Lifecycle for FAIR Machine Learning

Leyla Jael Castro[1,2][0000-0003-3986-0510]*, Fotis Psomopoulos[3][0000-0002-0222-4273], Beatriz Serrano-Solano[4,5][0000-0002-5862-6132], Curtis Sharma[6][0000-0002-3375-0604], Kirubel Biruk Shiferaw[7][0000-0002-7411-1411], Dhwani Solanki[1,2][0009-0004-1529-0095], Yue Zhang[8][0009-0007-6432-1259]

[1] *ZB MED Information Centre for Life Sciences, Cologne, Germany*
[2] *NFDI4DataScience, Berlin, Germany*
[3] *Institute of Applied Biosciences, Centre for Research and Technology Hellas, Thessaloniki, Greece*
[4] *Euro-BioImaging ERIC Bio-Hub, European Molecular Biology Laboratory (EMBL) Heidelberg, Heidelberg, Germany*
[5] *AI4Life, Turku, Finland*
[6] *4TU.ResearchData/TU Delft, Delft, Netherlands*
[7] *Department of Medical Informatics, Institute for Community Medicine, University Medicine Greifswald, Greifswald, Germany*
[8] *Technical University of Berlin, Berlin, Germany*

*Corresponding author

Abstract

Despite the advances in Machine Learning Operations and the availability of variation of the Machine Learning lifecycle, there is none yet aligned to the Findable, Accessible, Interoperable and Reusable (FAIR) principles. Here we present our proposal of such a lifecycle, including an initial analysis on which and how the FAIR principles apply together with some additional information on reporting best practices and existing resources that could support the different phases in the lifecycle.

Keywords

Machine Learning lifecycle, ML lifecycle, FAIR, FAIR4ML

1. Introduction

Thanks to advancements in what is nowadays known as Machine Learning Operations (MLOps) [1], the lifecycle of Machine Learning (ML) pipelines, from data collection to model monitoring, are well covered, see Figure 1. However, such variations of the ML lifecycle do not necessarily cover the aspects of Findability, Accessibility, Interoperability and Reusability (FAIR) of ML models. In fact, there is not yet clarity on what the FAIR principles would mean in the case of ML [2] as it could include elements from FAIR for data [3], FAIR for software [4,5], FAIR for workflows [6,7] and might also require some new elements explicitly defined for FAIR for ML (or FAIR for Artificial Intelligence), for instance [8–11].

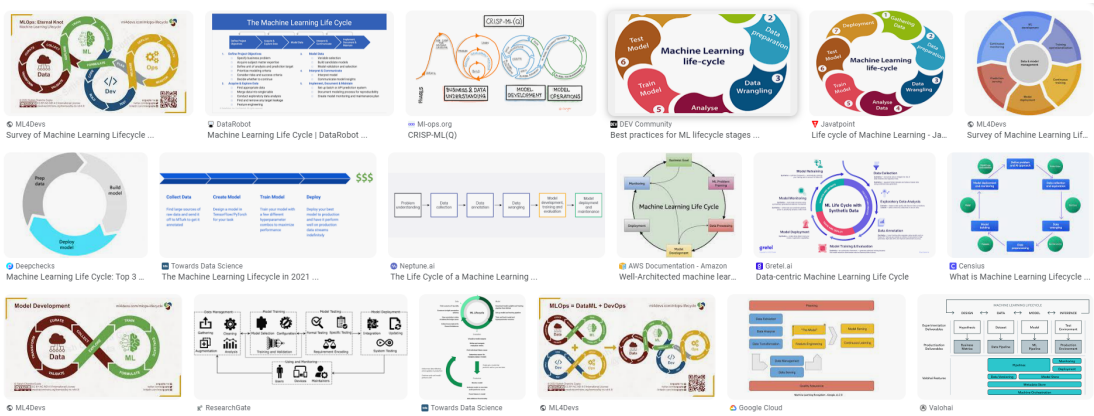

Figure 1. Results from an image search on Google using the query “ml lifecycle”.

To tackle and tune FAIR for ML, the Research Data Alliance (RDA) FAIR4ML Interest Group[1] was officially launched in 2023. Currently, the group has two task forces; Task Force 1 is working on a white paper on FAIR for ML, and Task Force 2 on structured metadata to describe ML approaches. During the RDA Plenary 21[2] in October 2023, Task Force 1 asked participants to list the necessary steps in an ML lifecycle. With the purpose of accelerating and benefiting from the RDA FAIR4ML definition of an ML lifecycle within the scope of FAIR, the Semantic Technologies team (SemTec)[3] at ZB MED Information Centre for Life Sciences (ZB MED)[4] organized a 2-day hackathon on behalf of the National Research Data Infrastructure (NFDI) for Data Science and Artificial Intelligence (NFDI4DS)[5], one of the NFDI[6] consortia in Germany, in which FAIRness and metadata for ML are core activities. This ML Lifecycle hackathon took place with seven participants coming from six European-based institutions. Here we report the results from the hackathon.

On the first day, the same challenge as in the RDA plenary was presented to the hackathon participants: "*list the different phases in the ML lifecycle*". Based on the provided answers and a follow-up discussion, the group identified a set of steps and the corresponding graphical representation. On the second day, the group focused on aligning these steps to the FAIR principles best practices and some possible corresponding resources. In the rest of this document, we elaborate on the results of this 2-day hackathon.

2. Lifecycle for FAIR ML

Participants worked in pairs or individually in the activity "list the different phases in the ML lifecycle", and this yielded four sets of responses as shown in Table 1.

Table 1. List of phases in the ML lifecycle

Please list the different phases in the ML lifecycle	
1. Problem definition (requirements, assumption, etc.) 2. Data acquisition (cleaning, restructuring, AI-ready?) 3. Method selection (identify algorithm, relevant parameters, relevant software) 4. Model creation (use of computer infrastructure, software execution, model optimization) 5. Model evaluation (use of indicators, evaluation, validation)	6. Model retrain? 7. Model deposition (relevant metadata, repository, registry) 8. Model re-use (other dataset, environment, deployment?) 9. Feedback on the above
1. Data collection/data selection 2. Data preprocessing 3. Define the ML problem or approach that we would to solve, select the ML task, and select the algorithm 4. Model Training	5. Parameters 6. Evaluation 7. Optimization 8. Model Deployment

[1] Research Data Alliance (RDA) FAIR4ML Interest Group https://www.rd-alliance.org/groups/fair-machine-learning-fair4ml-ig
[2] RDA Plenary 21 https://www.rd-alliance.org/plenaries/international-data-week-2023-salzburg
[3] ZB MED SemTec https://zbmed-semtec.github.io/
[4] ZB MED https://www.zbmed.de/en/
[5] NFDI4DataScience (NFDI4DS) https://www.nfdi4datascience.de/
[6] NFDI https://www.nfdi.de/?lang=en

Please list the different phases in the ML lifecycle	
1. Definition of the problem 2. Design 3. Data collection 4. Data preprocessing 5. Model training 6. Model evaluation	7. Model validation 8. Documentation 9. Model sharing 10. Publication 11. Retraining 12. Reuse
1. Problem-Method argument and definition 2. Data gathering 3. Preprocessing 4. Training and testing/ retraining /experiment tracking	5. Validation 6. Reporting/ 7. Communicating results 8. Model+SC archiving

Based on the proposed steps for the ML lifecycle collected in Table 1, the activity carried out during the RDA Plenary 21, and the discussions during the hackathon itself, we came up with names for the individual steps/phases, including descriptions, outcomes and notes for each step (see Section 2.1 Step by Step). Questions arose during the discussion that produced further debate and suggested additional clarifications were needed. We have collected these in Section 2.1 Definitions and Q&A. Figure 2 shows a graphical depiction of the phases/steps corresponding to the ML lifecycle.

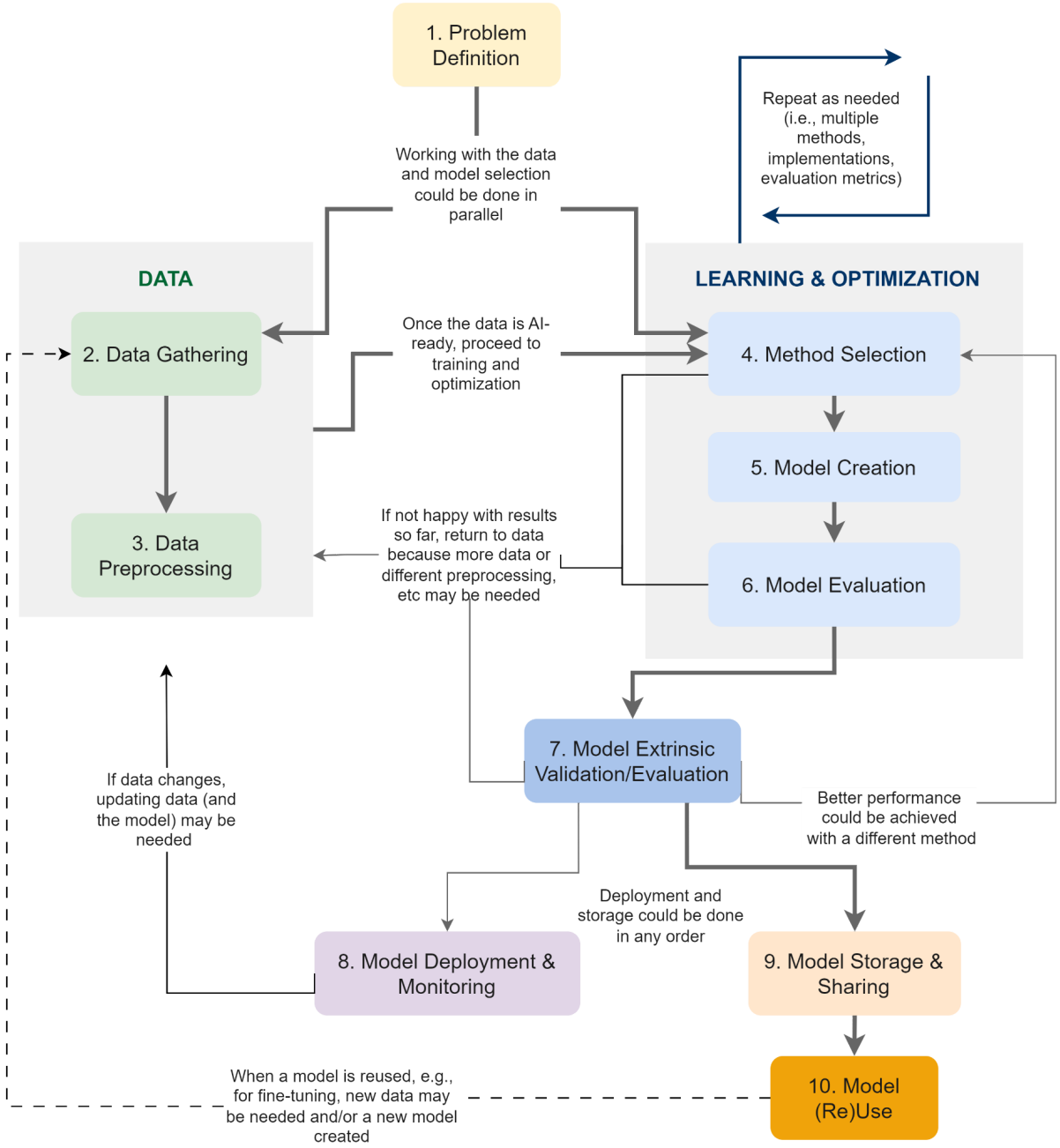

Figure 2. ML lifecycle from problem definition to model (re)use. The bolded lines show the most common steps in ML done during research (e.g., model deployment is not yet a common step in research, though it may be in industry). This bolded path also shows a somewhat ideal scenario where everything goes well from the first try (which rarely happens). Some backward and loop lines show deviations due to the need for further data processing, or alternative learning and optimization approaches to obtain a better evaluation.

2.1. Step by step

Name: Problem Definition

Description: Define the problem that we would like to solve using ML and identify the approach that we propose (i.e. the corresponding ML task). This will require documenting the problem assumptions and listing the expected requirements (both algorithmic and computational), as well as confirming that ML is indeed necessary.
Outcome: By the end of this, you will have an understanding of the problem assumption and the expected ML requirements.
Notes: (i) The appropriateness of the selected ML method and the problem at hand (nature of data) should be carefully evaluated, (ii) The computational infrastructure capacity at hand should be considered during problem definition.

Name: Data Gathering
Description: Collect the data that will be used in the ML process. This will require defining the access protocol, the data type and the file format used, as well as any Ethical, Legal and Societal Aspects and Implications (ELSA/ELSI).
Outcome: By the end of this, the necessary data for tackling the problem will be available and accessible to you.
Notes: Access protocol encompasses obtaining the actual data directly, from multiple sources, or from a single source. Data type could be image, text, sequence, etc. Data format could be proprietary, open, etc.

Name: Data Preprocessing
Description: Using the available data, apply the necessary preprocessing steps to ensure that the data is AI-ready (such as data imputation, class rebalancing, characterizing dataset, restructuring data, feature selection, etc.). An important aspect of this step is to be aware of any effects that the preprocessing has on the structure/distribution of the original data (i.e. introducing biases).
Outcome: By the end of this, the data will be ready to be used in the appropriate ML process.
Notes: A good practice in this step is for the final AI-ready data to be deposited to a repository, including all characterization so that it can be reused (transparency).

Name: Method Selection
Description: Identify the appropriate ML method and/or algorithm that fits the problem definition and the underlying assumptions (i.e. requirements). These requirements will also allow the identification of the key method parameters that will need to be taken into consideration. Based on this, also identify existing software that can be used under the appropriate license (or implement it if none exists) that will run the model creation process.
Outcome: By the end of this, you will have the ML method and software ready to use on the AI-ready data for the model creation.
Notes: For example, the selection of a clustering method would be a k-mean implementation in R, that will only use the k parameter to investigate the segmentation of the AI-ready dataset.

Name: Model Creation
Description: Using the method and/or algorithm selected, with the AI-ready data, run the appropriate software (either re-used or implemented) on the appropriate compute infrastructure, and train an ML model.

Outcome: By the end of this, you will have a trained, optimized, working ML model to address the problem defined.
Notes: This process could also include iterations of training and test phases (training/test, training/test/validation, etc.), towards the optimization of the model parameter values (parameter tuning). A good practice here is detailed documentation of every individual step of the internal process (documentation).

Name: Model Evaluation
Description: Using a set of metrics, indicators and other descriptors (ideally community-backed and problem-specific) to understand whether the ML process has run appropriately. These metrics will be assessed against given thresholds (or other ground-truth metrics) to assess model performance.
Outcome: By the end of this, you will have a list of model performance indicators.
Notes: —

Name: Extrinsic evaluation/validation of the model
Description: The model output is validated against a completely foreign/external dataset or experimentally, in order to assess whether it addresses the problem it was created for in the first place. Another example of validation could be a benchmarking process.
Outcome: By the end of this, you will have some measure of validation of the created model.
Notes: —

Name: Model Deployment and Monitoring
Description: Your created model is being used **as is** in practice as part of an established process (such as supporting a Supply Chain Management -SCM, workflow). While part of the overall workflow, the output of the model is constantly assessed against the expected performance (such as when the input data is starting to significantly deviate from the data used to train the original model). In that case, you need to re-train the model.
Outcome: By the end of this, you will have a framework to continuously monitor a deployed ML model, so that you can assess whether a full retraining process is required.
Notes: —

Name: Model Storage and Sharing
Description: The created model should be deposited and stored in a repository - the repository selection is driven by the characteristics of the model itself (such as size, license, etc.) and also community practices. In order to be able to share the model, the respective metadata of the model needs to be made available through an appropriate registry. The metadata of the model includes all aspects of the model creation including data, software, evaluation, etc.
Outcome: By the end of this, your model will be made available to others, with clear metadata describing it.
Notes: The selection of repository/registry services is completely up to the user and also according to the license / requirements of the model itself. There are services (such as Hugginface) that are both a repository and a registry, but there are also ML-specific registries (such as the DOME registry) and general repositories (such as Zenodo and GitHub).

Name: Model (re)use
Description: You retrieve a model that has been made available by others, and you use it (as it is) using your own data that fits the model input requirements. Another option is to fine-tune the model by retraining it to fit your own needs (such as fine-tuning LLMs for a domain-specific corpus or using a set of Earth observation images with a model originally trained for microscopy data).
Outcome: By the end of this, you will have a variation of a created model to fit your own needs.
Notes: —

2.2. Definitions / Q&A

Q: What does AI-ready mean?
A: Everything is done, so that it's ready to run ML on it. Example from AI4Life: https://ai4life.eurobioimaging.eu/the-bia-launches-a-collection-of-explorable-ai-ready-image-datasets/ From the raw-data you can have multiple AI-ready datasets generated from this, depending on the objective. Also, an AI-ready dataset for a particular objective might require additional preprocessing towards an AI-ready dataset for another objective

Q: What is the difference between evaluation and validation?
A: Evaluation is assessing whether the process runs as intended. Validation is if the model addresses the problem it was designed for.

Q: What was not considered as part of the ML lifecycle presented here?
A: There are some things that we ignored (as part of the ML Lifecycle):

- Publication of the model (either as a model on its own or via a traditional scholarly publication describing methods, materials and results)
- Documentation of the software and model
- Continuous monitoring of the process
- Lifecycle of data and software

Q: What is really an ML model?
A: Here we present some points to keep in mind that should help clarify what we mean by ML model

- An ML model is the result of a machine learning process. Examples include
  - A clustering analysis produces a set of clusters which model that data
  - A topic modeling algorithm produces a set of topics from a corpus of documents that may be useful, for example, when comparing new texts.
  - A classifier is a model that results of training an algorithm with certain supervised data
- An executable ML model is a result of an ML training process that produces a file (pkl, bin, etc.) that can be used for a given task accepting an input and producing an output (i.e., prediction). Examples of tasks are regression, classification, etc.
- Examples of ML models that are not executable ML models: clustering, PCA analysis.
- Examples of ML executable models include LLMs, classifiers, fine-tuned models, etc.

3. FAIR aspects of the ML lifecycle

The discussion on day 2 is summarized in Table 2 which shows aspects related to FAIR and best practices for reporting the ML lifecycle steps as well as related information and research artifacts. Data and software FAIRness were not considered independently, i.e., data and software are analyzed only as parts of the ML lifecycle. This table is not intended to be exhaustive and collects mostly approaches known to the participants, and it focuses mostly on the software and model aspects of ML.

Table 2. FAIR and good practices to consider for the ML lifecycle. The first column indicates which set of FAIR principles may apply, the second column aligns to either the Data, Optimization, Model and Evaluation (DOME) [12] recommendations for supervised machine learning in computational biology or the ML model cards [13], the third column collects related metadata schemas, the fourth column show some services that could offer some support to improve FAIRness and the fifth column has some activities that could help to improve FAIRness.

		FAIR Principles	Best practices on reporting	Metadata schemas	Resources	What do you need to do here
1	Problem Definition	FAIR Data				- Documentation
		FAIR Software				- Documentation
		FAIR AI Models				- Documentation
2	Data Gathering	FAIR Data (for training dataset)	DOME (D part)		- Data Management, e.g., Data Stewardship Wizard (DSW)[7] [14]and Research Data Management Organizer (RDMO)[8] [15] - Report data provenance and availability DOME registry[9] and BioImage Archive[10] - SPDX licenses[11]	- Create a DMP - Fill in information on the data in the DOME registry through the DOME Wizard
		FAIR Software				
		FAIR AI Models				- Fill in information on the data in the DOME registry through the DOME Wizard
3	Data Preprocessing	FAIR Data (for training dataset and splits)	DOME (D part)	ML Commons Croissant[12]	- Data Management, e.g., DSW and RDMO - Report data splits DOME registry - SPDX licenses	- Create a DMP of the AI-ready data - Report data features - Report data splits and data distribution - Report feature selection / augmentation

[7] DSW https://ds-wizard.org/
[8] RDMO https://github.com/rdmorganiser/rdmo
[9] DOME registry https://registry.dome-ml.org/intro
[10] BioImage Archive https://www.ebi.ac.uk/bioimage-archive/galleries/AI.html
[11] SPDX https://spdx.org/licenses/
[12] ML Commons Croissant https://github.com/mlcommons/croissant

		FAIR Software (for data pre-processing)		Codemeta[13] [16], Bioschemas[14] [17,18][15], maSMP[16] [19–21]	- ELIXIR SMPs [22] - SMPs in RDMO and metadata extraction [23] - SPDX licenses	- Create the SMP - Create the software - Report feature selection / augmentation
		FAIR AI Models			- DOME registry - BioImage Archive (AI-ready datasets)	
4	Method Selection	FAIR Data				
		FAIR Software		Codemeta, Bioschemas Computational tool, maSMP	- ELIXIR SMPs - SMPs in RDMO and metadata extraction - SPDX licenses	- Update/create the SMP - Update software metadata - Selection of hyperparameters (actually used vs available)
		FAIR AI Models	DOME (O part) ML cards		DOME registry	
5	Model Creation	FAIR Data				
		FAIR Software		Codemeta, Bioschemas Computational tool, maSMP	- ELIXIR SMPs - SMPs in RDMO and metadata extraction - SPDX licenses	- Update/create the SMP - Update software metadata
		FAIR AI Models	DOME (M part) ML cards		- DOME registry - NFDI4DS PADME analytics and federated ML[17]	- Report model characteristics, limitations, bias - Report optimization/tuning strategy and hyperparameters
6	Model Evaluation	FAIR Data				
		FAIR Software				
		FAIR AI Models (possibly only the “winning”	DOME (E part) ML cards		- DOME registry - BioImage Model Zoo[18] - NFDI4DS Gerbil benchmark[19]	- Archive/Publish model - Report evaluation metrics and values - Report final hyperparameters

[13] Codemeta https://codemeta.github.io/
[14]Bioschemas https://bioschemas.org
[15] Bioschemas ComputationalTool https://bioschemas.org/profiles/ComputationalTool
[16] maSMP https://zbmed-semtec.github.io/maSMPs/ and https://discovery.biothings.io/ns/maSMP
[17] PADME analytics and federated ML https://websites.fraunhofer.de/PersonalHealthTrain/
[18] BioImage Model Zoo https://bioimage.io/
[19] Gerbil https://aksw.org/Projects/GERBIL.html

		model)			- NFDI4DS European Language Grid (ELG)[20] [24]	and model weights - Fill in an evaluation report
7	Model (extrinsic) Validation / Evaluation	FAIR Data				- Document data validation set including provenance trace
		FAIR Software				
		FAIR AI Models	DOME (E part)		- DOME registry - NFDI4DS Gerbil benchmark - NFDI4DS ELG	- Report extrinsic validation / evaluation results
8	Model Deployment & Monitoring	FAIR Data				
		FAIR Software				
		FAIR AI Models				- For "executable" models, report how this can be used and run from elsewhere
9	Model Storage & Sharing	FAIR Data			- SPDX licenses - Hugging Face Licenses[21]	Licensing
		FAIR Software			- SPDX licenses - Hugging Face Licenses	Licensing
		FAIR AI Models	For instance DECIDE AI [25] - CONSORT AI [26] - SPIRIT AI [27]		- DOME registry - NFDI4DS ELG - HuggingFace[22], MLFlow[23], OpenML[24] [28], BioImage Model Zoo - GitHub[25], GitLab[26] - Zenodo[27]	- Choosing repositories, metadata management - Create/update metadata for model - Report This includes the environment configuration and dependency information (YAML,TOML, JSON, etc.)
10	Model (Re)Use	FAIR Data				
		FAIR Software				
		FAIR AI Models			- NFDI4DS ELG - HuggingFace[28], MLFlow[29],	- This includes fine tuning / transfer learning

[20] ELG https://live.european-language-grid.eu/
[21] HuggingFace licenses https://huggingface.co/docs/hub/repositories-licenses#licenses
[22] HuggingFace https://huggingface.co/
[23] MLFlow https://mlflow.org/
[24] OpenML https://www.openml.org/
[25] GitHub https://github.com/
[26] GitLab https://about.gitlab.com/
[27] Zenodo https://zenodo.org/
[28] HuggingFace https://huggingface.co/
[29] MLFlow https://mlflow.org/

					OpenML[30], BioImage Model Zoo	

4. Conclusions and future work

The RDA FAIR4ML group will continue working on the FAIRness for ML while NFDI4DS will adopt (and further develop) their recommendations for the ML FAIRness evaluator currently under development. Moreover, the ELIXIR Machine Learning Focus Group will further develop the DOME registry, to encompass both the common metadata schema as well as support indicators for FAIRness. ZB MED/NFDI4DS will deliver a visualization tool for the ML lifecycle that takes input from the FAIRness, good practices and resources table so that readers can easily interact with the different elements and learn more about related information.

Acknowledgements

All activities reported here were carried out during the NFDI4DS Machine Learning Lifecycle hackathon from the 21st to the 22nd of November 2023 at ZB MED. This hackathon was organized by the Semantic Technologies team and sponsored by NFDI4DataScience. NFDI4DataScience is a consortium funded by the German Research Foundation (DFG), project number 460234259.

References

1. Kreuzberger D, Kühl N, Hirschl S. Machine Learning Operations (MLOps): Overview, Definition, and Architecture. IEEE Access. 2023;11: 31866–31879. doi:10.1109/ACCESS.2023.3262138
2. Castro LJ, Katz DS, Psomopoulos F. Working Towards Understanding the Role of FAIR for Machine Learning. PUBLISSO; 2021. doi:10.4126/FRL01-006429415
3. Wilkinson MD, Dumontier M, Aalbersberg IjJ, Appleton G, Axton M, Baak A, et al. The FAIR Guiding Principles for scientific data management and stewardship. Sci Data. 2016;3: 160018. doi:10.1038/sdata.2016.18
4. Chue Hong NP, Katz DS, Barker M, Lamprecht A-L, Martinez C, Psomopoulos FE, et al. FAIR Principles for Research Software (FAIR4RS Principles). 2022. doi:10.15497/RDA00068
5. Barker M, Chue Hong NP, Katz DS, Lamprecht A-L, Martinez-Ortiz C, Psomopoulos F, et al. Introducing the FAIR Principles for research software. Sci Data. 2022;9: 622. doi:10.1038/s41597-022-01710-x
6. Visser C de, Johansson LF, Kulkarni P, Mei H, Neerincx P, Velde KJ van der, et al. Ten quick tips for building FAIR workflows. PLOS Computational Biology. 2023;19: e1011369. doi:10.1371/journal.pcbi.1011369
7. Goble C, Cohen-Boulakia S, Soiland-Reyes S, Garijo D, Gil Y, Crusoe MR, et al. FAIR Computational Workflows. Data Intelligence. 2020;2: 108–121. doi:10.1162/dint_a_00033
8. Duarte J, Li H, Roy A, Zhu R, Huerta EA, Diaz D, et al. FAIR AI Models in High Energy Physics. arXiv; 2022. doi:10.48550/arXiv.2212.05081
9. Ravi N, Chaturvedi P, Huerta EA, Liu Z, Chard R, Scourtas A, et al. FAIR principles for AI models with a practical application for accelerated high energy diffraction microscopy. Sci Data. 2022;9: 657. doi:10.1038/s41597-022-01712-9
10. Katz DS, Pollard T, Psomopoulos F, Huerta E, Erdmann C, Blaiszik B. FAIR principles for Machine Learning models. 2020 [cited 25 Aug 2023]. doi:10.5281/ZENODO.4271996
11. Huerta EA, Blaiszik B, Brinson LC, Bouchard KE, Diaz D, Doglioni C, et al. FAIR for AI: An interdisciplinary and international community building perspective. Sci Data. 2023;10: 487. doi:10.1038/s41597-023-02298-6
12. Walsh I, Fishman D, Garcia-Gasulla D, Titma T, Pollastri G, Capriotti E, et al. DOME:

[30] OpenML https://www.openml.org/

recommendations for supervised machine learning validation in biology. Nature Methods. 2021. doi:10.1038/s41592-021-01205-4
13. Mitchell M, Wu S, Zaldivar A, Barnes P, Vasserman L, Hutchinson B, et al. Model Cards for Model Reporting. Proceedings of the Conference on Fairness, Accountability, and Transparency. 2019. pp. 220–229. doi:10.1145/3287560.3287596
14. Pergl R, Hooft R, Suchánek M, Knaisl V, Slifka J. "Data Stewardship Wizard": A Tool Bringing Together Researchers, Data Stewards, and Data Experts around Data Management Planning. Data Science Journal. 2019;18: 59. doi:10.5334/dsj-2019-059
15. Klar J, Michaelis O, Engelhardt C, Enke H, Frenzel J, Hausen D, et al. Research Data Management Organizer (RDMO). 2023. doi:10.5281/zenodo.596581
16. Jones MB, Boettiger C, Mayes AC, Arfon Smith, Slaughter P, Niemeyer K, et al. CodeMeta: an exchange schema for software metadata. KNB Data Repository. KNB Data Repository; 2016. doi:10.5063/SCHEMA/CODEMETA-1.0
17. Gray AJG, Goble C, Jimenez RC. From Potato Salad to Protein Annotation. ISWC Posters and Demo session. Vienna, Austria; 2017. p. 4. Available: http://ceur-ws.org/Vol-1963/paper579.pdf
18. Gray A, Castro LJ, Juty N, Goble C. Schema.org for Scientific Data. Artificial Intelligence for Science. WORLD SCIENTIFIC; 2022. pp. 495–514. doi:10.1142/9789811265679_0027
19. Giraldo O, Geist L, Quiñones N, Solanki D, Rebholz-Schuhmann D, Castro LJ. machine-actionable Software Management Plan Ontology (maSMP Ontology). Zenodo; 2023. doi:10.5281/zenodo.7806638
20. Giraldo O, Dessi D, Dietze S, Rebholz-Schuhmann D, Castro LJ. Machine-Actionable Metadata for Software and Software Management Plans for NFDI. Proceedings of the Conference on Research Data Infrastructure. 2023. doi:10.52825/cordi.v1i.279
21. Giraldo O, Geist L, Quiñones N, Solanki D, Alves R, Bampalikis D, et al. A metadata schema for machine-actionable Software Management Plans. PUBLISSO-FRL; 2023. doi:10.4126/FRL01-006444988
22. Alves R, Bampalikis D, Castro LJ, González JMF, Harrow J, Kuzak M, et al. ELIXIR Software Management Plan for Life Sciences. BioHackrXiv; 2021. doi:10.37044/osf.io/k8znb
23. Castro LJ, Geist L, Gonzalez E, Gonzalez-Ocanto M, Grossmann YV, Pronk T, et al. Five Minutes to Write a Software Management Plan – A Machine-actionable Approach to Simplify the Creation of SMPs. Zenodo; 2023. doi:10.5281/zenodo.10374839
24. Rehm G, Berger M, Elsholz E, Hegele S, Kintzel F, Marheinecke K, et al. European Language Grid: An Overview. In: Calzolari N, Béchet F, Blache P, Choukri K, Cieri C, Declerck T, et al., editors. Proceedings of the Twelfth Language Resources and Evaluation Conference. Marseille, France: European Language Resources Association; 2020. pp. 3366–3380. Available: https://aclanthology.org/2020.lrec-1.413
25. Vasey B, Clifton DA, Collins GS, Denniston AK, Faes L, Geerts BF, et al. DECIDE-AI: new reporting guidelines to bridge the development-to-implementation gap in clinical artificial intelligence. Nat Med. 2021;27: 186–187. doi:10.1038/s41591-021-01229-5
26. Liu X, Cruz Rivera S, Moher D, Calvert MJ, Denniston AK. Reporting guidelines for clinical trial reports for interventions involving artificial intelligence: the CONSORT-AI extension. Nat Med. 2020;26: 1364–1374. doi:10.1038/s41591-020-1034-x
27. Cruz Rivera S, Liu X, Chan A-W, Denniston AK, Calvert MJ. Guidelines for clinical trial protocols for interventions involving artificial intelligence: the SPIRIT-AI extension. Nat Med. 2020;26: 1351–1363. doi:10.1038/s41591-020-1037-7
28. Vanschoren J, van Rijn JN, Bischl B, Torgo L. OpenML: networked science in machine learning. SIGKDD Explor Newsl. 2014;15: 49–60. doi:10.1145/2641190.2641198