further computations. The work in [7] implements the coefficient of variation of automatically learned clusters in the feature space.

The approach in [7] performs data clustering resulting from the fuzzy k-means algorithm. The coefficient of variation is measured as follows

$$\delta = \frac{1}{C}\sum_{i,j}^{C}\frac{\bar{x}_i - \bar{x}_j}{s_j}$$

The coefficient of variance δ is calculated for a number of clusters C, with centroid x and standard deviation s. A lower coefficient of variation is correlated to lower data heterogeneity. In **[7] two different medical imaging datasets were used to measure** $\delta$with different dataset sizes. As expected, δ decreases with a larger dataset, and reaches a stable point, depending on the dataset.

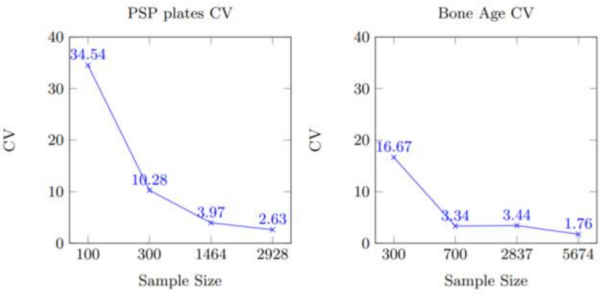


Fig. 4. The Coefficient of Variation for PSP plates clusters over different sample sizes (left). The Coefficient of Variation for Bone Age clusters over different sample sizes (right).

Figure x – Missing caption

The accuracy vs the coefficient of variation can be used as a robustness measure. This is plotted as follows.

[CHECK IF WE CAN INCLUDE IMAGE]

Noisy labels

Noisy labels are a frequent short-coming in different domains, and also in the medical domain. High quality datasets are expensive, as good labels need a number of experts labeling each observation. The use of a normalized metric for robustness to label noise is still not extended. However, literature on label noise robustness is increasing. For instance, in **[10,11]**, the usage of pretraining and self supervised learning (respectively) is tested to measure the robustness increase to noisy labels. The plots below show the accuracy of the model at different degrees of adversarial label corruption.

[CHECK IF WE CAN INCLUDE IMAGE]

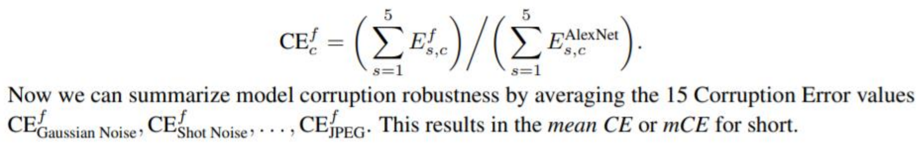
Input noise and degradation

Noise in the inputs, as in images and data records is also usual in both training and test datasets. For images, deep learning architectures are known to be robust to common gaussian noise and impulsive noise [5]. However other types or degradations can hinder model's accuracy [5].

In [12] an extensive testing of different CNN architectures is performed to different types of input corruption. The next Figure enlists common types of image input degradations.

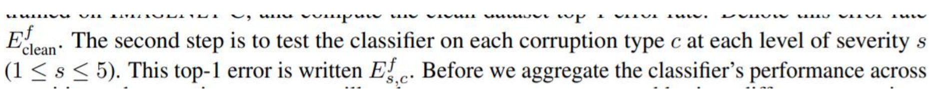
[CHECK IF WE CAN INCLUDE IMAGE]

Authors proposed a summarized metric for all the types of corruption tested, as seen below:



$$\mathrm{CE}_c^f = \left(\sum_{s=1}^{5} E_{s,c}^f\right) \Big/ \left(\sum_{s=1}^{5} E_{s,c}^{\mathrm{AlexNet}}\right).$$

Now we can summarize model corruption robustness by averaging the 15 Corruption Error values $\mathrm{CE}^f_{\text{Gaussian Noise}}, \mathrm{CE}^f_{\text{Shot Noise}}, \ldots, \mathrm{CE}^f_{\text{JPEG}}$. This results in the *mean CE* or *mCE* for short.



trained on IMAGENET-C, and compute the clean dataset top-1 error rate. Denote this error rate E^f_{clean}. The second step is to test the classifier on each corruption type c at each level of severity s $(1 \leq s \leq 5)$. This top-1 error is written $E^f_{s,c}$. Before we aggregate the classifier's performance across

Robustness can be measured using bounds, defining tolerance intervals

Class imbalance as a perturbation

Class imbalance is also a usual shortcoming faced in practical applications. For instance, in the medical domain, is hard to get observations of certain pathologies, leading to strongly imbalanced datasets. Measuring the robustness to data imbalance and correcting it, can be useful in different settings. In **[15]**, the robustness of different deep classifiers to data imbalance is analysed.

References

[1] Singh, Karanjit, and Shuchita Upadhyaya. "Outlier detection: applications and techniques." *International Journal of Computer Science Issues (IJCSI)* 9.1 (2012): 307.

[2] Sreeramdass, V., & Sarawagi, S. OUT-OF-DISTRIBUTION IMAGE DETECTION WITH DEEP NEURAL NETWORKS

[3] Carlini, N., & Wagner, D. (2017, May). Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)* (pp. 39-57). IEEE.

[4] Vahdat, Arash. "Toward robustness against label noise in training deep discriminative neural networks." *Advances in Neural Information Processing Systems*. 2017.

[5] Nazaré, Tiago S., et al. "Deep convolutional neural networks and noisy images." *Iberoamerican Congress on Pattern Recognition*. Springer, Cham, 2017.

[6] Mendez, M., Calderon, S., & Tyrrell, P. N. (2019, September). Using Cluster Analysis to Assess the Impact of Dataset Heterogeneity on Deep Convolutional Network Accuracy: A First Glance. In *Latin American High Performance Computing Conference* (pp. 307-319). Springer, Cham.

[7] Cabitza, F., & Campagner, A. (2019). Who wants accurate models? Arguing for a different metrics to take classification models seriously. *arXiv preprint arXiv:1910.09246.*

[8] Cabitza, F., Campagner, A., & Sconfienza, L.. (Submitted). As if sand were stone. New concepts and metrics to probe the ground on which to build trustable AI. *Submitted to BMC Medical Informatics and Decision Making*.

[9] Wang, Shuo, and Xin Yao. "Multiclass imbalance problems: Analysis and potential solutions." *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 42.4 (2012): 1119-1130.

[10] Hendrycks, Dan, Kimin Lee, and Mantas Mazeika. "Using pre-training can improve model robustness and uncertainty." *arXiv preprint arXiv:1901.09960* (2019).

[11] Hendrycks, Dan, et al. "Using self-supervised learning can improve model robustness and uncertainty." *Advances in Neural Information Processing Systems*. 2019.

[12] Hendrycks, Dan, and Thomas Dietterich. "Benchmarking neural network robustness to common corruptions and perturbations." *arXiv preprint arXiv:1903.12261* (2019).

[13] Rafajłowicz, Ewaryst. "Robustness of raw images classifiers against the class imbalance–a case study." *IFIP International Conference on Computer Information Systems and Industrial Management*. Springer, Cham, 2018.

[15] Rafajłowicz, Ewaryst. "Robustness of raw images classifiers against the class imbalance–a case study." *IFIP International Conference on Computer Information Systems and Industrial Management*. Springer, Cham, 2018.

Machine learning, robustness and health applications

Machine Learning is a flexible, powerful tool that has shown remarkable successes especially in cases where the design of an explicit algorithm might be too evolved or even impossible to perform. The key idea of using a generic model and adjusting it using data comes, however, with a price. While the trained AI might perform well in practice, it is usually hard to understand why it works (Olden & Jackson, 2002). This immediately raises the question if we can trust an AI build with machine learning. Szegedy had in 2013 the discomforting inside that it's in fact rather easy to fool such a system (Szegedy et al., 2013). We can find a small perturbation of the input, often on a scale that's imperceptible to a human, that will lead the AI to a wrong decision. These *adversarial examples* seem to exist for all kinds of applications provided the data is sufficiently high dimensional (Goodfellow et al., 2014), (Kurakin et al., 2016).

Relevance to medicine

For a medical application of AI the existence of adversarial examples is relevant for two reasons:

– Obviously, they allow for *human manipulation* of such systems, which might be done for all kinds of motifs including for instance insurance frauds (Mirsky et al, 2019).

– But even neglecting the risk of human manipulation, robustifying against adversarial examples has shown to be a good defense against *unexpected behavior* of a system in practice (Shaham et al., 2018).

Metrics and strategies

Strategies to cope with the issue of adversarial attacks can be split into two philosophies:

– Including adversarial examples while training and thus robustifying the system.

– Detect adversarial attacks in the input before they are handed to the AI.

The first method is known to robustify the AI and also to increase its general performance. However, it only applies to the adversarial attacks used in training and can only be applied during training time and by someone who is in possession of a suitable dataset.

Methods to detect adversarial examples are

- Using another AI for detection (Metzen et al., 2017), (Xu et al, 2017)
- Using quantities that evaluate the trustworthiness of the input, such as
  - Entropy and mutual information (Feinman et al., 2017)
  - Uncertainty, e.g. using Variational inference or dropout (Smith & Gal., 2018), (Rawat et al., 2017)
  - statistical quantities such as the Fisher information (Martin & Elster, 2019)

While using AI systems seem to achieve the highest performance rates, they suffer once more from the "Black Box" property and are thus explaining why an input is classified as adversarial. Statistical quantities often have a rather clear interpretation and can even be used to visualize unusual regions in the input (Martin & Elster, 2019).

References

Olden, J. D., & Jackson, D. A. (2002). Illuminating the "black box": a randomization approach for understanding variable contributions in artificial neural networks. *Ecological modelling*, *154*(1-2), 135-150.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Kurakin, A., Goodfellow, I., & Bengio, S. (2016). Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*.

Mirsky, Y., Mahler, T., Shelef, I., & Elovici, Y. (2019). CT-GAN: Malicious tampering of 3D medical imagery using deep learning. In *28th {USENIX} Security Symposium ({USENIX} Security 19)* (pp. 461-478).

Shaham, Uri, Yutaro Yamada, and Sahand Negahban. "Understanding adversarial training: Increasing local stability of supervised models through robust optimization." *Neurocomputing* 307 (2018): 195-204.

Metzen, J. H., Genewein, T., Fischer, V., & Bischoff, B. (2017). On detecting adversarial perturbations. *arXiv preprint arXiv:1702.04267*.

Xu, W., Evans, D., & Qi, Y. (2017). Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*.

Feinman, R., Curtin, R. R., Shintre, S., & Gardner, A. B. (2017). Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*.

Smith, L., & Gal, Y. (2018). Understanding measures of uncertainty for adversarial example detection. *arXiv preprint arXiv:1803.08533*.

Rawat, A., Wistuba, M., & Nicolae, M. I. (2017). Adversarial phenomenon in the eyes of Bayesian deep learning. *arXiv preprint arXiv:1711.08244*.

Martin, J., & Elster, C. (2019). Inspecting adversarial examples using the Fisher information. *Neurocomputing*.

6.4 Generalization

Authors: Alexandre Chiavegatto Filho, Federico Cabitza

The ability to generalize a model, i.e. how well it can correctly predict the occurrence of events when exposed to a new set of data, depends on a balance between the bias and the variance of the model. Bias in machine learning, if not controlled, increases training and generalization error by oversimplifying model assumptions. Variance, on the other hand, happens when small fluctuations in the training set lead to a significant increase in generalization error. Some important issues arise when trying to achieve a balance between bias and variance.

Importance:

Machine learning is a practical area that requires simulations to verify generalization potential. It is increasingly possible to understand how the automation of procedures can facilitate the daily lives of health professionals, from simple tasks to the most complex ones (Obermeyer & Lee, 2017; Rajkomar et al., 2019). The goal of most machine learning studies is to one day apply the models in real world settings, which means the algorithms must be generalizable to new datasets.

Generalization types:

a. Local Generalization: when the objective is to predict an outcome using data from the same location as the training set. Also known as internal validation or reproducibility (Steyerberg, 2019).
Limitation: new data may not follow the same pattern over time as the data used in model training: new interventions may be introduced after the training set was collected and new diseases may emerge.

b. Extrapolation: There are increasing challenges when the objective is to extrapolate, that is, to apply the model to a different area from the one used in training the algorithm. It is also known as external validation or transportability (temporal, geographical, methodological and spectrum) (Steyerberg, 2019).
Limitation: The algorithms need to be applied to plausibly related populations, i.e. populations in which there are similar relationships between the predictors and the outcome. However, this is hard to be tested empirically and could increase prediction error.

How to generalize?

To assess the potential generalization ability of a model, it is necessary to use a test sample, preferably by following these steps:

I. First, the sample must be divided into a training set and a test set. The training set can be further subdivided into a training set and a validation set in order to tune hyperparameters (or by using cross-validation exclusively in the training set). The test set will only be analysed at the end, in order to measure the actual performance of the algorithm.

II. Pre-processing is a crucial step to guarantee model generalization, which frequently consists of:

- Verifying the need for data standardization (regarding scaling and missing data);
- Exclusion of variables that allow the algorithm to identify the outcome, i.e. that is a proxy for the outcome (known as data leakage);
- Exclusion of predictors with high correlation, or application of dimensionality reduction techniques;
- Exclusion of predictors with degenerate distribution or variance close to zero;

– Transformations in qualitative predictors (by applying one-hot encoding or dummy transformations);

– Treatment of missing data (removal of observations, create a new category for missing values in case of categorical variable, or imputing by interpolation);

Note: The data pre-processing step occurs only with information from the training data (e.g. in the case of calculating the mean and the standard deviation for variable scaling), in order to prevent the algorithm from learning from the test data.

III. Hyperparameters are optimized with a validation or cross-validation sample. This adjustment is necessary to improve the generalization performance of the algorithm in order to avoid overfitting and to balance the trade-off between bias and variance;

IV. The final model is trained in the entire training set and its parameters are recorded;

V. Performance is then measure in the test set, i.e. the algorithm makes prediction for the data set separated in the initial phase, which allows to estimate the efficiency of the algorithm in an unknown data set;

VI. Most common model performance metrics:

– Regression Model: Mean Squared Error (MSE) or Root Mean Squared Error (RMSE);

– Classification Model: Accuracy, Sensitivity, Specificity, F-Score and AUC.

After analysing predictive performance in the test, new questions arise:

2.1 Will the prediction work in clinical practice? It is necessary to understand the reality of where the prediction will later be performed, and which are the possible problems when incorporating it into clinical practice. It is necessary to understand, for example, whether there will be resistance from doctors and other health professionals to use the results of predictive models. In addition, it is necessary to understand possible abnormal variations of the data and other reasons that may prevent the prediction from working in its practical use (Ghassemi et al., 2019).

2.2 Similarity of variables. If the training dataset has the same variables as the target generalization dataset, replication or transportability of the predictive model is possible. However, changes in data collection methodologies (e.g. exclusion of a variable or re-categorization of qualitative variables) may hinder generalization. For example, in the case of an algorithm that was trained with a continuous variable regarding the frequency of fruit intake during the week, but in the new data set there are only grouped frequencies (e.g. < 3 our ≥ 3 intakes per week). In such cases, if this variable is meaningful for prediction, any pre-processing of this variable to try to turn it into new continuous values will probably decrease the overall performance of the model.

2.3 To which regions or groups (even across time in the same place) is the original model generalizable? Generalizations need to take into account a few issues (König et al., 2007; Steyerberg, 2019):

– Temporal transportability: when the datasets are from different periods, especially in the presence of seasonal diseases.

– Geographic transportability: when the datasets are from different locations, that may not have the same patterns regarding the relationship between the predictors and the outcome of interest.

– Spectral transportability: when the datasets have different diseases or stages of the disease. This is especially important if the outcome is binary disease prediction, which may have a wide range of severity.

2.4 Importance of representativeness of training data. Diversity of training data is a growing issue in data generalization. It is necessary that the data in which the algorithm was trained contains enough observations for each subgroup in which the model will be generalized, in order for the algorithm to learn subgroup-specific patterns (Géron, 2019).

References

Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media.

Ghassemi, M., Naumann, T., Schulam, P., Beam, A. L., Chen, I. Y., & Ranganath, R. (2019). Practical guidance on artificial intelligence for health-care data. The Lancet Digital Health, 1(4), e157-e159.

Kim, H. E., Kim, H. H., Han, B. K., Kim, K. H., Han, K., Nam, H., ... & Kim, E. K. (2020). Changes in cancer detection and false-positive recall in mammography using artificial intelligence: a retrospective, multireader study. The Lancet Digital Health, 2(3), e138-e148.

König, I. R., Malley, J. D., Weimar, C., Diener, H. C., & Ziegler, A. (2007). Practical experiences on the necessity of external validation. *Statistics in medicine*, *26*(30), 5499-5511.

Obermeyer, Z., & Lee, T. H. (2017). Lost in thought: The limits of the human mind and the future of medicine. The New England journal of medicine, 377(13), 1209.

Rajkomar, A., Dean, J., & Kohane, I. (2019). Machine learning in medicine. New England Journal of Medicine, 380(14), 1347-1358.

Steyerberg, E. W. (2019). Clinical prediction models. Springer International Publishing.

Zhang, J. M., Harman, M., Ma, L., & Liu, Y. (2020). Machine learning testing: Survey, landscapes and horizons. IEEE Transactions on Software Engineering.

Proposal: Degree of correspondence

Generalization refers to the ability of a predictive model to correctly behave on unseen data. In order to test the generalization ability of a model, typically the data is split into a training set, which is used to fit the model’s parameters, and a test set which is used to assess whether the model generalizes and is supposed to be representative of the reference population. Then generalization can be assessed evaluating if the test really is representative of the reference population.

We propose a metrics, that we call *degree of correspondence* Φ. The main idea is to try to match each data point in the test set with those in the reference population: after exchanging the matched objects we then assess if there have been any changes in the topology.

More precisely, the method to compute the Φ is the following one:

1. Compute the distance distribution in the larger group (e.g., reference population)
2. Match each object in the smaller sample (e.g. the test set) with the most similar object in the larger sample.
3. Then substitute the objects in the larger dataset and check if there is a significant difference between the pre- and post-substitution distance distributions with an equality test.
4. The value of Φ is 1 – p-value.

In particular, step 2 can be obtained by solving the following linear assignment problem

$$min \sum_{i,j} x_{i,j}\, d(i,j), \;\; where \begin{cases} \forall i \sum_{j} \; x_{i,j} \leq 1 \\ \forall j \sum_{i} \; x_{i,j} = 1 \\ \; x_{i,j} \in \{0,1\} \end{cases}$$

Where i is an instance in the larger group (reference population) and j is an instance in the smaller sample (test set) and d(i,j) is their distance.

References

Cabitza, F., Campagner, A., & Sconfienza, L., (Submitted). As if sand were stone. New concepts and metrics to probe the ground on which to build trustable AI. *Submitted to BMC Medical Informatics and Decision Making*.

6.5 Uncertainty

Authors: Luis Oala, Vignesh Srinivasan, Wojciech Samek

Reference Summary

Overview

Modern AI systems based on deep learning, reinforcement learning or hybrids thereof are powerful technologies. They are also fickle technologies whose behaviour is often hard to fathom. This creates a risk for system failure which is of particular concern when attempting to deploy AI systems in health applications.

Decades of machine learning research has produced a number of tools to enhance the so-called robustness of AI systems. Many of today's research efforts around deep and reinforcement learning attempt to find improved ways of doing so. In this report we provide a working definition for robustness as well as a high-level illustration of the two potential sources of robustness risks for AI systems. We explain how robustness enhancing tools can contribute to making AI systems safer and more reliable. In addition, we identify four action areas along the life cycle of AI systems for mitigating robustness risks. While we are not claiming completeness we note that a breadth of tools are covered, the time-tested alongside the very recent. We hope this report provides meaningful concepts and categories for facilitating an informed and interdisciplinary discussion of AI system robustness in the context of health applications.

Measures and Recommendations

Data fidelity
Data diversity [6]
Pre-processing (zero centring, PCA [7], whitening [7])
Normalization (standardization, min-max scaling)

Robust training
Adversarial training [12]
Generative methods [18,19,20,21,22, 24]
Stability training [25]
FAT optimization objectives [44,45,46,47,48]

Robustness validation
Cross-validation [26]
Classical tests (e.g. t-test, F-test [27], serial autocorrelation [29, 30])
Information criteria (AIC [31], BIC [32], Occam-weighted likelihood [33])
Vulnerability tests (PGD [12])
FAT validation metrics [38,39,40,41,42,43]
FAT validation toolboxes [49, 50, 51]

Alarm systems
Outlier tests (generative methods [52,53], Bayesian uncertainty [62,66])
Attribution methods (radient*input [54], integrated gradients [55], layerwise relevance propagation [56]/deep Taylor decomposition [57], perturbation-based attribution [58])
Uncertainty quantification (Gaussian Processes [62], aleatoric uncertainty for deep learning AI systems [67], epistemic uncertainty for deep learning AI systems [66])

References

[6] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumeé III, and Kate Crawford. Datasheets for datasets.*arXiv preprint arXiv:1803.09010*, 2018.

[7] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2Nd Edition)*. Wiley-Interscience, New York, NY, USA, 2000.

[12] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[18] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*, volume 9, 2018.

[19] Andrew Ilyas, Ajil Jalal, Eirini Asteri, Constantinos Daskalakis, and Alexandros G Dimakis. The robust manifold defense: Adversarial training using generative models. *arXiv preprint arXiv:1712.09196*, 2017.

[20] Shiwei Shen, Guoqing Jin, Ke Gao, and Yongdong Zhang. Ape-gan: Adversarial perturbation elimination with gan. *ICLR Submission, available on OpenReview*, 4, 2017.

[21] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *arXiv preprint arXiv:1710.10766*, 2017.

[22] Lukas Schott, Jonas Rauber, Wieland Brendel, and Matthias Bethge. Robust perception through analysis by synthesis. *arXiv preprint arXiv:1805.09190*, 2018.

[24] Vignesh Srinivasan, Arturo Marban, Klaus-Robert Müller, Wojciech Samek, and Shinichi Nakajima. Counterstrike: Defending deep learning architectures against adversarial samples by langevin dynamics with supervised denoising autoencoder. *arXiv preprint arXiv:1805.12017*, 2018.

[25] Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow. Improving the Robustness of Deep Neural Networks via Stability Training. *arXiv:1604.04326 [cs]*, April 2016. arXiv: 1604.04326.

[26] Christopher M. Bishop. *Pattern recognition and machine learning*. Information science and statistics. Springer, New York, NY, corrected at 8th printing 2009 edition, 2009. OCLC: 845772798.

[29] L. G. Godfrey. Testing Against General Autoregressive and Moving Average Error Models when the Regressors Include Lagged Dependent Variables. *Econometrica*, 46(6):1293–1301, 1978.

[30] T. S. Breusch. Testing for Autocorrelation in Dynamic Linear Models*. *Australian Economic Papers*, 17(31):334– 355, 1978.

[31] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716– 723, December 1974.

[32] Gideon Schwarz. Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2):461–464, 1978.

[33] David J. C. MacKay. *Bayesian methods for adaptive models*. phd, California Institute of Technology, 1992.

[38] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, & R. Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference* (pp. 214-226). ACM, 2012.

[39] N. Papernot, P. McDaniel, A. Sinha, & M. Wellman. Towards the science of security and privacy in machine learning. Preprint at:arXiv:1611.03814, 2016.

[40] P. Gajane, & M. Pechenizkiy. On formalizing fairness in prediction with machine learning. Preprint at:arXiv:1710.03184, 2017.

[41] A. Weller. Challenges for Transparency. CoRR. Preprint at:arXiv:1708.01870v1, 2017.

[42] G. Yona & G. Rothblum. Probably Approximately Metric-Fair Learning. In *International Conference on Machine Learning* (pp. 5666-5674). Retrieved from http://proceedings.mlr.press/v80/yona18a.html, 2018.

[43] E. D. Moss. Translation Tutorial: Toward a Theory of Race for Fairness in Machine Learning. *In Proceedings of FAT* conference (FAT* Conference). ACM, New York, NY, USA, 2 pages*. Retrieved from https://drive.google.com/file/d/1jKSbnAg7NPotjhvNb8o4YXyjcjF8RTJW/view, 2019.

[44] S. Shaikh, H. Vishwakarma, S. Mehta, K. R. Varshney, K. N. Ramamurthy, & D. Wei. An End-To-End Machine Learning Pipeline That Ensures Fairness Policies. *CoRR, abs/1710.06876*. Preprint at:arXiv:1710.06876v1, 2017.

[45] A. Paul, C. Jolley & A. Anthony. Reflecting the Past, Shaping the Future: Making AI Work for International Development. Retrieved from USAID: https://www.usaid.gov/sites/default/files/documents/15396/AI-ML-in-Development.pdf, 2018.

[46] R. Dobbe & M. Ames. Translation Tutorial: Values, Reflection and Engagement in Automated Decision-Making. *In Proceedings of ACM Conference on Fairness, Accountability, and Transparency (ACM FAT* 2019). ACM, New York, NY, USA, Article 4, 2 pages.* Retrieved from https://drive.google.com/file/d/1vhkLBfDUtsd9hXfbXz8QwFo8GYGZyAWP/view, 2019.

[47] A. Albarghouthi, & S. Vinitsky. Fairness-Aware Programming. In *FAT* '19: Conference on Fairness, Accountability, and Transparency*, January 29–31, 2019, Atlanta, GA, USA. ACM, New York, NY, USA, 9 pages. Doi:10.1145/3287560.3287588. 2019.

[48] K. Holstein, J. W. Vaughan, H. Daumé III, M. Dudík, & H. Wallach. Improving fairness in machine learning systems: What do industry practitioners need?. *ACM CHI Conference on Human Factors in Computing Systems (CHI 2019).* Preprint at:arXiv:1812.05239v2, 2018.

[49] R. K. Bellamy, K. Dey, M. Hind, S. C. Hoffman, S. Houde, K. Kannan, ... & S. Nagar. AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. Preprint at:arXiv:1810.01943v1, 2018.

[50] S. A. Friedler, C. Scheidegger, S. Venkatasubramanian, S. Choudhary, E. P. Hamilton, & D. Roth. A comparative study of fairness-enhancing interventions in machine learning. Preprint at:arXiv:1802.04422v1, 2018.

[51] Z. Epstein, B. H. Payne, J. H. Shen, C. J. Hong, B. Felbo, A. Dubey, ... & I. Rahwan. TuringBox: An Experimental Platform for the Evaluation of AI Systems. In *Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, PMLR 80*, 2018. (pp. 5826-5828). Retrieved from http://www.fatml.org/media/documents/turing_box.pdf, 2018.

[52] Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 135–147. ACM, 2017.

[53] Nicholas Frosst, Sara Sabour, and Geoffrey Hinton. Darccc: Detecting adversaries by reconstruction from class conditional capsules. *arXiv preprint arXiv:1811.06969*, 2018.

[54] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning Important Features Through Propagating Activation Differences. In *ICML*, 2017.

[55] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic Attribution for Deep Networks. In *ICML*, 2017.
[57] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep Taylor decomposition. *Pattern Recognition*, 65:211–222, 2017.
[58] Matthew D. Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks. In *ECCV*, 2014.
[66] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *ICML*, 2016.

Introduction

Modern AI systems based on deep learning, reinforcement learning or hybrids thereof are powerful technologies. They are also fickle technologies whose behaviour is often hard to fathom. This creates a risk for system failure which is of particular concern when attempting to deploy AI systems in health applications.

Decades of machine learning research has produced a number of tools to enhance the so-called robustness of AI systems. Many of today's research efforts around deep and reinforcement learning attempt to find improved ways of doing so. In this report we provide a working definition for robustness as well as a high-level illustration of the two potential sources of robustness risks for AI systems. We explain how robustness enhancing tools can contribute to making AI systems safer and more reliable. In addition, we identify four action areas along the life cycle of AI systems for mitigating robustness risks. While we are not claiming completeness we note that a breadth of tools are covered, the time-tested alongside the very recent. We hope this report provides meaningful concepts and categories for facilitating an informed and interdisciplinary discussion of AI system robustness in the context of health applications.

The AI System Life Cycle and Robustness

The life cycle of an AI system can be organized into for general steps which are visualized in Figure XX1. The first step comprises defining an AI system. This includes the choice of a model H, a training environment Φ (data) and requirements Ψ (e.g. optimization objective, evaluation metrics). Then in the second step the model H is trained until it fulfills the specified requirements Ψ. After the training has concluded the model is typically validated in step three. After successful validation the model can then be considered for deployment in step four.

In the words of Peter J. Huber robustness can broadly be understood as a model's "insensitivity to small deviations from the assumptions" [1] that were initially made in steps one and three. While useful this definition appears too narrow for the setting, we regularly find ourselves in with modern AI systems. It is often not even known what assumptions can be made when working deep learning AI systems. Thus, a broader definition is needed. Thomas G. Dietterich advances a robustness view that distinguishes between how an AI system behaves in the face of known unknowns and unknown unknowns [2]. Stuart Russell, Daniel Dewey and Max Tegmerk group similar concerns under the term validity. They point to two relevant dimensions of robustness evaluation: the environment and requirements under which an AI system operates [3]. For this report we utilize the following working definition of robustness, drawing from the previous views: robustness is a desideratum we place on an AI system to not commit any gross, unexpected errors under slight changes of the environment Φ or to at least handle them benignly, e.g. by letting a human AI system operator know that something unusual has happened. As we later explain in detail contributions to enhanced model robustness can be made at each step of the AI system life cycle. Robustness during step four, deployment, is of particular concern when considering the application of AI systems. Following the analysis of [3] robustness risks at the deployment step can originate from two potential sources:

– First, it is possible that the environment Φ_n in which a model operates is different from the one Φ_m it was calibrated in. For example, it has been shown that standard convolutional neural networks used for image classification are not invariant to common perturbations like blurring [4].

– Second, it may turn out that the requirements Ψ_m we originally specified were insufficient to capture some behaviour we actually care about. This might mean we need to come up with new evaluation metrics to capture the erroneous behaviour and make it visible. For example, new metrics are actively being researched to avoid racial and gender bias in image classification models [5].

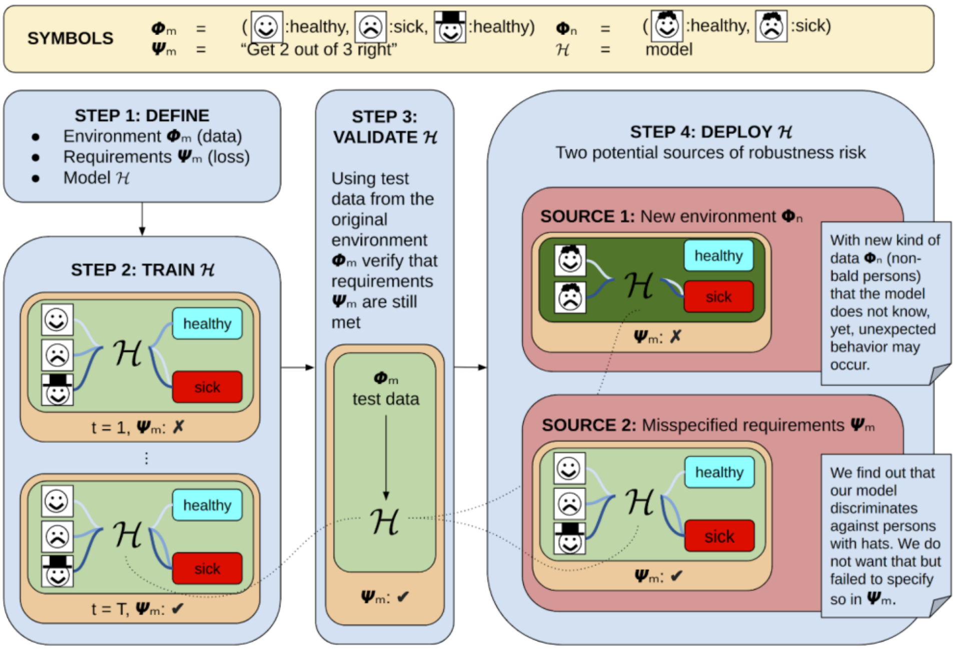

SYMBOLS
Φm = (:healthy, :sick, :healthy)
Ψm = "Get 2 out of 3 right"
Φn = (:healthy, :sick)
H = model
STEP 1: DEFINE
Environment Φm (data)
Requirements Ψm (loss)
Model H
STEP 2: TRAIN H
healthy
sick
t = 1, Ψm: ✗
t = T, Ψm: ✓
STEP 3: VALIDATE H
Using test data from the original environment Φm verify that requirements Ψm are still met
Φm test data
H
Ψm: ✓
STEP 4: DEPLOY H
Two potential sources of robustness risk
SOURCE 1: New environment Φn
Ψm: ✗
With new kind of data Φn (non-bald persons) that the model does not know, yet, unexpected behavior may occur.
SOURCE 2: Misspecified requirements Ψm
Ψm: ✓
We find out that our model discriminates against persons with hats. We do not want that but failed to specify so in Ψm.

Figure XX1 – AI system life cycle and possible sources of robustness risk

Figure XX1 describes the life cycle of an AI system and the two possible sources of robustness risk. In Step 1 the environment Φm (training data), requirements Ψm (optimization objective, evaluation metrics) and model H for the AI system are decided upon. Then in step 2 the AI system is trained until the requirements Ψm are fulfilled. Step 3 comprises validating the AI system on test data from the same distribution as Φm. Finally, in step 4 the model is deployed. During deployment two possible sources of robustness risk can be identified. Source 1 constitutes changes in the environment, i.e. a new type of data the model has not seen during training. Source 2 comprises misspecifications of the requirements that were used for model training, e.g. we might have failed to account for the fact that no discrimination should take place on account of a person wearing a hat or not.

Four Action Areas for Enhancing Robustness

We grouped the available tools to enhance the robustness of an AI system into four action areas along the life cycle steps. As visualized in Figure XX2 these four groups comprise data fidelity, robust training, robustness validation and alarm systems. In the following we will explain these in detail and list available tools for each action area (see Table XX1 for an overview of all tools).

Data fidelity can be understood as imposing desiderata on the data that is being used for training an AI system. This can take the form of diversity criteria to mandate a balance with respect to certain features like age, socioeconomic status or race. The datasheets approach proposed by [6] is a case in point. Datasheets would summarize dataset key statistics along with usage recommendations and aspects that users of this dataset should be aware of. Another, and often used, data fidelity tool is pre-processing and normalization. This is commonly used to ensure that input data during deployment lie in the same range as during training or to satisfy certain modelling assumptions, e.g. uncorrelated inputs. Popular tools include zero centring data (each input dimension will have a mean of zero), principal component analysis (commonly used for decorrelating data) [7], whitening (scaling decorrelated data to unit variance) [7], standardization (normalize scales across dimensions) and min-max scaling (normalize data to ranges [-1,1] or [0,1]).

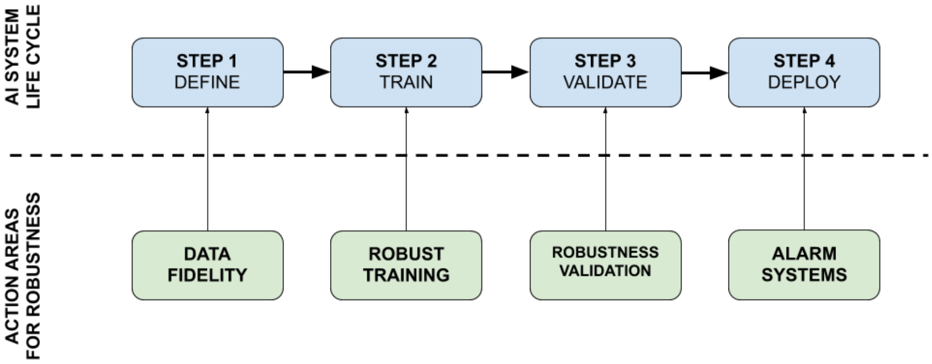


Figure XX2 – AI system life cycle alongside robustness areas

Figure XX2 illustrates four steps of an AI system life cycle alongside four possible action areas for enhanced robustness. We can impose requirements on the fidelity of the data, e.g. by restricting the type of data a model can take as input. Another strategy is to design the training procedure in a way that robustness enhancing methods are being used, e.g. adversarial training. In addition, we can use the validation step in the AI system workflow to probe the model in new environments or under new requirements. Finally, we can mandate the use of different alarm systems that indicate to an AI system operator when the AI system is confronted with an unknown situation during deployment.

Robust training comprises a group of methods that help exposing an AI system to changes in the data environment that would otherwise be likely to induce robustness risks during deployment. In this way the AI system can be seen as "getting used" to the types of environment changes that would otherwise cause it to break. An important tool in this action area is adversarial training [8,9,10,11] which aims at reducing an AI system's vulnerability to adversarial examples: data points that to humans are visually indistinguishable from original inputs but the AI system nevertheless misclassifies. Under adversarial training such examples are included in the training data so that the AI system can learn to treat them correctly. Currently, the most popular method follows [12]. Another strategy to achieve this is by employing generative models. Generative algorithms [13,14,15,16,17] model how the data was generated before classifying it. They are also an effective alternative for protecting AI systems against adversarial attacks [18,19,20,21,22]. However, most of these methods have been found not to work effectively at protecting the AI system classifier. An attacker can specifically target the weakness of the reconstruction algorithm and craft an adversarial example for the AI system classifier [23]. This problem can be alleviated by employing Langevin Dynamics (LD) [24]. Finally, robust training can be improved by employing stability training. The aim of stability training [25] is to improve robustness against data distortions without compromising classification performance. Instead of using distortion instances in the training data, stability training generates images that are disturbed by Gaussian noise and feeds the images to the network at the same time as the reference samples. The network then has the following task: make the outputs for the disturbed image similar to the outputs of the reference images. This implicitly forces a restriction on the sensitivity of a model to small perturbations in the input data.

Robustness validation features a set of tools aimed at verifying the performance of an AI system. This can include new data environments, evaluating the model under new requirement metrics or exposing it to a stress test for specific edge cases and vulnerabilites. A simple strategy for obtaining a less noisy estimate of an AI system 's predictive performance is so called cross-validation. Under this evaluation regime the available test data from Φ is partitioned into groups. The AI system is then trained on a subset of the groups, alternating the test group for each training run for all possible choices of a test group [26]. A major drawback of this approach is the computational burden it incurs for models with expensive training procedures which oftentimes is the case for deep learning AI systems. In the realm of statistical scholarship hypothesis testing forms an important methodological pillar. Many of the popular data modelling approaches in this field have been studied for decades and, owing to their analytic accessibility, in many cases their behaviors are well understood. The ordinary least squares (OLS) estimator is such a well studied approach which boasts a plethora of tests to better interpret the resulting model. This includes tests for hypotheses on individual regression coefficients, e.g. the so called *t*-test, or linear combinations of hypotheses, e.g. the *F*-test, [27] as well as tests for properties like conditional heteroskedasticity [28] or serial autocorrelation [29,30]. This level of model understanding and interpretation has not carried over to deep learning based AI systems. An important reason for this absence is that deep learning AI systems typically do not lend themselves to the type of analytical treatment that is possible with hallmark approaches from classical statistics. Some theoretically motivated model selection criteria that have been carried over from statistical theory to the deep learning setting include variations on the log evidence like Akaikes's information criterion (AIC) [31], Schwarz's Bayesian information criterion (BIC) [32] or the Occam-weighted likelihood used in Bayesian model selection [33].

Table XX1: Summary overview of robustness enhancing tools per action area

Action areas	Tools
Data fidelity	– Data diversity [6] – Pre-processing (zero centring, PCA [7], whitening [7]) – Normalization (standardization, min-max scaling)
Robust training	– Adversarial training [12] – Generative methods [18,19,20,21,22, 24] – Stability training [25] – FAT optimization objectives [44,45,46,47,48]
Robustness validation	– Cross-validation [26] – Classical tests (e.g. *t*-test, *F*-test [27], serial autocorrelation [29, 30]) – Information criteria (AIC [31], BIC [32], Occam-weighted likelihood [33]) – Vulnerability tests (PGD [12]) – FAT validation metrics [38,39,40,41,42,43] – FAT validation toolboxes [49, 50, 51]

Action areas	Tools
Alarm systems	– Outlier tests (generative methods [52,53], Bayesian uncertainty [62,66]) – Attribution methods (radient*input [54], integrated gradients [55], layerwise relevance propagation [56]/deep Taylor decomposition [57], perturbation-based attribution [58]) – Uncertainty quantification (Gaussian Processes [62], aleatoric uncertainty for deep learning AI systems [67], epistemic uncertainty for deep learning AI systems [66])

Furthermore, adversarial vulnerability tests can be used to simulate attacks on the AI system. There are several attacking strategies developed to pose a threat to a AI system. Almost all of them follow the principle that the classification should be changed with only minimal modification of the input. Projected Gradient Descent (PGD) [12] is in its core the fundamental version of a first-order attack. Other attacking strategies like Carlini-Wagner (CW) [34], Momentum Iterative Method (MIM) [35], Elastic-Net Attack against DNN (EAD) [36] or Fast Gradient Sign Method (FGSM) [37] can be considered to be variations of this attack. Finally, new requirements can be introduced to probe the trained AI system. These new metrics can for example be drawn from insights of so called fair, accountable, transparent (FAT) AI research which offers various approaches to formalizing fairness and biases [38,39,40,41,42,43]. There also exist proposals how to incorporate such FAT measurements in AI system training and applications [44,45,46,47,48] which can be utilized in the robust training action area. Lastly one should note that FAT research has already produced a number of software repositories aimed at benchmarking an algorithm’s susceptibility to problems of bias and fairness, for example AI Fairness 360 [49], Python Fairness Package [50], or TuringBox [51].

Alarm systems are important to monitor the AI system during deployment. Their purpose is to alert an AI system operator when something unusual is happening. Outlier tests are a case in point. Such tests signal when input deviates strongly from the types of input the model has been trained on. Generative models can be used to detect outliers for a given data distribution [52,53]. Any input lying on the manifold of the training data distribution will be given a high score by the generative AI system, as the model has seen data from this manifold during training. Conversely, an input which is very different from the training data will be given a low score, meaning that it is an outlier. Attribution methods can also be used. Attribution methods typically deal with methods that aim to map input features to relevance scores that reflect the features’ contribution to the output of a model. As an alarm system these methods can be used to signal when the model bases its decision on input features very different from the ones a medical expert would use. Popular schemes include gradient*input [54], integrated gradients [55], layerwise relevance propagation [56]/deep Taylor decomposition [57] or perturbation-based attribution [58]. Finally, uncertainty quantification methods may be employed to signal uncertainty in the inputs - so called aleatoric uncertainty – or uncertainty in the model decision - so called epistemic uncertainty – as well as unfamiliar inputs, which is related to epistemic uncertainty. Classic Bayesian modelling with Gaussian Processes provides built in epistemic uncertainty estimates [59,60,61,62,63]. As deep learning AI systems are not as amenable to an analytic treatment as Gaussian Processes numerous approximating treatments have surfaced [64, 65, 66]. A popular epistemic uncertainty quantification scheme for deep learning AI systems, called Monte Carlo dropout, was proposed by [66]. Aleatoric uncertainty quantification for deep learning AI systems has already been sketched out as early as 1994 by [67].

Recommendations

To enhance the robust performance of AI systems in a real world application checks and safety measures, as presented above, should be incorporated at each step of the AI system life cycle.

Below we provide a summarized check list for important tools in each step of the AI system workflow.

Ensuring **data fidelity** can help to enhance the robustness of AI systems. Through pre-processing and normalization, the input data can be brought into a shape that accommodates modelling assumptions such as decorrelated inputs or inputs in a certain value range. Thus, an AI system should include the following data fidelity protocols

– If possible, restrict input data during deployment to be similar to input data during training

– Ensure that input data adheres to the modelling assumptions by e.g. using zero centring, PCA, whitening, standardization or min-max scaling

– Evaluate and ensure the diversity of the data as per the requirements of the specific task, e.g. racial diversity

The robustness of AI systems can be enhanced by employing **robust training** protocols such as adversarial training, generative models or stability training. Thus, it should be ensured the following tools were used

– Adversarial training using PGD attacks to shield against adversarial vulnerability

– Generative models

– Stability training to shield against common data perturbations

AI systems obtained after robust training should also undergo rigorous **robustness validation** before going into deployment. The following tools should be in place for the validation step

– Cross-validation

– If the model allows: hypothesis testing

– Adversarial and perturbation stress tests

– Pending suitably labelled data FAT metrics should be employed to evaluate task specific requirements that go beyond the original optimization objectives

Alarm systems are critical to making sure that failures are sufficiently signalled by the AI system. Any error or malfunction should be caught with the help of available tools. To this end the following should be in place

– Outlier test for inputs

– Include integrated gradients, or another attribution method of your choice, so that a human can verify the basis of the AI system decisions

– Include an uncertainty quantification tool, like Monte Carlo dropout or an aleatoric proxy, so that the AI system’s decision confidences are signalled to the AI system operator

References

1. P. J. Huber. Robust statistics. Wiley Series in Probability and Mathematical Statistics, New York: Wiley, 1981.

2. T. G. Dietterich. Steps toward robust artificial intelligence. AI Magazine, 38(3), 3-24, 2017.

3. Stuart Russell, Daniel Dewey, and Max Tegmark. Research Priorities for Robust and Beneficial Artificial Intelligence. AI Magazine, 36(4):105, December 2015.

4. Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. arXiv preprint arXiv:1903.12261, 2019.

5. Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In Conference on Fairness, Accountability and Transparency, pp. 77-91. 2018.

6. Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumeé III, and Kate Crawford. Datasheets for datasets.arXiv preprint arXiv:1803.09010, 2018.

7. Richard O. Duda, Peter E. Hart, and David G. Stork. Pattern Classification (2Nd Edition). Wiley-Interscience, New York, NY, USA, 2000.

8. Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. arXiv preprint arXiv:1611.01236, 2016.

9. Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. arXiv preprint arXiv:1803.06373, 2018.

10. Xuanqing Liu, Yao Li, Chongruo Wu, and Cho-Jui Hsieh. Adv-bnn: Improved adversarial defense through robust bayesian neural network. arXiv preprint arXiv:1810.01279, 2018.

11. Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. arXiv preprint arXiv:1812.03411, 2018.

12. Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083, 2017.

13. Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th international conference on Machine learning, pages 1096–1103. ACM, 2008.

14. Diederik P. Kingma, Tim Salimans, and Max Welling. Variational Dropout and the Local Reparameterization Trick. CoRR, abs/1506.02557, 2015.

15. Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Advances in neural information processing systems, pages 2672–2680, 2014.

16. Taesup Kim and Yoshua Bengio. Deep directed generative models with energy-based probability estimation. arXiv preprint arXiv:1606.03439, 2016.

17. Rithesh Kumar, Anirudh Goyal, Aaron Courville, and Yoshua Bengio. Maximum entropy generators for energybased models. arXiv preprint arXiv:1901.08508, 2019.

18. Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. In International Conference on Learning Representations, volume 9, 2018.

19. Andrew Ilyas, Ajil Jalal, Eirini Asteri, Constantinos Daskalakis, and Alexandros G Dimakis. The robust manifold defense: Adversarial training using generative models. arXiv preprint arXiv:1712.09196, 2017.

20. Shiwei Shen, Guoqing Jin, Ke Gao, and Yongdong Zhang. Ape-gan: Adversarial perturbation elimination with gan. ICLR Submission, available on OpenReview, 4, 2017.

21. Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. arXiv preprint arXiv:1710.10766, 2017.

22. Lukas Schott, Jonas Rauber, Wieland Brendel, and Matthias Bethge. Robust perception through analysis by synthesis. arXiv preprint arXiv:1805.09190, 2018.

23. Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. arXiv preprint arXiv:1802.00420, 2018.

24. Vignesh Srinivasan, Arturo Marban, Klaus-Robert Müller, Wojciech Samek, and Shinichi Nakajima. Counterstrike: Defending deep learning architectures against adversarial samples by langevin dynamics with supervised denoising autoencoder. arXiv preprint arXiv:1805.12017, 2018.

25. Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow. Improving the Robustness of Deep Neural Networks via Stability Training. arXiv:1604.04326 [cs], April 2016. arXiv: 1604.04326.

26. Christopher M. Bishop. Pattern recognition and machine learning. Information science and statistics. Springer, New York, NY, corrected at 8th printing 2009 edition, 2009. OCLC: 845772798.

27. Fumio Hayashi. Econometrics. Princeton Univ. Press, Princeton, 2000. OCLC: 247253903.

28. Halbert White. A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity. Econometrica, 48(4):817–838, 1980.

29. L. G. Godfrey. Testing Against General Autoregressive and Moving Average Error Models when the Regressors Include Lagged Dependent Variables. Econometrica, 46(6):1293–1301, 1978.

30. T. S. Breusch. Testing for Autocorrelation in Dynamic Linear Models*. Australian Economic Papers, 17(31):334– 355, 1978.

31. H. Akaike. A new look at the statistical model identification. IEEE Transactions on Automatic Control, 19(6):716– 723, December 1974.

32. Gideon Schwarz. Estimating the Dimension of a Model. The Annals of Statistics, 6(2):461–464, 1978.

33. David J. C. MacKay. Bayesian methods for adaptive models. phd, California Institute of Technology, 1992.

34. Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In Security and Privacy (SP), 2017 IEEE Symposium on, pages 39–57. IEEE, 2017.

35. Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. arXiv preprint, 2018.

36. Yash Sharma and Pin-Yu Chen. Breaking the madry defense model with l_1-based adversarial examples. arXiv preprint arXiv:1710.10733, 2017.

37. Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572, 2014.

38. C. Dwork, M. Hardt, T. Pitassi, O. Reingold, & R. Zemel. Fairness through awareness. In Proceedings of the 3rd Innovations in Theoretical Computer Science Conference (pp. 214-226). ACM, 2012.

39. N. Papernot, P. McDaniel, A. Sinha, & M. Wellman. Towards the science of security and privacy in machine learning. Preprint at:arXiv:1611.03814, 2016.

40. P. Gajane, & M. Pechenizkiy. On formalizing fairness in prediction with machine learning. Preprint at:arXiv:1710.03184, 2017.

A. Weller. Challenges for Transparency. CoRR. Preprint at:arXiv:1708.01870v1, 2017.

41. G. Yona & G. Rothblum. Probably Approximately Metric-Fair Learning. In International Conference on Machine Learning (pp. 5666-5674). Retrieved from http://proceedings.mlr.press/v80/yona18a.html, 2018.

42. E. D. Moss. Translation Tutorial: Toward a Theory of Race for Fairness in Machine Learning. In Proceedings of FAT* conference (FAT* Conference). ACM, New York, NY, USA, 2 pages. Retrieved from https://drive.google.com/file/d/1jKSbnAg7NPotjhvNb8o4YXyjcjF8RTJW/view, 2019.

43. S. Shaikh, H. Vishwakarma, S. Mehta, K. R. Varshney, K. N. Ramamurthy, & D. Wei. An End-To-End Machine Learning Pipeline That Ensures Fairness Policies. CoRR, abs/1710.06876. Preprint at:arXiv:1710.06876v1, 2017.

A. Paul, C. Jolley & A. Anthony. Reflecting the Past, Shaping the Future: Making AI Work for International Development. Retrieved from USAID: https://www.usaid.gov/sites/default/files/documents/15396/AI-ML-in-Development.pdf, 2018.

44. R. Dobbe & M. Ames. Translation Tutorial: Values, Reflection and Engagement in Automated Decision-Making. In Proceedings of ACM Conference on Fairness, Accountability, and Transparency (ACM FAT* 2019). ACM, New York, NY, USA, Article 4, 2 pages. Retrieved from https://drive.google.com/file/d/1vhkLBfDUtsd9hXfbXz8QwFo8GYGZyAWP/view, 2019.

A. Albarghouthi, & S. Vinitsky. Fairness-Aware Programming. In FAT* ’19: Conference on Fairness, Accountability, and Transparency, January 29–31, 2019, Atlanta, GA, USA. ACM, New York, NY, USA, 9 pages. Doi:10.1145/3287560.3287588. 2019.

45. K. Holstein, J. W. Vaughan, H. Daumé III, M. Dudík, & H. Wallach. Improving fairness in machine learning systems: What do industry practitioners need?. ACM CHI Conference on Human Factors in Computing Systems (CHI 2019). Preprint at:arXiv:1812.05239v2, 2018.

46. R. K. Bellamy, K. Dey, M. Hind, S. C. Hoffman, S. Houde, K. Kannan, ... & S. Nagar. AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. Preprint at:arXiv:1810.01943v1, 2018.

47. S. A. Friedler, C. Scheidegger, S. Venkatasubramanian, S. Choudhary, E. P. Hamilton, & D. Roth. A comparative study of fairness-enhancing interventions in machine learning. Preprint at:arXiv:1802.04422v1, 2018.

48. Z. Epstein, B. H. Payne, J. H. Shen, C. J. Hong, B. Felbo, A. Dubey, ... & I. Rahwan. TuringBox: An Experimental Platform for the Evaluation of AI Systems. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. (pp. 5826-5828). Retrieved from http://www.fatml.org/media/documents/turing_box.pdf, 2018.

49. Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pages 135–147. ACM, 2017.

50. Nicholas Frosst, Sara Sabour, and Geoffrey Hinton. Darccc: Detecting adversaries by reconstruction from class conditional capsules. arXiv preprint arXiv:1811.06969, 2018.

51. Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning Important Features Through Propagating Activation Differences. In ICML, 2017.

52. Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic Attribution for Deep Networks. In ICML, 2017.

53. Sebastian Bach, Alexander Binder, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. Analyzing Classifiers: Fisher Vectors and Deep Neural Networks. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2912–2920, 2016.

54. Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep Taylor decomposition. Pattern Recognition, 65:211–222, 2017.

55. Matthew D. Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks. In ECCV, 2014.

56. John S. Denker, Daniel B. Schwartz, Ben S. Wittner, Sara A. Solla, Richard E. Howard, Lawrence D. Jackel, and John J. Hopfield. Large Automatic Learning, Rule Extraction, and Generalization. Complex Systems, 1, 1987.

57. Geoffrey E. Hinton and Radford M. Neal. Bayesian Learning for Neural Networks. 1995.

58. Christopher K. I. Williams. Computing with Infinite Networks. In NIPS, 1996.

59. Carl Edward Rasmussen and Christopher K. I. Williams. Gaussian processes for machine learning. Adaptive computation and machine learning. MIT Press, Cambridge, Mass., 3. print edition, 2008. OCLC: 552376743.

60. David Barber and Christopher M Bishop. Ensemble Learning in Bayesian Neural Networks. page 20.

61. Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15:1929–1958, 2014.

62. Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight Uncertainty in Neural Networks. arXiv:1505.05424 [cs, stat], May 2015. arXiv: 1505.05424.

63. Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In ICML, 2016.

64. D. A. Nix and A. S. Weigend. Estimating the mean and variance of the target probability distribution. In Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN’94), volume 1, pages 55–60 vol.1, June 1994.

6.6 Performance

Authors: Federico Cabitza

For task-dependent performance measures please refer to the TDDs of the individual topic groups of FG-AI4H.

Novel Decision-Informed Metrics for Medical AI Validation

Problem

– Many metrics and measures to assess the performance of predictive (classification) models embedded in Medical AI (MAI) exist and they have been proposed before the AI community and the medical community to understand the value of prospective ML-based decision support tools.

– Besides their number, these metrics are either trivial (standard accuracy), or prone to bias/distortion (e.g., class unbalance), or to misunderstanding (e.g., AUROC, logloss).

– Generally speaking, medical doctors don’t understand these measures (besides accuracy). Moreover, these measures are not practice-aware (or informed by medical practice).

– ***In short, we need a simple (one number), intuitive notion of “how good a decision support is”.***

Solution

– Three novel metrics are proposed to evaluate the 'pragmatic validity' of a model w.r.t a Benchmark Test Dataset for a specific discriminative task and these metrics can guarantee merits beyond the typical ' statistical validity' norm used for model validity evaluation.

– It looks as at a data driven approach to assess 'human perceived complexity' and taking it into account for evaluating AI model validity.

– In short, an 'accuracy' metric (H-accuracy), a 'representativeness' metric (Pi-Representativeness) and a 'Robustness' metric (Ratio of H-accuracy and Pi-Representativeness) are proposed.

H-accuracy (Ha) : a novel formulation of accuracy to represent the practical value and to assess the machine classification model with respect to any Benchmark Test Dataset for which we collect some additional information based on true labelling (priority of classes to predict, minimum acceptable confidence, case complexity).

– H-accuracy helps to curb the model drift towards 'over diagnosis'.

  o It can seen as a balanced, class and case weighted expression or measure of accuracy of a machine learning model

  o It is equivalent to 'Standardized Net Benefit' (very important measure to balance costs and benefits of diagnostic tools)measure (Kerr. et al, 2016)

– The main novel contribution of H-accuracy is that it takes into consideration- the complexity of the cases of the test data set. This formulation of accuracy is safe w.r.t to the limitations that affect other common measures of model performance.

– For H-accuracy implementation, annotation should set or code the following 3 types of information:

  a) The threshold of minimum confidence the model should have to provide advice (to penalize right prediction that have low confidence)

  b) The priority of positive class w.r.t negative class (preference between sensitivity and specificity)

c) The complexity of each case in terms of rarity, difficulty, impact of missing, etc

- H-accuracy is completely backward compatible- In the formulation, if one parameter (TAU) is set to 50% then it is equivalent to regular accuracy. Then if 'P' (priority) parameter is set to 50% then it is equivalent to balanced accuracy and if 'complexity' parameter is set to constant, it is then it is equivalent to regular accuracy.
- H-accuracy can be tailored to a specific diagnostic task by tuning the 3 parameters to make it more suitable to the preferences of the domain experts. The parameter configuration can be local (e.g. hospital setting) or for a specialist community, scientific society or association. That sense it can be considered as a parametric version of accuracy

Pi-Representativeness (Pi): a simple and effective way to calculate the representativeness of a dataset (e.g. training dataset) w.r.t another dataset e.g. benchmark dataset representing reference population)

- Pi-Representativeness is an alternative way of comparing data set distributions equality tests like Kolmogorov-Smirnov test
- Pi-Representativeness helps us compute a measure of how similar the training dataset is to the benchmark test dataset
- Pi-Representativeness can be used as a point-of-care interpretation tool to get a measure of robustness because here the accuracy score is normalized with a measure of how fair was the competition.
- Pi-Representativeness is used to understand the extent to which the test data set is representative w.r.t the reference population and here in some way the common biases like gender bias, sampling bias are getting minimized because you are verifying that the datasets used are representative of the reference population

Robustness=(~Ha/Pi): A ratio of H-accuracy and Representativeness as a simple measure of Robustness and generalisability of the model

- If the training set and the benchmark test set are very similar, i.e. Pi=1, accuracy estimates are not reliable and hence then we cannot be sure that the model will have the same performance given a completely different data set
- If the training set and the benchmark test set are significantly different, i.e. Pi=0, then accuracy estimates are reliable and that the model skill will be maintained given new datasets in real-world conditions

References

https://arxiv.org/pdf/1910.09246.pdf

____