|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ITU Logo | INTERNATIONAL TELECOMMUNICATION UNION  **TELECOMMUNICATION STANDARDIZATION SECTOR**  STUDY PERIOD 2017-2020 | | FG-AI4H-I-032 | |
| **ITU-T Focus Group on AI for Health** | |
| **Original: English** | |
| **WG(s):** | | Plenary | E-meeting, 7-8 May 2020 | |
| **DOCUMENT** | | | | |
| **Source:** | | Editors | | |
| **Title:** | | DEL06: AI training best practices specification | | |
| **Purpose:** | | Discussion | | |
| **Contact:** | | Sim Xinming AI Singapore | | Email: [xinming@aisingapore.org](mailto:xinming@aisingapore.org) |
| **Contact:** | | Stefan Winkler AI Singapore | | Email: [stefan@aisingapore.org](mailto:stefan@aisingapore.org) |

|  |  |
| --- | --- |
| **Abstract:** | Machine learning models for 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 documentprovides a review of best practices for proper AI model training. The second part of this document provides guidelines for model reporting. |

Contents

[1. AI Pipeline: A Brief Description 3](#_Toc39261201)

[2. Best Practices for Proper Model Training 3](#_Toc39261202)

[2.1. Selection of AI Models 3](#_Toc39261203)

[2.2. Addressing Underfit and Overfit 3](#_Toc39261204)

[2.3. Hyperparameter Optimization 4](#_Toc39261205)

[2.4. Treatment of Missing Data 5](#_Toc39261206)

[2.5. AI Model Generalisability 6](#_Toc39261207)

[3. Model reporting framework 7](#_Toc39261208)

[3.1. Basic Model Details 7](#_Toc39261209)

[3.2. Use Case and Users 8](#_Toc39261210)

[3.3. Model Performance Metrics 8](#_Toc39261211)

[3.4. Model Performance Factors 8](#_Toc39261212)

[3.5. Data 9](#_Toc39261213)

[3.6. Analysis 9](#_Toc39261214)

[3.7. Ethical Considerations 9](#_Toc39261215)

[4. Other References 10](#_Toc39261216)

1. AI pipeline: A brief description

While there are many different aspects to consider throughout the AI pipeline, we will be focusing on the few points highlighted in Figure 1 below for this paper.



Figure 1: A brief description of an AI pipeline.

1. Best practices for proper model training

Machine learning for healthcare is one that requires a high level of customisation, given the complexities of diseases as well as the human body. As with all types of machine learning applications, proper training needs to be ensured for the model to give accurate and high-performance results.

* 1. Selection of AI models

TBD

* 1. Addressing underfit and overfit

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 generalising 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 poor overfitting and underfitting results in poor performance of the AI model. Both of these could happen due to factors such as model parameters/weights, algorithm and model selection.

Selection of model or algorithms. It is important to ensure that the complexity of the model is befitting of the dataset size. In general, models or algorithms which are overly restrictive limits the learning of the dataset, and is likely to lead to an underfit, while models or algorithms which are overly complex captures unnecessary randomness within the training data and is likely to product an overfit. 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. This could be achieved by either reducing the number of hidden layers, reducing the number of neurons or adding dropout layers within the hidden layers.

Recognizing the warning signs. Recognizing when the data is overfitted or underfitted is crucial in addressing the issue, and more often than not, an overfit is typically the case compared to an underfit. One of the ways to know when the best fit can be achieved for a given setting is by analysing the bias-variance trade-off. 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 right balance between bias and variance which minimizes the total error function.

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 is 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 to 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.

* 1. Hyperparameter optimization

Hyperparameter are a set of parameters that cannot be learnt by the model, and whose values are specified before the machine learning process begins. They are used to control the model learning process, such as learning rates, dropout rates, batch sizes, epochs etc. Specific hyperparameters may also lead to possible overfits. For instance, overfitting could occur if the learning rate (LR) is too small. However, this is a trade-off to balance, as having large learning rates may run the risk of divergence in the model training phase.

Optimization of these hyperparameters ensures that the resultant model optimally minimises the predefined loss function for a given set of data. For healthcare professionals with little or no experience in machine learning, the process of optimizing these hyperparameters may be challenging. Here we discuss some common manual and automated techniques used for hyperparameter optimization which could be used.

Grid Search. An exhaustive search of a manually specified subset of the hyperparameter space is done. This is one of the simplest techniques and is easy to implement. However, because it explores the space exhaustively, it spends substantial time evaluating hyperparameter combinations which are unpromising and is therefore less efficient that Random Search. At greater dimensions of hyperparameters, the number of function evaluations grows exponentially, which inevitable makes this technique computationally expensive.

Random Search. Similar to the Grid Search, Random Search evaluates random combinations of hyperparameters within the search space and defined number of iterations and is also relatively easy to implement. 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 to Grid Search.

Bayesian Optimization. Bayesian optimization 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[[1]](#footnote-1). 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.

* 1. Treatment of missing data

As with most publicly collected datasets, it is not realistic to assume that the data comes in a form which can be readily consumed by the AI model. In many cases, it is common to expect missing data within healthcare records. How the missing data (e.g. missing true positive alarms) is managed or rationalized could drastically impact the final model performance (e.g. resulting in false positives or false negatives), so this is of critical importance.

Missing data could be classified into three 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 of course to simply ignore and remove either the value or the entire case itself. When doing so an underlying assumption is made, which is that the missing data is independent of both the observed and unobserved data, and this may not be the case in reality. Wong et al (2010) have summarized several extensive methods below which could be used to address missing data in various healthcare context[[2]](#footnote-2), such as:

* Last observation carried forward (LOCF) and intention to treat (ITT) analysis.
* Complete-case analysis and available-case analysis
* Mean imputation, hot-deck imputation, regression imputation
* Mixed effects models
* Generalized estimating equations
* Inference for NMAR data
  1. AI model generalisability

The generalisability of AI models has always been a key challenge in the community, and many are far from achieving reliable generalisability. In the healthcare context, AI models are relatively constrained in their deployment to populations and settings similar to those they were trained on. For example, diseases may manifest itself slightly different across ethnicity. What happens if the AI models were trained on a particular ethnic distribution and tested on a different one? While there has been research showing some success in developing limited generalisable AI models across ethnic distributions[[3]](#footnote-3), this remains a key concern for healthcare professionals in ensuring that the diagnosis made by the AI model is as accurate as it is expected to be.

Generalisability across different populations. Kelly et al (2019)[[4]](#footnote-4) proposed several measures which could help to overcome the issue of AI generalisability across different populations:

* Site-specific training to adapt an existing system for a new population, particularly for complex tasks like EHR predictions. Methods to detect out-of-distribution inputs[[5]](#footnote-5),[[6]](#footnote-6) and 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 re-training becomes computationally demanding and expensive should the number of out-of-distribution cases far exceed a certain threshold.

Generalisability across different settings. Healthcare institutions around the world vary in their protocols, hardware such as medical equipment and medical equipment software. This results in vastly different healthcare data generated by each unique healthcare institution, which makes it a nightmare to generalise AI system from one institution to another. While there have been efforts by organisations such as Observational Health Data Sciences and Informatics (OHDSI) to standardize a common data model called Observational Medical Outcomes Partnership (OMOP) for electronic healthcare 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[[7]](#footnote-7).

Generalisability across time. Besides the issue of adapting an AI system to new populations, one has to also consider the temporality and everchanging nature of diseases. In a study on Deep Learning for Healthcare, Miotto et al (2018)[[8]](#footnote-8) highlighted that diseases are always progressing and changing over time in a nondeterministic way. 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) is a good choice to utilize when healthcare data is sequentially ordered, as they are well designed to handle temporal dependencies[[9]](#footnote-9).

1. Model reporting framework

This section describes a framework based on the “model cards” proposed by Mitchell et al. (2019)[[10]](#footnote-10) 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. Such a reporting framework is particularly important in the healthcare context, where model failures or misuse can have serious repercussions

When preparing the model description, it is important to consider the different stakeholders that may rely on this information. These include other model developers, software engineers, executives, policymakers, and the general public.

The following subsections discuss various aspects of a model that should be addressed in the model reporting framework. Note that there will be cases where some of this information may be sensitive; for e.g., the amount of detail a company can disclose might be different from academic research groups.

* 1. Basic model details
* **Persons and/or organization developing model**: Who developed the model? This can be used by all stakeholders to infer details pertaining to model development and potential conflicts of interest.
* **Funding**: Who funded the development of the model?
* **Model date**: When was the model developed? This is useful for all stakeholders to become further informed on what techniques and data sources were likely available then.
* **Model version**: Which version of the model is it, and how does it differ from previous versions? This is useful for all stakeholders to track whether the model is the latest version, associate bugs to the correct model versions, and aid in model comparisons.
* **Model type**: This includes basic model architecture details, such as Naive Bayes classifier, Convolutional Neural Network, etc. This is particularly relevant for AI practitioners to highlight what kinds of assumptions are encoded in the model.
* **Sources**: Papers or other resources with more information.
* **Citation details**: How should the model be cited?
* **License**: Licensing information as appropriate.
* **Feedback channels**: for example, an email address that people can write to for further information.
  1. Use case and users
* **Intended uses**: What general or specific tasks was the model developed for?
* **Intended users**: Was the model developed for entertainment purposes, for hobbyists, or enterprise solutions? This helps users gain insight into how robust the model may be to different kinds of inputs.
* **Out-of-scope uses**: Similar in spirit to warning labels or disclaimers found on products. What are the known limitations of the model?
  1. Model performance metrics
* **Metrics**: What measures of model performance are being reported, and why were they selected? For classification systems, the error types that can be derived from a confusion matrix are useful, as their relative importance depends on context.
* **Thresholds**: If decision thresholds are used, what are they, and how were they chosen?
* **Uncertainty**: What metrics are used to estimate model uncertainty and variability? How were they estimated?
  1. Model performance factors

Which factors are being reported, and why were these chosen? What are foreseeable (other) salient factors for which model performance may vary?

* **Groups**: Groups refers to distinct categories with similar characteristics that are present in the evaluation data instances. For human-centric models, these are people who share one or multiple characteristics. Determining which groups to include depends on the intended use of the model and the context under which it may be deployed.
* **Instrumentation**: The performance of a model can vary depending on what instruments were used to capture the input data to the model.
* **Environment**: Helping the user understand how model performance depends on different environmental conditions.
  1. Data

All referenced datasets would ideally point to a set of documents that provide visibility into the source and composition of the dataset. Evaluation datasets in particular should include datasets that are publicly available for third-party use.

* What datasets were used to train and evaluate the model?
* **Motivation**: Why were these datasets chosen?
* **Pre-processing**: How was the data pre-processed for training or evaluation? What augmentation was performed?

The datasets should not only be representative of the model’s typical use cases but also anticipated test scenarios and challenging cases.

* 1. Analysis

The quantitative analyses should provide the results of evaluating the model according to the chosen metrics, providing measures or uncertainty or confidence where possible. The analysis should be broken down by the chosen factors.

* **Unitary analysis**: How did the model perform with respect to each factor?
* **Intersectional analysis**: How did the model perform with respect to the intersection of evaluated factors?
  1. Ethical considerations

Provide information on ethical considerations that went into model development, surfacing ethical challenges and solutions to stakeholders.

* **Data**: Is the model based on any sensitive data?
* **Risk management**: What risk mitigation strategies were used during model development?
* **Risks and harms**: What risks may be present in model usage? Try to identify the potential recipients, likelihood, and magnitude of harms.
* **Use cases**: Are there any known model use cases that are especially sensitive?
* **Review process**: Was the work reviewed by an external board?

1. Other References

* Qayyum, Adnan & Qadir, Junaid & Bilal, Muhammad & Al-Fuqaha, Ala. (2020). Secure and Robust Machine Learning for Healthcare: A Survey.
* Zinkevich, M. (2019). Rules of Machine Learning: Best Practices for ML Engineering.
* Mitchell, M. et al. T. (2019). Model cards for model reporting. In Proceedings of the Conference on Fairness, Accountability, and Transparency (pp. 220-229). ACM.
* Luo, Gang. (2015). MLBCD: a machine learning tool for big clinical data. Health Information Science and Systems. 3. 10.1186/s13755-015-0011-0.
* Probst, Philipp & Bischl, Bernd & Boulesteix, Anne-Laure. (2018). Tunability: Importance of Hyperparameters of Machine Learning Algorithms.
* Center for Open Data Enterprise (CODE), Sharing and Utilizing Health Data for AI Applications, Roundtable Report, 2019

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Frazier, P.I. (2018). A Tutorial on Bayesian Optimization. ArXiv, abs/1807.02811. [↑](#footnote-ref-1)
2. Wong, W.K., Boscardin, W.J., Postlethwaite, A.E., & Furst, D.E. (2011). Handling missing data issues in clinical trials for rheumatic diseases. Contemporary clinical trials, 32 1, 1-9 . [↑](#footnote-ref-2)
3. Bellemo, Valentina & Lim, Zhan & Lim, Gilbert & Nguyen, Duc Quang & Xie, Yuchen & Yip, Michelle & Hamzah, Haslina & Ho, Jinyi & Lee, Xin & Hsu, Wynne & Lee, Mong & Musonda, Lillian & Chandran, Manju & Chipalo-Mutati, Grace & Muma, Mulenga & Tan, Gavin & Sivaprasad, Sobha & Menon, Geeta & Wong, T-Y & Ting, Daniel. (2019). Artificial Intelligence using Deep Learning to Screen for Referable and Vision-threatening Diabetic Retinopathy in Africa: a Clinical Validation Study. 1. e35-44. 10.1016/S2589-7500(19)30004-4. [↑](#footnote-ref-3)
4. Kelly, C.J., Karthikesalingam, A., Suleyman, M. et al. Key challenges for delivering clinical impact with artificial intelligence. BMC Med 17, 195 (2019). [↑](#footnote-ref-4)
5. Liang, Shiyu & Li, Yixuan & Srikant, R.. (2017). Principled Detection of Out-of-Distribution Examples in Neural Networks. [↑](#footnote-ref-5)
6. Hendrycks, D., & Gimpel, K. (2017). A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. ArXiv, abs/1610.02136. [↑](#footnote-ref-6)
7. De Fauw, J., Ledsam, J.R., Romera-Paredes, B. et al. Clinically applicable deep learning for diagnosis and referral in retinal disease. Nat Med 24, 1342–1350 (2018). [↑](#footnote-ref-7)
8. Miotto R, Wang F, Wang S, Jiang X, Dudley JT. Deep learning for healthcare: review, opportunities and challenges. Brief Bioinform. 2018;19(6):1236–1246. doi:10.1093/bib/bbx044 [↑](#footnote-ref-8)
9. Shickel, Benjamin & Tighe, Patrick & Bihorac, Azra & Rashidi, Parisa. (2017). Deep EHR: A Survey of Recent Advances on Deep Learning Techniques for Electronic Health Record (EHR) Analysis. Journal of Biomedical and Health Informatics.. PP. 10.1109/JBHI.2017.2767063. [↑](#footnote-ref-9)
10. Mitchell, Margaret et al. “Model Cards for Model Reporting.” Proceedings of the Conference on Fairness, Accountability, and Transparency - FAT\* ’19 (2019): n. pag. Crossref. Web. [↑](#footnote-ref-10)