ITUPublications

International Telecommunication Union
Standardization Sector

ITU-T Focus Group Deliverable

(03/2023)

Focus Group on Artificial Intelligence for Health (FG-AI4H)

FG-AI4H DEL06

AI training best practices specification

ITU-T FG-AI4H Deliverable DEL06

AI training best practices specification

Summary

Machine learning models for artificial intelligence (AI) in health are deployed in high-impact tasks. As a result, it is important to follow best practices for training and documentation so as to achieve maximum performance and transparency. The first part of this document provides a review of best practices for proper AI model training. The second part of this document provides guidelines for model reporting.

Keywords

AI training; best practices.

Note

This is an informative ITU-T publication. Mandatory provisions, such as those found in ITU-T Recommendations, are outside the scope of this publication. This publication should only be referenced bibliographically in ITU-T Recommendations.

Change Log

This document contains Version 1 of the Deliverable DEL06 on "*AI training best practices specification*" [approved at the ITU-T Focus Group on AI for Health (FG-AI4H) meeting held in 16 March 2023| approved on 16 March 2023 via the online approval process for the ITU-T Focus Group on AI for Health (FG-AI4H)].

Editor: Sim Xinming
AI Singapore
Email: xinming@aisingapore.org

Stefan Winkler
National University of Singapore
Email: winkler@nus.edu.sg

Table of Contents

Page

ITU-T FG-AI4H Deliverable DEL06

AI training best practices specification

1 Scope

This deliverable provides a general introduction and background to model training across a typical AI pipeline. It explains basic key concepts, considerations, practices and limitations on the different aspects of model training in the health-care domain and points readers to cited sources or studies where advanced materials can be found.

2 References

[FG-AI4H DEL0.1] ITU-T FG-AI4H DEL0.1 (2022), *Common unified terms in artificial intelligence for health*.
https://www.itu.int/dms_pub/itu-t/opb/fg/T-FG-AI4H-2022-1-PDF-E.pdf

Additional references are found in the Bibliography of this document.

3 Definitions

3.1 Terms defined elsewhere

The terms defined in [FG-AI4H DEL0.1] are applicable to this Technical Report.

3.2 Terms defined in this Technical Report

This Technical Report defines the following term:

3.2.1 alarm fatigue: Also known as alert fatigue, is the sensory overload and desensitization that occurs when one is exposed to a large number of alerts and alarms in the health care context, leading to missed alarms or delayed response. (Adapted from [b-Woo].)

4 Abbreviations and acronyms

This Technical Report uses the following abbreviations and acronyms:

AI	Artificial Intelligence
AutoML	Automated Machine Learning
EHR	Electronic Health Record
LOCF	Last Observation Carried Forward
MAR	Missing At Random
MCAR	Missing Completely At random
NMAR	Not Missing At Random
RNN	Recurrent Neural Networks

5 Conventions

None.

6 AI pipeline: A brief description

Given the complexities of diseases and their interaction with the human body, AI for health requires a high degree of customization. This document broadly covers the important considerations throughout an AI pipeline highlighted in Figure 1.

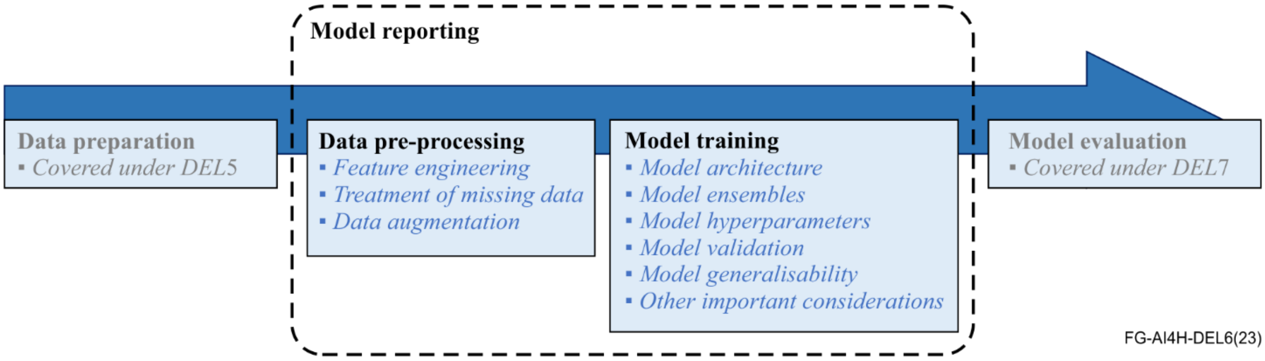

Figure 1 – A brief description of an AI training pipeline

7 Best practices for data pre-processing

Data pre-processing is an important aspect in the AI model training pipeline. Health-care data exist in many forms, such as physicians' notes, patient details, medication and medical images, which are typically consolidated and referred to as an electronic health record (EHR). Within EHR data, the number of domains (e.g., patient demographic, lifestyle, and health-related variable groups) and their individual subdomains (e.g., occupation, smoking habits, exercise frequency and intensity, family history etc.) predictor variables, or features, present are large and complex.

However, not all features are equally significant to the problem statement and certain specific biomarkers hold a greater weightage, or are more clinically meaningful, in determining the outcome of a medical examination, e.g., specific cancer detection. It is imperative to scrutinize datasets, not only through the lens of AI or data scientists, but also through the lens of medical domain experts in the fields around the problem statement. However, employing traditional clinical approaches and models to peruse all available dataset features would be tedious, time-consuming and resource-intensive. How then should the data be explored meticulously to identify and select data of significance?

7.1 Feature engineering

Before embarking on feature engineering, the problem statement must be clearly articulated to understand the desired outcomes. This highlights the necessary features, if present within the dataset, that contribute significantly to the desired outcomes, and elucidates other possible features that could be generated to complement them. As not all features are necessary, careful feature selection, reduction and generation is critical in building an accurate AI model, while saturating the AI model with irrelevant or inconsequential data degrades its performance or exacerbates resource costs. As AI applications proliferate within the health-care space, a shift of approaches that guide feature selection or reduction is seen.

– Traditional approaches. In traditional prediction models, predictor variables are typically predefined [b-Goldstein], such as blood pressure or cholesterol in the Framingham Heart Study [b-Wilson]. In most cases, these features are selected based on expert judgment or knowledge. However, the depth of patient data is not completely utilized, which may explain poorer model performances when compared with newer, automated approaches.

– Semi-automatic approaches. These approaches may consist of two components in varying degree; one which relies on automatic data-driven techniques to select and extract clinically meaningful features from large datasets, while complementing existing predefined features selected manually based on expert judgment or established clinical studies and industry standards. This usually results in an AI model that outperforms traditional approaches [b-Luo].

– Deep learning approaches. Deep learning allows the AI model to execute feature engineering without being explicitly programmed, removing the need to pre-define features. Deep learning AI models are also able to handle large, messy or incomplete datasets sufficiently well. In a study by Google AI on deep learning using EHR data [b-Rajkomar], the utilization of the full spectrum of EHR data using deep learning techniques has been shown to outperform traditional clinical models in prediction tasks such as in-hospital mortality, 30-day unplanned readmission, prolonged length of stay and final discharge diagnoses.

Feature generation is often done in tandem with feature selection. Additional features derived from the dataset is sometimes required to help the AI model learn better. Determining what additional features should or could be generated depends greatly on the problem statement and its relevant medical domain knowledge. For example, when determining the effectiveness of a particular treatment course and recommending treatment options to the health-care professional for patient management, it is necessary to generate and provide AI models with data on historical information and lag features, such as patient health markers over a time period. It is therefore useful to always keep in mind a guiding framework on approaching feature extraction, generation and aggregation [b-Tran] throughout the AI model training pipeline.

7.2 Treatment of missing data

It is unrealistic to assume that EHR data can be readily consumed for model training upon retrieval. It is common to expect missing values within health-care records. How these missing data (e.g., missing true positive alarms) is managed, rationalized, and imputed could drastically impact the final model performance (e.g., resulting in false positives or false negatives).

Missing data could be in statistical terms classified into three main types: (1) missing completely at random (MCAR); (2) missing at random (MAR); and (3) not missing at random (NMAR). The easiest way to handle missing data is to simply ignore and remove either the value or the entire case itself. When doing so an underlying assumption that the missing data is independent of both the observed and unobserved data is made, which may not be the case. Several extensive methods used to address missing data in various health-care contexts can be found in open literature [b-Wong], and are summarized below:

– Last observation carried forward (LOCF). LOCF carries forward observation from the last observed time point to fill the missing data at the endpoints.

– Complete-case analysis and available-case analysis. Complete-case analysis uses only data from patients with a complete record of all visits and ignores all patients with any missing data. Available-case analysis on the other hand uses all available data.

– Mean imputation, hot-deck imputation, regression imputation. Mean imputation uses the mean across all data at the time point to fill in the missing data. Hot-deck imputation fills the missing data with observed responses from another randomly selected but similar data point. Regression imputation uses a regression model to predict and fill the missing value.

– Mixed effects models and generalized estimating equations. For mixed effects models, a statistical distribution (e.g., Gaussian distribution) is used to account for the missing data. On the contrary, an estimating function could also be used to predict the missing data.

– Inference for NMAR data. There are two methods that can be used to infer for NMAR data: pattern mixture models and selection models. Pattern mixture models [b-Little] express the joint distribution of the responses and missing indicators using the distribution of all possible missingness patterns and the distribution of the responses given a specific missingness pattern. Selection models [b-Diggle] express the joint distribution of the responses and the missing indicators using the opposite decomposition; the distribution of the responses and the distribution of the missing indicators given the responses.

7.3 Data augmentation

Data augmentation is another important component of data pre-processing and has slight overlaps with feature generation. It is often used for computer vision applications when image data is limited or unbalanced (e.g., the proportion of a certain data class is much higher or lower than the others), such as medical imaging and diagnosis applications. Data augmentation generates additional training data to complement existing datasets, providing the training process with more information to learn from, and boosting the model's accuracy, training stability or generalizability.

The types of data augmentation techniques that could be deployed depend on specific use cases. Below are some basic techniques frequently deployed for image-based AI models, as well as references to more advanced techniques [b-Shorten].

– Geometric transformations. Techniques such as image cropping, zooming, rotating and translating are commonly used to create synthetic images to complement existing datasets. This helps expand dataset size while preserving authenticity and may help to reduce positional biases during model training.

– Colour space transformations. Poor image lighting or colour balance may affect a model's ability to identify features during model training, which typically happens for medical imaging applications. Techniques such as equalizing or normalizing colour histograms often helps to improve model performance. However, caution must be exercised to ensure any colour manipulations do not affect key definitions within the image data that characterizes these features.

Data augmentation is less frequently deployed for text-based AI models than for image-based ones due to a few factors. First, it is much easier to acquire large text-based data than image-based data, hence it less likely to need to overcome challenges such as limited data or class imbalance. Second, text-based data is much more complex due to the intricacies of the language medium. The resulting sentence sentiment, position and nuance could be drastically altered if one word is changed incorrectly. Yet, the benefits of data augmentation seen for image-based models has been recently demonstrated if they are used correctly for text-based models. Recent studies include easy data augmentation (EDA) techniques [b-Wei] that could be deployed for text-based deep learning models, such as synonym replacement, random insertion, random swap and random deletion.

8 Best practices for model training

Model training is not a linear process. Many iterations are required to properly train and optimize before the model sufficiently performs its intended functions. The international research community has explored countless aspects of model training over the past decades, but what is important to note is that the benefits of these developed techniques, tools, takeaways and conclusions from various studies are very dependent on the use cases and settings, context or environment they are applied to. As such, this section will only broadly focus on common basic best practices that help to bring a potential user up to speed on how to train and deploy an AI model.

8.1 Model architecture

The wide spectrum of AI models available is daunting, and the complexity, number and types of models will only proliferate further over time. It would be laborious for users with little background in AI to even begin to understand which types of models to use for a particular use case, much less evaluate their performance.

For experimentation purposes, it is advisable to start with AI models readily available and updated on open-source libraries (e.g., ResNet-152, GoogLe-Net19, BERT) that have been trained on similar datasets to achieve similar objectives, and conduct transfer-learning techniques to adapt them to the defined use case. A survey of open-sourced deep learning architectures for various use cases provides some basic evaluations of each architecture, their strengths and their limitations [b-Alom]. Additionally, for more advanced models, there have been recent works summarizing different neural network architectures and their limitations for the health-care domain based on their general use cases for bioinformatics, medical informatics, medical imaging and public health [b-Zion].

While most of this document focuses on the basic application of AI, it is important to consider throughout the AI pipeline the scalability implications of factors such as model architecture. Where scalability becomes an issue, it may become necessary to adopt advanced techniques that could help to reduce model complexity while preserving model fidelity [b-Xu-1].

8.2 Model ensembles

It is a common misconception that AI applications are one singular AI model working behind the scenes. Experimentation during the model training pipeline will show that some models are inherently better at certain tasks than others. Each model has its own strengths and weaknesses. It is difficult to optimize a model that is good at everything, as this is usually limited by either an unfeasibly large amount of data required for it to learn from, or an unrealistic amount of time needed to train it.

As such, the principle of model ensembles is to build multiple, smaller, implementable models and combine them together based on a set of decision-making or learning strategies. Multimodel ensembles have been demonstrated to achieve equal, if not better, performance than individual models for the detection of cancer [b-Xiao] and heart disease [b-Mienye] and for diabetes prediction [b-Akula]. Here, basic decision-making techniques for model ensembles are summarized, and reference is made to more advanced techniques on ensemble learning [b-Dong].

- Maximum or majority vote. Maximum or majority vote is employed by counting the prediction of each model. The prediction with the highest or majority vote will be used as the eventual ensemble prediction.
- Average or weighted vote. Average or weighted vote is employed by summing each models' output probabilities and averaging them either uniformly (average vote) or based on a defined weightage (weighted vote). The output class with the highest probability will be used as the eventual ensemble prediction.

The type of ensemble methods to be used depends on factors such as model learning characteristics and domain knowledge. A study comparing the effectiveness of various ensemble methods and summarizing the strengths and weaknesses of each type can be found in Ref. [b-Ju].

8.3 Model hyperparameters

Hyperparameters are a set of parameters that cannot be learned by the model, and whose values are specified before model training begins to control the training process, such as learning rates, dropout rates, batch sizes and epochs. Control over these hyperparameters throughout the model training process is crucial to ensuring model accuracy, generalizability and training stability (convergence). For example, high values for the learning rate at the start of the model process may lead to divergence, while low values of the learning rate increase the resource cost of model training. As such, it is common to see learning rate annealing techniques being used for better control over the training process.

Optimization of these hyperparameters ensures that the resultant model optimally minimizes the predefined loss function for a given set of data. For professionals with little or no experience in machine learning, the process of optimizing these hyperparameters may be challenging. Furthermore, it becomes increasingly resource-intensive to carry out hyperparameter optimization as the datasets increase in volume and complexity. Here some common manual and automated techniques used for hyperparameter optimization which could be used are discussed.

- Grid search. Grid search conducts an exhaustive search of a manually specified subset of the hyperparameter space. This is one of the simplest techniques and is easy to implement. However, it spends substantial time evaluating hyperparameter combinations which are unpromising and is therefore less efficient that random Search.
- Random search. Like grid search, random search evaluates random combinations of hyperparameters within the search space and within a defined number of iterations. As not all combinations are evaluated over the search space, this makes hyperparameter optimization computationally less demanding when optimizing for higher dimensions of hyperparameters as compared with Grid Search.
- Bayesian optimization. Bayesian optimization is a commonly adopted optimization framework for many automated machine learning (AutoML) systems [b-Waring]. It is an approach to optimizing objective functions that take a long time (minutes or hours) to evaluate. It is best suited for optimization over continuous domains of less than 20 dimensions and tolerates stochastic noise in function evaluations. It builds a surrogate for the objective and quantifies the uncertainty in that surrogate using a Bayesian machine learning technique, Gaussian process regression, and then uses an acquisition function defined from this surrogate to decide where to sample [b-Frazier]. The difference between Bayesian optimization methods from grid search or random search is that it uses past evaluation results to determine the subsequent values to evaluate. In essence this makes the optimization process more efficient as it limits iterations with poor hyperparameter combinations while focusing on promising hyperparameter combinations obtained from past results.

8.4 Model validation

Validating the AI model is a crucial step within the model training pipeline and occurs as the model trains. In this clause, discusses knowing how well the model is performing to its intended objective based on a dataset that has not yet been seen by the model. It gives an indication of the model's performance (how accurate are these decisions based on the input dataset?) and generalizability (how accurate are these decisions based on another unseen dataset?). To do so, instances of overfitting or underfitting within the AI model must be considered.

Overfitting occurs when the AI model or algorithm fits the data too well and unnecessarily captures the noise present within which affects the performance of the model when generalizing to data beyond that used in its training. Underfitting, on the contrary, is when the AI model or algorithm cannot capture the trends or features of the data well enough. Both overfitting and underfitting result in poor model performance and could occur due to factors related to model parameters/weights, algorithm and model selection.

- Selection of model or algorithms. It is impossible to represent a sine function well with a linear equation. Similarly, ensuring that the complexity of the model is befitting of the dataset affects how good the model outputs will be. In general, using models or algorithms which are overly restrictive limits the learning of the dataset, and is likely to lead to an underfit, while using models or algorithms which are overly complex captures unnecessary randomness within the training data and is likely to produce an overfit.
- Recognizing the warning signs. Knowing when the data is overfitted or underfitted is crucial in addressing the issue. Often, an overfit more frequent than an underfit. There are two quick ways to identify overfit-underfit:
  - Bias-variance trade-off is one of the ways to know when the best fit can be achieved for a given setting. Bias is an error due to erroneous assumptions and is an indication of a possible underfit. Conversely, variance refers to the model or algorithm's sensitivity to perturbations within the datasets. High variance generally occurs when the model or

algorithm is overly complex and captures the randomness with the sample, giving an indication of a possible overfit. To ensure a well- performing model, one needs to find the right balance between bias and variance which minimizes the total error function.

  - Training and validation performance plots over the training cycles are another common indicator to look at to determine whether an AI model is overfitting its data. The performance (accuracy or loss) plots indicate how well the model is performing at that given training cycle. One clear indication of overfitting is when the validation performance starts to deteriorate (accuracy decreases or loss increases) and deviate from its converged training performance over the number of training cycles.

- Cross-validation to prevent overfits. While there are other methods which could detect and prevent overfitting such as back-testing or regularization, one of the more popular methods is cross-validation. Though there are many different variations of cross-validation methods such as k-fold or stratified k-fold cross-validation, the underlying principles are the same, which are to partition the data into:

  1) training subset; and

  2) validation subset. At any iteration, the model only trains and tunes on the training subset. The validation on the validation subset gives an indication of the performance on unseen (test) data. The final evaluation of the model on the completely unseen test dataset gives an unbiased assessment of its performance. If the model outperforms on the training data subset compared with the test set, it is likely to be an indicator that the model is producing an overfit.

- Number of model parameters/weights. The number of parameters affects the resulting performance of the model. Having too few leads to underfitting, while conversely having too many may lead to an overfit. It is recommended for parameter optimization to be carried out to determine what is the optimal number to be used to train the model. Underfitting can be resolved by increasing the number of parameters or using weaker regularization during model training.

Overfitting is often encountered during model training, compared to underfitting. Underfitting is generally an easier issue to tackle with simple solutions such as increasing model complexity, increasing model training over more epochs. However, tackling overfitting requires a greater depth of understanding in both the dataset and the model algorithms. Below are a few fundamental strategies that could be employed, as well as references to more advanced techniques [b-Moradi].

- Simplifying model complexity. Neural networks can sometimes be prone to overfitting. While there is no single hard rule or guideline to follow, in instances of an overfit, one should aim to reduce the complexity of the network, such as reducing the number of hidden layers or reducing the number of neurons.

- Employing regularization techniques. There are many different types of regularization techniques developed for different types of AI models, but their main objective is to limit the capacity of the AI by penalizing the model training loss functions (e.g., L1 or L2 regularization) or by reducing the complexity of the algorithm through dropouts [b-Srivastava]. Recent work around some common and novel regularization techniques for computer vision applications and their effectiveness can also be found here [b-Xu-2].

- Early stopping. As discussed in the previous section, one of the common indicators of overfitting is when the validation performance deteriorates and deviates from the converged training performance. As such, early stopping of model training (or rolling back of model weights to a saved point before overfitting occurs) is a strategy that is always used due to its simplicity.

- Data augmentation. Highly unbalanced datasets with disproportionate distribution of classes (e.g., in applications where patients at high risk of a disease belongs to a minority class) should be carefully managed to prevent overfitting. One common best practice is to increase the training samples through data augmentation.

8.5 Model generalizability

So far, the various best practices for model training have been discussed. As model training is a highly iterative process, there is the need to identify useful and significant metrics to evaluate its performance besides model accuracy, i.e., accuracy of the model predictions with the ground truth. In the health-care context, due to complexities across populations (e.g., some population groups are more susceptible to certain types of diseases), disease progressions (e.g., differing disease progression across various geographical regions) and health-care infrastructure among other factors, it is also critical to test the AI model on its generalizability across various factors before deployment and address them accordingly in the training process.

The generalizability of AI models has always been a key challenge in the community, and many are far from achieving reliable generalizability. In the health-care context, AI models are relatively constrained in their deployment to populations and settings to those similar to those they were trained on. For example, diseases may manifest themselves slightly differently across ethnicities. What happens if the AI models were trained on a particular ethnic distribution but tested on a different one? While there has been research showing some success in developing limited generalizable AI models across ethnic distributions [b-Bellemo], this remains a key concern for health-care professionals in ensuring that the diagnosis made by the AI model is as accurate as it is expected to be.

- Generalizability across different populations. Several measures have been proposed to help overcome the issue of AI generalizability across different populations [b-Kelly]:
  - Site-specific training to adapt an existing system for a new population, particularly for complex tasks such as EHR predictions. Methods to detect out-of-distribution inputs [b-Liang], [b-Hendrycks] provide a reliable measure of model confidence to prevent clinical decisions being made on inaccurate model outputs. For medical image classification, this problem may overcome by the curation of large, heterogenous, multi-centre datasets. However, the limitation to this method is that retraining becomes computationally demanding and expensive should the number of out-of-distribution cases far exceed a certain threshold.
- Generalizability across different settings. Health-care institutions around the world vary in their protocols, hardware such as medical equipment and medical equipment software. This results in vastly different health-care data being generated by each unique health-care institution, which makes it challenging to generalize AI systems from one institution to another. While there have been efforts by organizations such as Observational Health Data Sciences and Informatics (OHDSI) to standardize a common data model called the Observational Medical Outcomes Partnership (OMOP) for electronic health-care recording, this remains a barrier to the interoperability of AI models across institutional and international boundaries.
  - One possible way of overcoming this issue lies with the design of the AI model and training. Splitting a disease diagnosis step into segmentation and measurement may enable easier generalization to new imaging hardware by retraining only the segmentation model, which is more data- efficient [b-De Fauw].
- Generalizability across time. Besides the issue of adapting an AI system to new populations, one must also consider the temporality and everchanging nature of diseases. Diseases are always progressing and changing over time in a non-deterministic way [b-Miotto]. However, many existing deep learning models, including those already proposed in the medical domain, assume static vector-based inputs, which cannot handle the time factor in a natural way.

- To address this, it is recommended to design deep learning approaches that can handle temporal health care data. In this aspect, recurrent neural networks (RNN) are a good choice when health-care data is sequentially ordered, as they are well designed to handle temporal dependencies [b-Schickel].

8.6 Other important considerations

Beyond that however, there are other important considerations that should be considered as well throughout the model training pipeline.

– Trust in AI model and systems. Deep learning has been extensively explored in the medical imaging space; however, not many AI systems have been deployed in the real-world context. Despite showing high performance during model evaluations, a lack of trust by clinicians towards these systems still exists. More recent works documenting proposals to address this include using techniques such as incorporating uncertainty modelling into the AI system [b-Lim] so that clinicians are better informed of how certain the AI system is when deriving a certain decision outcome. Other techniques include using frameworks to provide patient-level interpretation derived from AI model features to help with patient management, as well as feature-level interpretation derived from aspects within the datasets, which could help with medical research [b-Zheng].

– Model validation source. Not all data are created equal. How a model was validated may be a factor to consider in the health-care context. Health-care data from prospective studies generally have fewer biases and are more favourable for model validation than that from retrospective studies. This, however, remains a challenge, as the number of established prospective studies are limited. Models that are trained and validated from clinical studies spanning multiple centres are also more likely to be generalizable, allowing the model to be adopted for a different population group significant model retraining effort.

– Workflow integration and implementation. Developing a superior but overly complex AI model is worthless if it does not fit well within a clinician's workflow and eventually ends up being side-lined, i.e., requiring additional user interface, software or hardware to integrate the AI solution within the hospital. If the AI model excessively prompts the clinician with alarms, it may lead to alarm fatigue, leading to these alerts being ignored or switched off.

– Clinical applicability and impact. AI models are typically assessed against a baseline performance. However, for the health-care context, such systems should be evaluated against a human clinician to determine if it does indeed result in an impact to the level of care received by a patient. Even so, a well performing model may not necessarily translate to an overall higher clinical impact. The system must be assessed prospectively in practice to determine the level of impact it can deliver to clinicians and patients.

While there are many other important aspects that need to be considered, such as model privacy, security and especially ethical considerations, these will not be discussed in detail here.

8.7 Model reporting

Given the pipeline for AI model training described in clause 7, responsible development protocols dictate that proper recording and accounting framework be in place to ensure that the AI model is used within its boundaries and limitations. This ensures that users fully appreciate the AI models' purpose, performance and, in particular, its limitations, so that they are appropriately deployed.

Proper accounting and reporting procedures are critical for the deployment of AI models in health care, where model failures or misuse can lead to serious ramifications. As best practice, this should be done to encompass important information throughout the AI pipeline, rather than covering only model training.

This section describes a framework based on model cards [b-Mitchell] proposed recently to encourage transparent reporting about a trained machine learning model. The purpose of this reporting framework is to enable those considering the use of a specific trained model in a particular context to better understand the systematic impacts of the model before deploying it. A standardized reporting format not only makes it easier for different stakeholders to assess and compare candidate models but also encourages forward-looking model analysis in the development phase. An extract from the paper is summarized below:

- Model details. Basic information about the model, e.g., persons and/or organization developing the model, funding, model development date, model version (and change log), model type, model outputs, sources and references, citation details, licence details and feedback channels.
- User case and users. Use cases that were envisioned during development, e.g., primary intended uses, primary intended users as well as out-of-scope uses.
- Factors. Factors could include demographic or phenotypic groups, environmental conditions, technical attributes or others.
- Metrics. Metrics should be chosen to reflect potential real-world impacts of the model, e.g., model performance measures, decision thresholds and approaches towards uncertainty and variation.
- Training and evaluation data. Details on the dataset(s) used for the quantitative analyses, e.g., datasets, motivations and pre-processing.
- Quantitative analyses. This provides the analyses and results of the model evaluation according to the metrics, and its corresponding confidence intervals, e.g., unitary analyses and intersectional analyses.
- Ethical considerations. Provide information on ethical considerations that informed the model development, surfacing ethical challenges and solutions to stakeholders on issues such as data, risk management, risks and harms, use cases and the review processes.

Bibliography

[b-Akula] Akula, R., Nguyen, N., and Garibay, I. (2019), *Supervised Machine Learning based Ensemble Model for Accurate Prediction of Type 2 Diabetes*, in 2019 SoutheastCon (pp. 1-8).

[b-Alom] Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., and Asari, V. K. (2019), *A State-of-the-art Survey on Deep Learning Theory and Architectures*, Electronics, Vol. 8(3), p. 292.

[b-Bellemo] Bellemo, V., Lim, Z. W., Lim, G., Nguyen, Q. D., Xie, Y., Yip, M. Y., and Ting, D. S. (2019), *Artificial intelligence using deep learning to screen for referable and vision-threatening diabetic retinopathy in Africa: a clinical validation study*, The Lancet Digital Health, Vol. 1, No. 1, pp. e35-e44.

[b-De Fauw] De Fauw, J., Ledsam, J. R., Romera-Paredes, B., Nikolov, S., Tomasev, N., Blackwell, S., and Ronneberger, O. (2018), *Clinically Applicable Deep Learning for Diagnosis and Referral in Retinal Disease*, Nature Medicine, Vol. 24, No. 9, pp. 1342-1350.

[b-Diggle] Diggle, P., and Kenward, M. G. (1994), *Informative Drop-out in Longitudinal Data Analysis*, Journal of the Royal Statistical Society: Series C (Applied Statistics), Vol. 43, No. 1, pp. 49-73.

[b-Dong] Dong, X., Yu, Z., Cao, W., Shi, Y., and Ma, Q. (2020), *A Survey on Ensemble Learning*, Frontiers of Computer Science Vol. 14, pp. 241-258.

[b-Frazier] Frazier, P. I. (2018), *A Tutorial on Bayesian Optimization*, arXiv preprint arXiv:1807.02811.

[b-Goldstein] Goldstein, B.A., Navar, A.M., Pencina, M.J., and Ioannidis, J. (2017), *Opportunities and challenges in developing risk prediction models with electronic health records data: a systematic review*, Journal of the American Medical Informatics Association, Vol. 24, No. 1, pp. 198-208.

[b-Hendrycks] Hendrycks, D., and Gimpel, K. (2016), *A Baseline for Detecting Misclassified and Out-of-distribution Examples in Neural Networks*, arXiv preprint arXiv:1610.02136.

[b-Ju] Ju, C., Bibaut, A., and van der Laan, M. (2018), *The Relative Performance of Ensemble Methods with Deep Convolutional Neural Networks for Image Classification*, Journal of Applied Statistics, Vol. 45, No. 15), pp. 2800-2818.

[b-Kelly] Kelly, C. J., Karthikesalingam, A., Suleyman, M., Corrado, G., and King, D. (2019), *Key challenges for delivering clinical impact with artificial intelligence*, BMC medicine, Vol. 17(1), p. 195.

[b-Liang] Liang, S., Li, Y., and Srikant, R. (2017), *Principled Detection of Out-of-distribution Examples in Neural Networks*, arXiv preprint arXiv:1706.02690, pp. 655-662.

[b-Lim] Lim, Z. W., Lee, M. L., Hsu, W., and Wong, T. Y. (2019), *Building Trust in Deep Learning System towards Automated Disease Detection*, in Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, pp. 9516-9521.

[b-Little] Little, R. J. (1993), *Pattern-mixture Models for Multivariate Incomplete Data*, Journal of the American Statistical Association, Vol. 88, No. 421, pp. 125-134.

[b-Luo] Luo, G. (2019), *A Roadmap for Semi-automatically Extracting Predictive and Clinically Meaningful Temporal Features from Medical Data for Predictive Modeling*, Global Transitions, Vol. 1, pp. 61-82.

[b-Mienye] Mienye, I. D., Sun, Y., and Wang, Z. (2020), *An Improved Ensemble Learning Approach for the Prediction of Heart Disease Risk*, Informatics in Medicine Unlocked, Vol. 20, 100402.

[b-Miotto] Miotto, R., Wang, F., Wang, S., Jiang, X., and Dudley, J. T. (2018), *Deep Learning for Healthcare: Review, Opportunities and Challenges*, Briefings in Bioinformatics, Vol. 19, No. 6, pp. 1236-1246.

[b-Mitchell] Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., and Gebru, T. (2019), *Model Cards for Model Reporting*, in Proceedings of the Conference on Fairness, Accountability, and Transparency, pp. 220-229.

[b-Moradi] Moradi, R., Berangi, R., and Minaei, B. (2019), *A survey of regularization strategies for deep models*, Artificial Intelligence Review, Vol. 53, pp. 3947-3986.

[b-Rajkomar] Rajkomar, A., Oren, E., Chen, K., Dai, A.M., Hajaj, N., Hardt, M., and Dean, J. (2018), *Scalable and Accurate Deep Learning with Electronic Health Records*, NPJ Digital Medicine, Vol. 1, No. 1, pp. 18.

[b-Schickel] Shickel, B., Tighe, P. J., Bihorac, A., and Rashidi, P. (2017), *Deep EHR: A Survey of Recent Advances in Deep Learning Techniques for Electronic Health Record (EHR) Analysis*, IEEE Journal of Biomedical and Health Informatics, Vol. 22, No. 5, pp. 1589-1604.

[b-Shorten] Shorten, C., and Khoshgoftaar, T. M. (2019), *A Survey on Image Data Augmentation for Deep Learning*, Journal of Big Data, Vol. 6, No. 1, pp. 60.

[b-Srivastava] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014), Dropout: *A Simple Way to Prevent Neural Networks from Overfitting*, The Journal of Machine Learning Research, Vol. 15(1), pp. 1929-1958.

[b-Tran] Tran, T., Luo, W., Phung, D., Gupta, S., Rana, S., Kennedy, R.L., and Venkatesh, S. (2014), *A Framework for Feature Extraction from Hospital Medical Data with Applications in Risk Prediction*, BMC Bioinformatics, Vol. 15, No. 1, p. 425.

[b-Waring] Waring, J., Lindvall, C., and Umeton, R. (2020), *Automated Machine Learning: Review of the State-of-the-art and Opportunities for Healthcare*, Artificial Intelligence in Medicine, Vol. 104, 101822.

[b-Wei] Wei, J., and Zou, K. (2019), *Eda: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks*, arXiv preprint arXiv:1901.11196.

[b-Wilson] Wilson, P.W., D'Agostino, R.B., Levy, D., Belanger, A.M., Silbershatz, H., and Kannel, W.B. (1998), *Prediction of Coronary Heart Disease using Risk Factor categories*, Circulation, Vol. 97 No. 18, pp. 1837-1847.

[b-Wong] Wong, W. K., Boscardin, W. J., Postlethwaite, A. E., and Furst, D. E. (2011), *Handling Missing Data Issues in Clinical Trials for Rheumatic Diseases*, Contemporary Clinical Rrials, Vol. 32, No. 1, pp. 1-9.

[b-Woo] Woo M, Bacon O (2020), Alarm Fatigue, in Hall KK, Shoemaker-Hunt S, Hoffman L, et al. *Making Healthcare Safer III: A Critical Analysis of Existing and Emerging Patient Safety Practices*, Rockville (MD): Agency for Healthcare Research and Quality (US). https://www.ncbi.nlm.nih.gov/books/NBK555522/

[b-Xiao] Xiao, Y., Wu, J., Lin, Z., and Zhao, X. (2018), *A Deep Learning-based Multi-Model Ensemble Method for Cancer Prediction*, Computer Methods and Programs in Biomedicine, Vol. 153, pp. 1-9.

[b-Xu-1] Xu, D., Lee, M. L., and Hsu, W. (2019), *Propagation Mechanism for Deep and Wide Neural Networks*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 9220-9228).

[b-Xu-1] Xu, D., Lee, M. L., and Hsu, W. (2019), *Patch-Level Regularizer for Convolutional Neural Network*, in 2019 IEEE International Conference on Image Processing (ICIP), pp. 3232-3236.

[b-Zheng] Zheng, K., Cai, S., Chua, H. R., Wang, W., Ngiam, K. Y., and Ooi, B. C. (2020), *TRACER: A Framework for Facilitating Accurate and Interpretable Analytics for High Stakes Applications*, in Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, pp. 1747-1763.

[b-Zion] Zion, I., Ozuomba, S., and Asuquo, P. (2020), *An Overview of Neural Network Architectures for Healthcare*, in 2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS) pp. 1-8.