Machine Learning Model Specification for Cataloging Spatio-Temporal Models (Demo Paper)

Francis Charette-Migneault*
francis.charette-migneault@crim.ca
Computer Research Institute of Montréal (CRIM)

Ryan Avery*
ryan@wherobots.com
Wherobots Inc.

Brian Pondi
brian.pondi@uni-muenster.de
Institute for Geoinformatics, University of Münster

Joses Omojola
jomojo1@arizona.edu
University of Arizona

Simone Vaccari
simone.vaccari@terradue.com
Terradue

Parham Membari
parham.membari@terradue.com
Terradue

Devis Peressutti
devis.peressutti@planet.com
Sinergise Solutions, a Planet Labs company

Jia Yu
jiayu@wherobots.com
Wherobots Inc.

Jed Sundwall
jed@radiant.earth
Radiant Earth

Abstract

The Machine Learning Model (MLM) extension is a specification that extends the SpatioTemporal Asset Catalogs (STAC) framework to catalog machine learning models. This demo paper introduces the goals of the MLM, highlighting its role in improving searchability and reproducibility of geospatial models. The MLM is contextualized within the STAC ecosystem, demonstrating its compatibility and the advantages it brings to discovering relevant geospatial models and describing their inference requirements.

A detailed overview of the MLM's structure and fields describes the tasks, hardware requirements, frameworks, and inputs/outputs associated with machine learning models. Three use cases are presented, showcasing the application of the MLM in describing models for land cover classification and image segmentation. These examples illustrate how the MLM facilitates easier search and better understanding of how to deploy models in inference pipelines.

The discussion addresses future challenges in extending the MLM to account for the diversity in machine learning models, including foundational and fine-tuned models, multi-modal models, and the importance of describing the data pipeline and infrastructure models depend on. Finally, the paper demonstrates the potential of the MLM to be a unifying standard to enable benchmarking and comparing geospatial machine learning models.

CCS Concepts

• **Information systems** → **Spatial-temporal systems**; *Information retrieval*; • **Computing methodologies** → Machine learning.

*Both authors contributed equally to this research.

ACM SIGSPATIAL 2024, October 29–November 1, 2024, Atlanta, GA, USA

ACM ISBN 979-8-4007-1148-0/24/10
https://doi.org/10.1145/3681769.3698586

Keywords

STAC, Catalog, Machine Learning, Spatio-Temporal Models, Search

ACM Reference Format:
Francis Charette-Migneault, Ryan Avery, Brian Pondi, Joses Omojola, Simone Vaccari, Parham Membari, Devis Peressutti, Jia Yu, and Jed Sundwall. 2024. Machine Learning Model Specification for Cataloging Spatio-Temporal Models (Demo Paper). In *3rd ACM SIGSPATIAL International Workshop on Searching and Mining Large Collections of Geospatial Data (GeoSearch'24), October 29–November 1 2024, Atlanta, GA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3681769.3698586

1 Introduction

1.1 The Challenge of Cataloging Spatio-Temporal Models

Identifying the right Machine Learning (ML) model for spatio-temporal data — data that varies across both space and time — presents notable challenges, primarily due to the lack of standardized descriptions in existing model catalogs. While platforms like Hugging Face [4], DLHub [5] and MLFlow [6] offer task-based filtering and model versioning, they fall short when it comes to the specific requirements of geospatial applications. These platforms do not inherently support filtering based on temporal dependencies or geographic scope, making them less suitable for applications like environmental monitoring, agriculture, disaster response, and urban planning. This limitation hampers effective search and discovery, and even when relevant models are found, inadequate documentation often hinders their proper application in inference pipelines.

1.2 How the MLM addresses these problems?

To address mentioned limitations with existing model catalogs, we introduce the Machine Learning Model[1] (MLM) extension, a standard extending the SpatioTemporal Asset Catalogs Specification (STAC). The MLM provides critical fields for both discovering relevant models and documenting their runtime requirements.

[1]https://github.com/stac-extensions/mlm

The core contributions of the MLM provide:

(1) a comprehensive schema to describe fields that are commonly used for geospatial model search and discovery. These include the relevant geographic domain of the model described as GeoJSON, the relevant temporal range based on it's training data, and relation types that describe how an ML model was developed from one or more geospatial datasets;
(2) critical fields required for model inference reproducibility, which include data provenance, input/output structures, data preparation and processing workflows.

Each of these contributions enable:

(1) building model collections that can be searched by dataset, geography, time, and attributes relevant to developers, machine learning practitioners and researchers;
(2) recording the metadata necessary to run model inference on real data for tasks it was trained to perform.

By incorporating this new specification into the larger ecosystem[2] of the STAC specification for describing spatio-temporal data, MLM enhances the reproducibility and reusability of published models utilizing such datasets. By design, MLM supports FAIR (findable, accessible, interoperable, and reusable) principles[3]. It achieves this through a dual approach:

(1) flat fields related to model and data provenance facilitate efficient search and discovery;
(2) nested objects organize complex model metadata essential for reproducing runtime requirements.

The MLM extension has evolved over many iterations, including building upon lessons learned from previous standardization efforts of DLM[4] [1] and `ml-model`[5], to account for the large diversity of sensor data, machine learning tasks, and implementations of different frameworks for encoding model artifacts. The geospatial community needs a standard that can support the growing diversity of models and MLM can serve as that standard.

1.3 Benefits of extending STAC to handle machine learning models

STAC is a standard for describing datasets with a space and time component, usually overhead imagery or other overhead sensor data. The STAC specification natively supports extensions to support different use cases, such as describing particular kinds of data (point cloud, radar, rasters, climate variables), additional metadata (scientific references, source satellite parameters) or other assets (precomputed bands, human annotations on satellite imagery, features of interest).

As a STAC extension, MLM's metadata fields are compatible with the rest of the STAC ecosystem, which is most commonly used for describing datasets. For example, the description of categories that a model predicts is interoperable with other STAC datasets that define relationships to those categories, like the STAC Label extension[6].

When the extension is used in a catalog, the flatness of fields describing search and discovery metadata enable faster queries. Deeply nested JSON objects complicate queries, and the absence of comprehensive metadata and proper indexing can make locating specific data files increasingly difficult over time[7]. Researchers and developers searching for relevant models often need to locate more than just the model artifact. They also typically require metadata on the following non-exhaustive list of metadata useful in search and discovery:

(1) training dataset product IDs to understand the sensor domain of the model;
(2) task type to understand the utility of the model (e.g. classification, segmentation, detection, etc.);
(3) categories that are satisfied by the model task, if the model includes a supervised classifier;
(4) the specific geographies and temporal ranges of the training dataset of a model.

Considering that model datasets can be constrained to specific geographies and their weights are trained to resolve single/multiple tasks, identifying relevant models easily can accelerate development for Geospatial Information Systems (GIS) and Earth Observation (EO) practitioners.

Providing this information in a standardized format allows users to execute complex queries like searching and ranking models trained using datasets that intersect specific geographic areas. Additionally, MLM fields improve spatiotemporal similarity between datasets by allowing users to find more data related to a single query. Queries can be routed through existing STAC APIs, and model artifacts can be hosted on cloud databases and data lakes for improved retrieval and accelerated development.

2 Overview

A MLM definition relies on its dependency on STAC core metadata definitions and its native interoperability with GeoJSON documents. In other words, many of its core properties, such as `start_datetime`, `end_datetime`, `bbox`, `geometry`, etc., are employed by MLM to indicate data domain constraints for which the model is vetoed to have sufficient domain knowledge or training examples to perform reliably on new inference data[8]. Similarly, `links` definitions with `derived_from` relation-types and `roles` established by MLM serve to identify reference Analysis-Ready Data employed to train the model, which are typically themselves represented by STAC collections.

Whenever possible, MLM relies on other `stac_extensions` indicated within its STAC item definition to provide additional metadata attributes already defined by those references. This allows MLM to take advantage of existing properties, assets, and field definitions, which facilitates reuse by clients that already support them, as well as generating derived MLM definitions for models that were trained using source datasets employing those STAC extensions. Notably, MLM supports the following extensions. The `ml-aoi`[9] and `label`[10] extensions are employed to describe the annotations involved in

[2] https://stacindex.org/ecosystem
[3] https://www.go-fair.org/fair-principles/
[4] https://github.com/crim-ca/dlm-extension
[5] https://github.com/stac-extensions/ml-model
[6] https://github.com/stac-extensions/label
[7] https://github.com/radiantearth/stac-spec/blob/v1.0.0/best-practices.md
[8] https://github.com/stac-extensions/mlm/blob/main/best-practices.md
[9] https://github.com/stac-extensions/ml-aoi
[10] https://github.com/stac-extensions/label

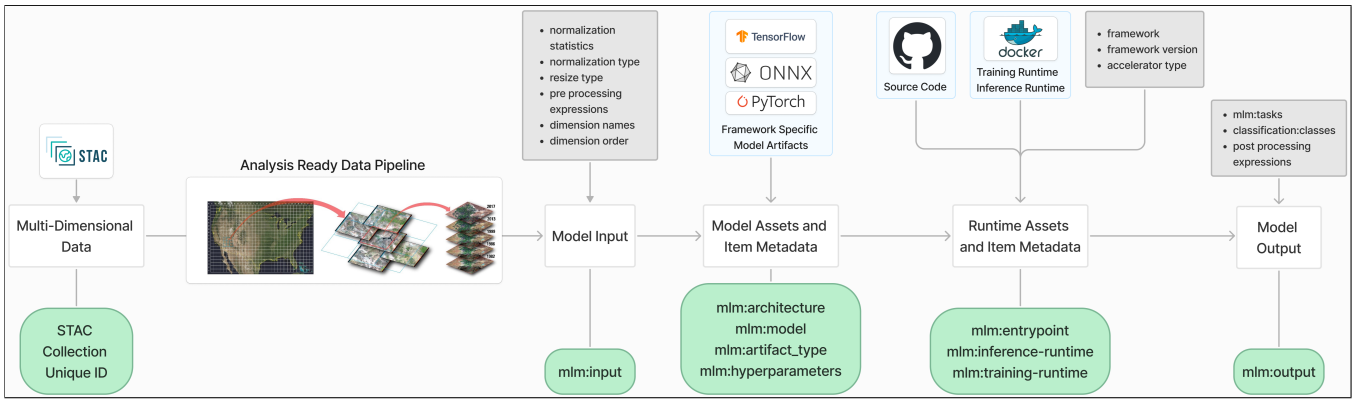

Figure 1: Overview of the STAC Machine Learning Model extension and the representation of its field metadata.

training the model. The `classification`[11] extension is used to describe the MLM outputs, which correspond to the training data ground truth. The `scientific`[12] extension is employed to provide citations for sharing, reusing, or employing the model, along with other core attributes such as `license`. The dependencies on specific satellite imagery rely on definitions from `eo`[13], `raster`[14], or STAC 1.1 `bands`[15] to define the mapping between model inputs and existing STAC data sources. Finally, a combination of `processing`[16], `file`[17], `example`[18], and `version`[19] extensions are employed to provide runtime requirements to execute the model, such as details regarding pre/post-processing steps, sample code, and versioned dependencies to model checkpoints, files, weights, or any other relevant data needed for its distribution and operation.

For any attributes that are not already defined by another STAC extension or that are specific only to the model, MLM fields[20] were created. The most notable entries are shown on Figure 1. Namely, the `mlm:input` characterizes the input structure of the model, which depends on the source bands (as applicable), their data types, dimensions, ordering, and any relevant pre-processing operations such as normalization and resizing to fit the model input requirements from Analysis-Ready Data. It's worth noting that this `mlm:input` can accommodate data input structures beyond individual raster images. Other dimensions than batch, bands, height, and width may be specified to accommodate time series inputs, variables with an elevation dimension or more complex multi-modal data. The `mlm:output` describes the tasks being predicted, which classification predictions can be generated, and the format under which they are returned, also including post-processing operations as needed. In between, the model itself is described with a combination of its relevant assets (weights, checkpoints, etc.), including runtime details to facilitate the replication of its execution. The MLM extension does not impose any runtime, and instead offers flexibility to accommodate virtually any ML framework possible.

To help users define the necessary properties to correctly describe a model, while respecting MLM JSON schema validation, the `stac-model`[21] utility is provided along with the MLM definitions. This tool can natively be employed with `pydantic`[22] and `pystac`[23], commonly used for building Web APIs, schema validators, and interacting with STAC in Python.

3 Experimental Demonstrations

This section briefly presents some applications making use of MLM to describe models. For each example, readers are invited to visit supplementary material referenced by the application to obtain an in-depth overview of features provided by MLM.

3.1 EuroSAT Image Classification with a ResNet Model

Wherobots uses the MLM to promote interoperability of open-source models with its Raster Inference[24] product, SQL, and Python APIs for running batch inference on georeferenced overhead imagery. These models and MLM metadata are hosted on Wherobots Cloud and Hugging Face[25] as examples of how the MLM describes different models for inference.

One inference example that the MLM enables is running a land cover classification model on the EuroSAT [2, 3] dataset, which contains Sentinel-2 satellite imagery and single label categories associated with each image. A guide for running a sample model for land cover classification is available in the inference tutorial[26].

[11] https://github.com/stac-extensions/classification
[12] https://github.com/stac-extensions/scientific
[13] https://github.com/stac-extensions/eo
[14] https://github.com/stac-extensions/raster
[15] https://github.com/radiantearth/stac-spec/blob/v1.1.0/commons/common-metadata.md#bands
[16] https://github.com/stac-extensions/processing
[17] https://github.com/stac-extensions/file
[18] https://github.com/stac-extensions/example-links
[19] https://github.com/stac-extensions/version
[20] https://github.com/stac-extensions/mlm#item-properties-and-collection-fields
[21] https://github.com/stac-extensions/mlm/blob/main/README_STAC_MODEL.md
[22] https://github.com/pydantic/pydantic
[23] https://github.com/stac-utils/pystac
[24] https://wherobots.com/wherobotsai-for-raster-inference/
[25] https://huggingface.co/wherobots/mlm-stac/blob/main/classification/landcover-eurosat-sentinel2/model-metadata.json
[26] https://docs.wherobots.com/latest/tutorials/wherobotsai/wherobots-inference/classification/

3.2 SATLAS Image Segmentation with a SwinTransformer based model

SATLAS is a project that trained a foundational model, a model that has seen lots of data modalities, and then trained task-specific models on top of this base model for various use cases.

An example notebook[27] highlights how to run a pixel segmentation model from the SATLAS[28] project. The notebook walks through the process of creating and validating model metadata that complies with the MLM, saving the JSON to S3 cloud storage, and then passing this JSON to SQL functions that handle loading MLM Model Artifact to run batch inference with the model. The resulting metadata for the model can be found on Hugging Face[29].

3.3 Water-Bodies Classification from Annotated STAC Datasets

Terradue demonstrates in their blog post[30] how the MLM extension and its metadata enabled extending their Geohazard Thematic Exploitation Platform (TEP)[31] systems, with AI capabilities.

The blog post presents a walkthrough of the steps involved to create a ML model for water-bodies masking, from data discovery in STAC catalogues, training the model, describing it using the MLM extension, to deploying it on a TEP for inference on EO data. These steps are illustrated with practical examples via Jupyter Notebooks and reference implementations. Each element in the process contributes to a FAIR pipeline by leveraging metadata from every component. The blog post highlights how MLM ensures adequate provisioning of reference metadata and links to derived data, allowing end-users to make informed decisions.

4 Discussion

The initial focus of MLM is to support FAIR principles. In that regard, providing sufficient and relevant metadata to identify appropriate models for a given task or objective, while respecting data domain constraints, is effectively supported with MLM.

Using MLM and the tools it provides, runtime dependencies, accelerator requirements, and input/output data structures allow reliable reuse of the described models for inference. Through presented examples, MLM also demonstrates its capability to be adapted to multiple applications. Its capability to work simultaneously with geo-referenced data, non-geospatial data, variable structures and dimensions of inputs and outputs, and its extensibility by relying on other STAC extensions, allows it to be applied for cross-domain and multi-task models effortlessly, without diminishing the granularity of attribute details required by each task.

However, MLM remains fairly limited in terms of training definition. While MLM provides means to indicate cross-references to training data, splits, hyperparameters, and other relevant attributes employed for training the model, it does not dive deeper into the requirements and subtleties involved in the definition of the training pipeline itself to perfectly replicate the training experiment. As of MLM `v1.2.0`, users can provide implementation references to the applied pipeline to train the described model, such as using a Docker image or linking to a Git repository, but connecting the dots between the MLM definition, the source datasets and the reference runtime is left up to the users working with the model. Future work could improve support of the training runtime with enhanced definitions, additional attributes to replicate training experiments, or by relying on another STAC extension dedicated for this purpose, while ensuring interoperability with MLM at its current stage.

[27] https://github.com/wherobots/wbc-examples/blob/main/python/raster-inference/byom_example.ipynb
[28] https://satlas.allen.ai/
[29] https://huggingface.co/wherobots/mlm-stac/blob/main/semantic-segmentation/solar-satlas-sentinel2/model-metadata.json
[30] https://discuss.terradue.com/t/1188/11
[31] https://geohazards-tep.eu/

5 Conclusion

In this paper, we highlight an absence of standards for efficient cataloging of geospatial models. Many models trained on geospatial imagery or other multidimensional data do not contain the necessary details to inform search, including the geographic domain, relationship to datasets they were derived from, and the temporal range for which they are relevant. In addition, models are typically not published with enough detail to reproduce their inference requirements. Missing information often includes the interpretation of model input and output parametrization, the accelerators required to run models, and the tasks that models fulfill.

Using the MLM enables improving search of models described with it and makes construction of inference pipelines easier to understand, once the right model has been located. Its support for describing complex input and output structures, and its relationship to the larger STAC ecosystem, allows to quickly discover which sensors should be used in conjunction with a model and how they must be processed for successful inference. The MLM's flexibility allows for better representation of complex model architectures and data processing pipelines, facilitating more effective model discovery, deployment, and interoperability across different platforms.

Furthermore, the MLM specification has been designed to accommodate current trends in geospatial model development, such as supporting multi-modal data and performing many tasks simultaneously. Readers are invited to try out the MLM when cataloging their model metadata in databases, through STAC APIs or static storage archives, henceforth facilitating sharing and reusing them.

References

[1] Samuel Foucher, Francis Charette-Migneault, and David Landry. 2020. *Project CCCOT03: Proposal for a STAC Extension for Deep Learning Models.* Technical Report. Computer Research Institute of Montréal (CRIM). https://doi.org/10.13140/RG.2.2.27858.68804

[2] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. 2018. Introducing EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification. In *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium.* IEEE, 204–207.

[3] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. 2019. EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* (2019). https://github.com/phelber/EuroSAT

[4] Jason Jones, Wenxin Jiang, Nicholas Synovic, George K. Thiruvathukal, and James C. Davis. 2024. What do we know about Hugging Face? A systematic literature review and quantitative validation of qualitative claims. arXiv:2406.08205 [cs.SE] https://arxiv.org/abs/2406.08205

[5] Zhuozhao Li, Ryan Chard, Logan Ward, Kyle Chard, Tyler J Skluzacek, Yadu Babuji, Anna Woodard, Steven Tuecke, Ben Blaiszik, Michael J Franklin, et al. 2021. DLHub: Simplifying publication, discovery, and use of machine learning models in science. *J. Parallel and Distrib. Comput.* 147 (2021), 64–76.

[6] Matei Zaharia, Andrew Chen, Aaron Davidson, Ali Ghodsi, Sue Ann Hong, Andy Konwinski, Siddharth Murching, Tomas Nykodym, Paul Ogilvie, Mani Parkhe, et al. 2018. Accelerating the machine learning lifecycle with MLflow. *IEEE Data Eng. Bull.* 41, 4 (2018), 39–45.