MEDICAL EVENT DATA STANDARD (MEDS): FACILITATING MACHINE LEARNING FOR HEALTH

MEDS Working Group[*]
Bert Arnrich, Edward Choi, Jason A. Fries, Matthew B. A. McDermott, Jungwoo Oh, Tom J. Pollard, Nigam Shah, Ethan Steinberg, Michael Wornow, Robin van de Water
Please direct correspondence to: https://github.com/Medical-Event-Data-Standard/meds

ABSTRACT

We introduce the *Medical Event Data Standard* (MEDS), a lightweight schema for enabling machine learning over electronic health record (EHR) data. Unlike common data models and data interoperability formats, MEDS is a minimal standard designed for maximum interoperability across datasets, existing tools, and model architectures. By providing a simple standardization layer between datasets and model-specific code, MEDS will enable more reproducible, robust, computationally performant, and collaborative machine learning research using EHR data. We highlight several existing MEDS integrations with models, datasets, and tools, and invite the community for further development and adoption. Please see our Github page for further details: https://github.com/Medical-Event-Data-Standard/meds.

1 INTRODUCTION

In Natural Language Processing (NLP) the Hugging Face platform (Wolf et al., 2019) has been able to provide access to many large language models by using a shared data and model interface, e.g, Jiang et al. (2023). Pre-trained neural network encoders from Computer Vision (CV) are likewise easy to share and reuse because images are always input into models in one of a small set of standard representations. The standardization of data input mechanisms has been instrumental in communal research, shareable model code, and model portability.

In healthcare, the small number of datasets that are released have no common data input format for building ML models. As a result, when models are released, they make different assumptions about dataset pre-processing and expected output formats, leading to significant reproducibility challenges (McDermott et al., 2021; Wornow et al., 2023). As the need to provide assurance of meeting minimal performance of models in healthcare rises (Shah et al., 2024), with calls for higher transparency, we need to enable the sort of model sharing and code re-usability as done in NLP and CV communities. To address this need, we introduce the Medical Event Data Standard (MEDS, accessible at https://github.com/Medical-Event-Data-Standard/meds), a data input standard for structured electronic health record (EHR) data for machine learning that has three key benefits:

1. *Interface Simplicity*: MEDS is an event-based format that reflects the natural structure of data from electronic health records—in the form of time-stamped events occurring one after the other.
2. *Code Reusability*: MEDS MEDS enables sharing and reuse of code for data pre-processing, model learning and evaluation; thus promoting collaborative research.
3. *Model Portability*: MEDS enables creation of models that port across datasets, facilitating faster development and easier deployment of pre-trained models.

MEDS complements common data models (CDMs) in medicine, such as the OMOP (Makadia & Ryan, 2014) or i2b2 (Murphy et al., 2010) CDMs, by providing a mechanism to read from those CDMs and create standardized inputs for deep learning setups. MEDS is particularly relevant in the "foundation model" era of ML due to the increasing pressure to deploy, audit, evaluate, and regulate such models in health settings.

[*] Authors have been sorted alphabetically.

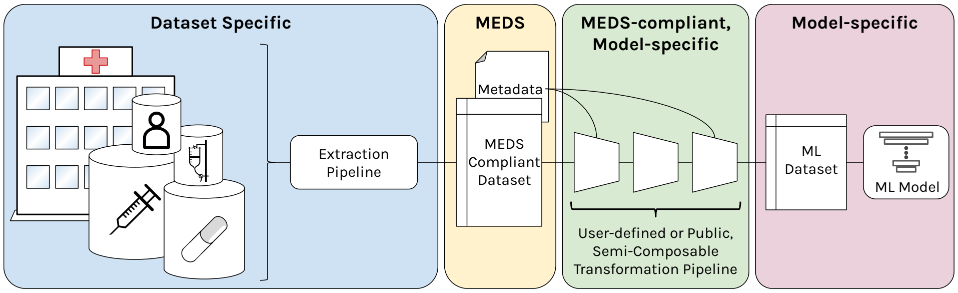

Figure 1: MEDS defines a dataset and metadata schema such that, after any raw EHR dataset is extracted into a compliant form, users can build streamlined, shareable, reproducible custom modeling pipelines atop that dataset.

2 Related Work

Healthcare data standards are a longstanding area of interest and broadly fall into two categories: (1) common data models (CDMs) supporting researchers performing cross-site observational health studies; and (2) interoperability standards supporting the exchange of data across different healthcare information technology systems. CDMs include initiatives such as i2b2, OMOP-CDM, and PCorNet (Fleurence et al., 2014) and have enabled large-scale population-level studies across different clinical sites. CDMs specify a canonical relational database structure for harmonizing patient-level data and have wide support across hundreds of health systems (Klann et al., 2018). Each CDM further supports its own ecosystem of analytics platforms and ML APIs for R, SQL/SAS, and Python. In contrast, interoperability standards such as FHIR and HL7 are designed for completeness of data exchanges across healthcare systems.

All of these data standards live upstream of ML workflows and require often complex transformations to facilitate model training and inference. Several software frameworks have been proposed to address this need, providing libraries of standardized transformation functions and supporting popular research datasets including MIMIC, eICU, HiRID, and AUMCdb. FIDDLE is a preprocessing library that generates feature vectors given tabular EHR data (Tang et al., 2020). PyHealth is an end-to-end deep learning framework for EHRs and provides several ML models out-of-the-box (Yang et al., 2023). TemporAI (Saveliev & van der Schaar, 2023) and Clairvoyance (Jarrett et al., 2021) focus on medical time-series data. YAIB supports benchmarking ICU tasks (van de Water et al., 2024).

MEDS is designed to complement existing data standards and ML frameworks in several ways. First, we define a simple data specification that enables sharing trained models across different frameworks. Current models, even when trained on the same underlying data model and data source, are siloed under the specific framework used to build the model. Second, our standard is designed to directly support sequence models which are currently state-of-the-art for patient classification tasks. Finally, our format is designed for scalability and speed, a critical need when training foundation models using millions of patients' longitudinal timelines.

3 MEDS Design Principles

MEDS adheres to four key design principles, outlined in Figure 1 and described below.

Health Data is Representable as a Stream of Discrete Events Event stream datasets are composed of irregularly sampled, continuous-time sequences of complex events (McDermott et al., 2024). Formulating health data as an event stream aligns with how patient data is collected in practice and enables defining a parsimonious data specification to represent health state. Our sequence formulation further aligns with recent optimizations in large-scale sequence modeling, providing access to open source, high-performance data pre-processing systems. ***MEDS assumes medical records are comprised of a sequence of atomic (indivisible) clinical events.***

```python
# This is an Apache Arrow schema
import pyarrow as pa

measurement = pa.struct([
    # A string code that describes what the event is, e.g. "LOINC/777-3"
    ("code", pa.string()),
    # Optional values
    ("text_value", pa.string()),
    ("numeric_value", pa.float32()),
    ("datetime_value", pa.timestamp("us")),
    # Metadata can be any arbitrary value
    ("metadata", pa.null()),
])

event = pa.struct([
    # The time at which each measurement in this event occurs
    ("time", pa.timestamp("us")),
    # Each event consists of a series of measurements
    ("measurements", pa.list_(measurement))
])

patient = pa.schema([
    ("patient_id", pa.int64()),
    # Fixed measurements which don't have a specific time of occurrence
    ("static_measurements", pa.list_(measurement)),
    # Events must be ordered by time
    ("events", pa.list_(event)),
])
```

Figure 2: The MEDS `patient` schema which stores patient information. Each row in the schema consists of all data for a single patient, consisting of a `patient_id` and a list of `events`. Each event contains a timestamp and a list of measurements that occurred at that event.

Preserve Source Data Provenance Modern deep learning algorithms learn feature representations directly from data input streams, subsuming most aspects of manual feature engineering, but coming at the cost of increased training set sizes and larger parameterized models. Under a representation learning regime, input data should resemble the distribution expected at inference time as closely as possible. Transformations of data external to a model workflow may not be easily reproducible, and can induce distribution shifts that impact model performance and portability. ***MEDS prioritizes preserving source data in the event stream to enable tracking the provenance of transformations.***

Computational Efficiency Most of the benefits of recent foundation models are due to large-scale self-supervised learning using massive datasets and compute infrastructures. In settings with shared standards, the open source community has made considerable progress in empowering the training of these models at scale in computationally efficient ways. Healthcare has lagged in engineering data infrastructures to take advantage of these advances, leaving practitioners to build their own, customized, often inferior solutions. ***MEDS reflects best practices in data organization, pre-processing, and training recipes from the outset.***

Code Reusability Models and codebases designed for a specific MEDS dataset can be applied to any dataset that adhering to the standard, enhancing reproducibility and enabling validation across different datasets. Moreover, MEDS provides composable preprocessing pipelines, akin to those in `torchvision` and Hugging Face for natural language datasets. ***MEDS enables code and pipeline component re-use to facilitate reproducibility and foster community-driven development of ML models.***

4 THE MEDS SCHEMA

Within the Medical Event Data Standard, datasets are represented as rows of `patient` objects, where each `patient` object contains all recorded information about a patient. This `patient` schema is shown in Figure 2 and described in the `Patient Schema` section of this document. We also define dataset-specific metadata as a `metadata` schema, shown in C and defined in the `Dataset Metadata Schema` section of this document. The canonical repository for these schemas is `https://github.com/Medical-Event-Data-Standard/meds`.

Patient Schema In order to store patient data, we define an Apache Arrow `patient` schema that is shown in Figure 2. We also provide some sample data in Appendix B. Each row in the `patient` schema represents contains a `patient_id` (a unique identifier for each patient), a list of `static_measurements` (observations about the patient that are true for the life of the patient), and a the list of timestamped `events` (observations which occur at a particular time). Each timestamp `event` in turn contains a list of `measurements` associated with that timestamp. Each measurement consists of a categorical `code`[1] field, and a set of optional fields for typed values (`text_value`, `numeric_value`, and `datetime_value`). Codes do not have a required format, but it is recommended that they be in the form of of `"VOCABULARY_NAME/CODE_TEXT"` when the code comes from a standard vocabulary with an OHDSI-defined vocabulary name. For example, it is recommended to represent the ICD10CM code I50.9 as `"ICD10CM/I50.9"`. `measurements` also support dataset-specific measurement properties outside the scope of a single code and value (e.g., the priority of a recorded diagnostic code or the fluid source for a microbiology lab). To include properties in a dataset, the data owner can simply store an arbitrary Arrow type within the properties field of the measurements struct. In this way, the core `patient` schema (defining patient identifiers, events, and measurements) does not change, but new data can still be included in a type-safe manner.

Dataset Metadata Schema Users may want to leverage dataset-specific information when working with an MEDS-compliant dataset. To support that, we specify a common format for that information with a `metadata` JSON schema, detailed in Appendix C. We have added various useful fields to this schema, including versioning information, string descriptions for dataset-specific non-standard codes, and optional mappings from dataset-specific non-standard codes to public ontologies. As MEDS is adopted, we hope that common metadata column and measurement properties conventions will emerge and be adopted by dataset curators. Such information can then be reliably used by modelers. If needed, metadata can be integrated to the `patient` schema itself. We leave the emergence of this to the community through organic usage.

5 THE MEDS ECOSYSTEM

Several existing tools, models, and pipelines have already established MEDS-compliant versions of their tools. We list each here.

Framework for Electronic Medical Records (FEMR) (Wornow et al., 2024) provides an MEDS-compliant suite of data preprocessing tools for building machine learning models on top of EHR and claims data at scale. FEMR has been used for the development and release of multiple clinical foundation models (Steinberg et al. (2024) and Steinberg et al. (2021)), benchmark dataset releases (Wornow et al. (2024) and Huang et al. (2023)), and evaluation of model performance across different hospitals (Guo et al., 2023).

Event Stream GPT (ESGPT) (McDermott et al., 2024) contains a data pre-processing and extraction system that can produce MEDS-compliant datasets from various raw source files and model those datasets via generative, autoregressive neural network foundation models.

General Healthcare Predictive Framework (GenHPF) (Hur et al., 2023) provides a set of transformation functions that converts MEDS-compliant datasets into GenHPF-formatted datasets that can be directly processed to GenHPF framework. In addition, by utilizing ESGPT as an extractor of MEDS, they provide a data pre-processing pipeline that can generate example MEDS-compliant datasets for

[1] Categorical codes are currently recorded as Apache Arrow string types to enable integration with HF datasets, but this will be changed soon.

three different public EHR datasets: MIMIC-III (Johnson et al., 2016), MIMIC-IV (Johnson et al., 2023), and eICU (Pollard et al., 2018).

Yet Another ICU Benchmark (YAIB) (van de Water et al., 2024) is an end-to-end ICU experimental framework designed for creating ML models over multiple cohorts from a range of ICU datasets. MEDS-compliant datasets can be used as an interoperable data input format for benchmarking clinical prediction tasks such as Sepsis, Mortality, Length of Stay, Kidney Function, and Acute Kidney Injury. MEDS support makes it easier to port and use new datasets, even if they are private.

6 Discussion

Limitations MEDS has three primary limitations. First, MEDS does not standardize medical code vocabularies. Medical data can be encoded in many different ways, varying from different international standards like SNOMED to country specific code sets, and even hospital specific codes. MEDS does not require any particular set of codes, so different MEDS datasets can and will use very different standards. While this is usually not an issue for modeling, it is important for phenotyping, as clinical knowledge of a particular code set is usually required to write phenotyping algorithms. However, users can work around this limitation by specifying the code ontologies that their phentopying algorithms support. For example, we would expect an ICD10 diabetes phenotype to work across MEDS datasets that use ICD10 codes.

Second, MEDS is a nested schema, in that it uses schema elements like lists to represent how patients contain events, which in turn contain measurements. While this has advantages due to the intrinsic nested structure of healthcare data, it means MEDS cannot be natively processed by tools such as MySQL, REDIVIS, and CSV that do not support nested data schemas. However, we address this limitation by introducing a companion unnested schema MEDS-FLAT. MEDS-FLAT is an unnested version of MEDS with simple ETLs between MEDS and MEDS-FLAT. A user who wants to process or store a MEDS dataset within a tool that does not support nested data can simply transform their data into MEDS-FLAT before using said tool.

Third, MEDS is primarily designed for structured data. It does not currently support image or waveform data, but we anticipate making such additions in the future.

Future Work Going forward, we envision MEDS serving as the bedrock for an EHR-native deep learning ecosystem that brings the same ease of use and composability to healthcare data that frameworks like HuggingFace have brought for NLP. This will eventually include software tools and best practices for model development, training, and evaluation, which will emerge organically through community use. We hope these efforts bring together a community of researchers excited by the prospect of bringing the latest advancements in deep learning to healthcare, and united by a set of shared best practices and standards which are currently absent from the field.

7 Acknowledgements

R. van de Water is funded by the "Gemeinsamer Bundesausschuss (G-BA) Innovationsausschuss" the framework of "CASSANDRA - Clinical ASSist AND aleRt Algorithms (project number 01VSF20015)". MBAM gratefully acknowledges support from a Berkowtiz Postdoctoral Fellowship at Harvard Medical School.

References

Rachael L Fleurence, Lesley H Curtis, Robert M Califf, Richard Platt, Joe V Selby, and Jeffrey S Brown. Launching pcornet, a national patient-centered clinical research network. *Journal of the American Medical Informatics Association*, 21(4):578–582, 2014.

Lin Lawrence Guo, Jason Fries, Ethan Steinberg, Scott Lanyon Fleming, Keith Morse, Catherine Aftandilian, Jose Posada, Nigam Shah, and Lillian Sung. A multi-center study on the adaptability of a shared foundation model for electronic health records. *arXiv preprint arXiv:2311.11483*, 2023.

Shih-Cheng Huang, Zepeng Huo, Ethan Steinberg, Chia-Chun Chiang, Matthew P. Lungren, Curtis Langlotz, Serena Yeung, Nigam Shah, and Jason Alan Fries. Inspect: A multimodal dataset for pulmonary embolism diagnosis and prognosis, 2023. URL https://openreview.net/forum?id=3sRR2u72oQ.

Kyunghoon Hur, Jungwoo Oh, Junu Kim, Jiyoun Kim, Min Jae Lee, Eunbyeol Cho, Seong-Eun Moon, Young-Hak Kim, Louis Atallah, and Edward Choi. Genhpf: General healthcare predictive framework for multi-task multi-source learning. *IEEE Journal of Biomedical and Health Informatics*, 2023.

Daniel Jarrett, Jinsung Yoon, Ioana Bica, Zhaozhi Qian, Ari Ercole, and Mihaela van der Schaar. Clairvoyance: A pipeline toolkit for medical time series. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=xnC8YwKUE3k.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

Alistair EW Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, et al. Mimic-iv, a freely accessible electronic health record dataset. *Scientific data*, 10(1):1, 2023.

Jeffrey G Klann, Lori C Phillips, Christopher Herrick, Matthew AH Joss, Kavishwar B Wagholikar, and Shawn N Murphy. Web services for data warehouses: Omop and pcornet on i2b2. *Journal of the American Medical Informatics Association*, 25(10):1331–1338, 2018.

Rupa Makadia and Patrick B Ryan. Transforming the premier perspective® hospital database into the observational medical outcomes partnership (omop) common data model. *Egems*, 2(1), 2014.

Matthew McDermott, Bret Nestor, Peniel Argaw, and Isaac S Kohane. Event stream gpt: a data pre-processing and modeling library for generative, pre-trained transformers over continuous-time sequences of complex events. *Advances in Neural Information Processing Systems*, 36, 2024.

Matthew B. A. McDermott, Shirly Wang, Nikki Marinsek, Rajesh Ranganath, Luca Foschini, and Marzyeh Ghassemi. Reproducibility in machine learning for health research: Still a ways to go. *Science Translational Medicine*, 13(586):eabb1655, 2021. doi: 10.1126/scitranslmed.abb1655. URL https://www.science.org/doi/abs/10.1126/scitranslmed.abb1655.

Shawn N Murphy, Griffin Weber, Michael Mendis, Vivian Gainer, Henry C Chueh, Susanne Churchill, and Isaac Kohane. Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2). *Journal of the American Medical Informatics Association*, 17(2):124–130, 2010.

Tom J Pollard, Alistair EW Johnson, Jesse D Raffa, Leo A Celi, Roger G Mark, and Omar Badawi. The eicu collaborative research database, a freely available multi-center database for critical care research. *Scientific data*, 5(1):1–13, 2018.

Evgeny S Saveliev and Mihaela van der Schaar. Temporai: Facilitating machine learning innovation in time domain tasks for medicine. *arXiv preprint arXiv:2301.12260*, 2023.

Nigam H. Shah, John D. Halamka, Suchi Saria, Michael Pencina, Troy Tazbaz, Micky Tripathi, Alison Callahan, Hailey Hildahl, and Brian Anderson. A Nationwide Network of Health AI Assurance Laboratories. 331(3):245–249, 2024. ISSN 0098-7484. doi: 10.1001/jama.2023.26930. URL `https://doi.org/10.1001/jama.2023.26930`.

Ethan Steinberg, Ken Jung, Jason A. Fries, Conor K. Corbin, Stephen R. Pfohl, and Nigam H. Shah. Language models are an effective representation learning technique for electronic health record data. *Journal of Biomedical Informatics*, 113:103637, 2021. ISSN 1532-0464. doi: https://doi.org/10.1016/j.jbi.2020.103637. URL `https://www.sciencedirect.com/science/article/pii/S1532046420302653`.

Ethan Steinberg, Yizhe Xu, Jason Alan Fries, and Nigam Shah. Motor: A time-to-event foundation model for structured medical records, 2024. URL `https://openreview.net/forum?id=NialiwI2V6`.

Shengpu Tang, Parmida Davarmanesh, Yanmeng Song, Danai Koutra, Michael W Sjoding, and Jenna Wiens. Democratizing ehr analyses with fiddle: a flexible data-driven preprocessing pipeline for structured clinical data. *Journal of the American Medical Informatics Association*, 27(12): 1921–1934, 2020.

Robin van de Water, Hendrik Nils Aurel Schmidt, Paul Elbers, Patrick Thoral, Bert Arnrich, and Patrick Rockenschaub. Yet Another ICU Benchmark: A Flexible Multi-Center Framework for Clinical ML. *International Conference on Learning Representations*, 2024. URL `https://arxiv.org/abs/2306.05109`.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

Michael Wornow, Yizhe Xu, Rahul Thapa, Birju Patel, Ethan Steinberg, Scott Fleming, Michael A. Pfeffer, Jason Fries, and Nigam H. Shah. The shaky foundations of large language models and foundation models for electronic health records. 6(1):1–10, 2023. ISSN 2398-6352. doi: 10.1038/s41746-023-00879-8. URL `https://www.nature.com/articles/s41746-023-00879-8`.

Michael Wornow, Rahul Thapa, Ethan Steinberg, Jason Fries, and Nigam Shah. Ehrshot: An ehr benchmark for few-shot evaluation of foundation models, 2024.

Chaoqi Yang, Zhenbang Wu, Patrick Jiang, Zhen Lin, Junyi Gao, Benjamin Danek, and Jimeng Sun. PyHealth: A deep learning toolkit for healthcare predictive modeling. In *Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD) 2023*, 2023. URL `https://github.com/sunlabuiuc/PyHealth`.

A CODE

We supply several code repositories to enable development with MEDS:

- The schema: `https://github.com/Medical-Event-Data-Standard/meds`
- An ETL package: `https://github.com/Medical-Event-Data-Standard/meds_etl`
- Transformations: `https://github.com/Medical-Event-Data-Standard/meds_transformations`
- A sample model: `https://github.com/Medical-Event-Data-Standard/sample_meds_model`

B Sample Data

```
>>> print(meds_dataset[0])
{
    ’patient_id’: 3,
    ’static_measurements’: [
        {
            {"code": "SNPs/rs429358"},
            {"race": "race/white"},
        }
    ],
    ’events’: [
        {
            "time": datetime(2010, 2, 3, 11, 52, 13),
            "measurements": [
                {"code": "ADMISSION"},
                {"code": "LOINC/...", "numeric_value": ...},
                {"code": "LOINC/...", "text_value": ...},
            ]
        }, {
            "time": datetime(2010, 2, 4, 6, 10, 0),
            "measurements": [
                {"code": "LOINC/...", "numeric_value": ...},
                {"code": "LOINC/...", "numeric_value": ...},
                {"code": "LOINC/...", "text_value": ...},
            ]
        }, {
            "time": datetime(2010, 2, 5, 15, 0, 0),
            "measurements": [
                {"code": "DISCHARGE"},
                {"code": "ICD10CM/..."},
                {"code": "ICD10CM/..."},
            ]
        },
        ...
    ],
}
```

Figure 3: An example `patient` data row from a MEDS-compliant dataset.

C Metadata Schema

```
# The dataset metadata schema.

# This is a JSON Schema (see https://json-schema.org/)

# Code specific metadata
code_metadata_entry = {
    "type": "object",
    "properties": {
        # A text description of the code
        "description": {"type": "string"},

        # Mappings between a code and either higher level codes or codes
    within standard ontologies
        "parent_codes": {"type": "array", "items": {"type": "string"}},
    },
}

# This is a simple code string to metadata entry map
code_metadata = {
    "type": "object",
    "additionalProperties": code_metadata_entry,
}

# Overall dataset metadata
dataset_metadata = {
    "type": "object",
    "properties": {
        "dataset_name": {"type": "string"},
        "dataset_version": {"type": "string"},
        "etl_name": {"type": "string"},
        "etl_version": {"type": "string"},
        "code_metadata": code_metadata,
    },
}
```

Figure 4: The MEDS `metadata` schema which stores general dataset specific metadata.