Open Geospatial Consortium

Submission Date: <yyyy-mm-dd>

Approval Date: <yyyy-mm-dd>

Publication Date: <yyyy-mm-dd>

External identifier of this OGC® document: http://www.opengis.net/doc/{doc-type}/{standard}/{m.n}

Internal reference number of this OGC® document: YY-nnnrx

Version: n.n.n

Editor: <Name(s) of Editor or Editors>

OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part 3: XML Encoding Standard

Warning

This document is not an OGC Standard. This document is distributed for review and comment. This document is subject to change without notice and may not be referred to as an OGC Standard.

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type:	OGC® Standard
Document subtype:	if applicable
Document stage:	Draft
Document language:	English

License Agreement

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD.

THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER'S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR's sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications. This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

Table of Contents

i. Abstract

The OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part 3: XML Encoding Standard defines requirements for encoding AI training datasets as XML. Extensible Markup Language (XML) is a widely utilized format for encoding data in various applications. It employs a markup structure where data is enclosed in tags represented by opening and closing elements. These elements can have attributes and may contain nested elements, allowing for a hierarchical representation of structured information. TrainingDML-AI Part 3 is based on the OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part 1: Conceptual Model Standard.

ii. Keywords

The following are keywords to be used by search engines and document catalogues.

ogcdoc, OGC document, artificial intelligence, machine learning, deep learning, earth observation, remote sensing, training data, training sample, encoding, XML

iii. Preface

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

iv. Security Considerations

No security considerations have been made for this Standard.

v. Submitting organizations

The following organizations submitted this Document to the Open Geospatial Consortium (OGC):

Organization name(s)

vi. Submitters

All questions regarding this submission should be directed to the editor or the submitters:

Name	Affiliation

vii. Contributors

Additional contributors to this Standard include the following:

Individual name(s), Organization

Chapter 1. Scope

This OGC TrainingDML - AI Part 3: XML Encoding Standard defines an XML encoding for the exchange of training datasets. The TrainingDML - AI Part 3 Standard provides an XML-based encoding for the exchange of information describing training datasets, both within and between different organizations.

The document model is derived from the conceptual models defined in the OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part 1: Conceptual Model Standard.

Chapter 2. Conformance

This Standard defines an XML encoding for AI training datasets. The standardization target for this Standard is:

- TrainingDML-AI XML Encoding Schema

Conformance with this Standard shall be checked using all the relevant tests specified in Annex A (normative) of this document. The framework, concepts, and methodology for testing, and the criteria to be achieved to claim conformance are specified in the OGC Compliance Testing Policies and Procedures and the OGC Compliance Testing website.

All requirements-classes and conformance-classes described in this document are owned by the standard identified.

Chapter 3. Normative References

The following normative documents contain provisions that, through reference in this text, constitute provisions of this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

- OGC: OGC 23-008r3, OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part1: Conceptual Model Standard, 2023
- OGC: OGC 07-036, OpenGIS Geography Markup Language (GML) Encoding Standard, 2007
- W3C Recommendation: W3C XML Schema Definition Language (XSD) 1.1 Part 1: Structures, 2012
- W3C Recommendation: W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes, 2012
- IETF: RFC 2046, Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types, 1996
- ISO 19107:2019 Geographic information — Spatial schema
- ISO 19115-1:2014 Geographic information — Metadata — Part 1: Fundamentals
- ISO 19157-1:2023 Geographic information — Data quality — Part 1: General requirements

Chapter 4. Terms and Definitions

This document used the terms defined in OGC Policy Directive 49, which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this Standard and OGC documents do not use the equivalent phrases in the ISO/IEC Directives, Part 2.

For the purposes of this document, the following additional terms and definitions apply.

4.1. Artificial Intelligence (AI)

refers to a set of methods and technologies that can empower machines or software to learn and perform tasks like humans.

SOURCE: OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part 1: Conceptual Model Standard

4.2. Machine Learning (ML)

is an important branch of artificial intelligence that gives computers the ability to improve their performance without explicitly being programmed to do so. ML processes create models from training data by using a set of learning algorithms, and then can use these models to make predictions. Depending on whether the training data include labels, the learning algorithms can be divided into supervised and unsupervised learning.

SOURCE: OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part 1: Conceptual Model Standard

4.3. Deep Learning (DL)

is a subset of machine learning, which is essentially a neural network with three or more layers. The number of layers is referred to as depth. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy.

SOURCE: https://www.ibm.com/topics/deep-learning

4.4. Dataset

identifiable collection of data

```
Note 1 to entry: A dataset can be a smaller grouping of data which, though limited by
some constraint such as spatial extent or feature type, is located physically within a
larger dataset. Theoretically, a dataset can be as small as a single feature or
feature attribute contained within a larger dataset. A hardcopy map or chart can be
```

```
considered a dataset.
```

[SOURCE: ISO 19115-1:2014, 4.3]

4.5. Training Dataset

is a collection of samples that have been derived or generated from reliable geospatial data sources, often labeled in terms of supervised learning. A training dataset can be divided into training, validation, and test sets. Training samples are different from samples in OGC Observations & Measurements (O&M). They are often collected in purposive ways that deviate from purely probability sampling, with known or expected results labeled as values of a dependent variable for generating a trained predictive model.

```
Note 1 to entry: the quality of data source considered higher than the AI data.
```

SOURCE: OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part 1: Conceptual Model Standard

4.6. Label

refers to known or expected results annotated as values of a dependent variable in training samples. A training sample label is different from those on a geographical map, which are known as map labels or annotations.

SOURCE: OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part 1: Conceptual Model Standard

4.7. Class

the known or expected category labeled on training samples that share some common attributes.

SOURCE: OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part 1: Conceptual Model Standard

4.8. Task

the specific goal that an AI application want to achieve.

SOURCE: OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part 1: Conceptual Model Standard

4.9. Provenance

information about entities, activities, and people involved in producing a piece of data or thing, which can be used to form assessments about its quality, reliability or trustworthiness. In this standard provenance is a record of how training data were prepared.

SOURCE: W3C (https://www.w3.org/TR/prov-overview/)

4.10. Quality

degree to which a set of inherent characteristics of an object fulfils requirements [ISO 9000:2015, 3.6.2, modified—Notes 1 and 2 to entry have been deleted]. Quality of training data (such as data imbalance and mislabeling) can impact the performance of AI/ML models.

SOURCE: OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part 1: Conceptual Model Standard

4.11. Earth Observation

data and information collected about our planet, whether atmospheric, oceanic or terrestrial. This includes space-based or remotely-sensed data, as well as ground-based or in situ data.

SOURCE: GEO (https://earthobservations.org/geo_wwd.php)

4.12. Scene Classification

task of identifying scene categories of images, on the basis of a training set of images whose scene categories are known.

SOURCE: OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part 1: Conceptual Model Standard

4.13. Object Detection

task of recognizing objects such as cars from images. The objects are often localized using bounding boxes.

SOURCE: OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part 1: Conceptual Model Standard

4.14. Semantic Segmentation

task of assigning class labels to pixels of images or points of point clouds.

SOURCE: OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part 1: Conceptual Model Standard

4.15. Change Detection

task that finds the changes in an area between images taken at different times.

SOURCE: OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part 1: Conceptual Model Standard

4.16. 3D Model Reconstruction

task that builds 3D objects and scenes from multi-view images.

SOURCE: OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part 1: Conceptual Model Standard

4.17. Generative Model

is one of the methods of large model training, which improve model performance through unsupervised pre-training. In the fine-tuning phase, labeled data plays a critical role in optimizing the model for specific vertical domains or tasks. By incorporating labeled data, the model can learn to accurately identify and extract relevant features, leading to better performance on specific downstream tasks. Overall, the combination of generative models and fine-tuning with labeled data can significantly improve the performance of large models in specialized domains or tasks.

SOURCE: OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part 1: Conceptual Model Standard

SOURCE: OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part 1: Conceptual Model Standard

4.18. Extensible Markup Language (XML)

is a simple, very flexible text format derived from SGML (ISO 8879). Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere.

SOURCE: https://www.w3.org/XML/

4.19. XML Schema

is a means for defining the structure, content and semantics of XML documents.

SOURCE: https://www.w3.org/XML/Schema

4.20. Training Dataset Publisher

refers to the entity or individual responsible for creating and releasing the XML-based serialization syntax for geospatial training datasets, as defined in the TrainingDML-AI Part 3: XML Encoding Standard.

Chapter 5. Conventions

This section provides details and examples for any conventions used in the document.

5.1. Identifiers

The normative provisions in this Standard are denoted by the URI:

http://www.opengis.net/spec/TrainingDML-AI-3/1.0

All requirements and conformance tests that appear in this document are denoted by partial URIs which are relative to this base.

5.2. Abbreviated Terms

In this document the following abbreviations and acronyms are used or introduced:

- AI — Artificial Intelligence
- DL — Deep Learning
- EO — Earth Observation
- IETF — Internet Engineering Task Force
- ISO — International Organization for Standardization
- JSON — JavaScript Object Notation
- ML — Machine Learning
- OGC — Open Geospatial Consortium
- RS — Remote Sensing
- TD — Training Data
- UML — Unified Modelling Language
- URL — Uniform Resource Locator
- UTC — Coordinated Universal Time
- W3C — World Wide Web Consortium
- XML — Extensible Markup Language

Chapter 6. Overview

The TrainingDML - AI Part 3: XML Encoding Standard defines a XML-based serialization syntax for geospatial training datasets. While other serialization formats are possible, such alternatives are not discussed in this Standard.

When serialized, absent properties in XML are typically represented by leaving the corresponding elements empty, without any content. Alternatively, certain XML-based formats may use attributes to signify the absence of a property. These representations convey semantic equivalence. An empty or missing element signifies that no value has been assigned, distinct from the interpretation that the provided value is empty or nil.

XML employs a formal class model through the use of elements and attributes, allowing for a more structured representation of data. XML documents are hierarchical in nature, composed of nested elements that can contain both data and metadata.

A training dataset document conforming to this Standard is an XML document whose root element is an "AI_TrainingDataset" element.

6.1. Extensible Markup Language

Extensible Markup Language (XML) is a versatile and widely used markup language designed for encoding documents in a format that is both human-readable and machine-readable. It provides a set of rules for defining custom markup languages and is often employed for representing structured data in a hierarchical and platform-independent manner. XML documents consist of elements, attributes, and text content organized within a tree-like structure. Elements are enclosed within tags, and attributes provide additional information about the elements.

XML has widespread applicability and is utilized in various domains, including web development, data interchange, configuration files, and more. Its flexibility allows users to define their own document structures, making it suitable for a broad range of applications. XML is supported by many programming languages and technologies, contributing to its interoperability and adoption.

Chapter 7. Requirements for TrainingDML-AI XML Encoding

7.1. Requirements Class: Base

7.1.1. Requirements Class: XML Base Type

The XML Base Type requirements class defines the base requirements for XML encodings, which includes definitions of common types used in the TrainingDML-AI XML encoding.

Requirements class	
/req/base/xmlbasetype	
Dependency	XML
Requirement 1	/req/base/xmlbasetype/xml
Requirement 2	/req/base/xmlbasetype/datetime
Requirement 3	/req/base/xmlbasetype/namedvalue
Requirement 4	/req/base/xmlbasetype/url

The first requirement is that a TrainingDML-AI XML document is a valid XML document.

Requirement 1	/req/base/xmlbasetype/xml An instance SHALL be a conformant XML document, as defined by W3C Recommendation

XML has a limited range of built-in types (https://www.w3.org/TR/xmlschema11-2/). The following requirements provide standard XML representations of additional types required across all requirements within this specification.

The DateTime is encoded as an XML element defined as one of three elements of type "data", "time", "dataTime".

Requirement 2	/req/base/xmlbasetype/datetime Each DateTime value SHALL be encoded as a text string defined in Section 3.3.7, Section 3.3.8 or Section 3.3.9 of W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes. The specification of date and time in any XML encoding of training set data SHALL be specified in UTC.

Examples:

```
a) 2022-08-08T08:08:00.00Z
b) 2022-08-08
c) 12:34:56
d) 12:34:56.123
```

The NamedValue is encoded as an XML element with two elements named “key” and “value”.

Requirement 3	/req/base/xmlbasetype/namedvalue Each NamedValue value SHALL be encoded as an XML element with elements "key" and "value", while the type of "key" is CharacterString.

Examples:

```
a) <key>forest</key>
   <value>RGB(0,255,255)</value>
```

```
b) <key>precision</key>
   <value>0.8</value>
```

The URL is encoded as an XML element defined as the type of “anyURI”.

Requirement 4	/req/base/xmlbasetype/url Each URL value SHALL be encoded as a text string defined in Section 3.3.17 of W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes.

Examples:

```
a) http://www.opengeospatial.org
```

```
b) /file.txt
```

7.1.2. Requirements Class: ISO Metadata Type

The ISO Metadata Type requirements class defines the requirements for XML encoding of ISO metadata types.

Requirements class	
/req/base/isometadatatype	
Dependency	XML
Requirement 5	/req/base/isometadatatype/band

Requirements class	
Requirement 6	/req/base/isometadatatype/extent
Requirement 7	/req/base/isometadatatype/citation
Requirement 8	/req/base/isometadatatype/scope

The MD_Band is encoded as a text string or an XML element.

Requirement 5	/req/base/isometadatatype/band Each MD_Band value SHALL be encoded as a text string or an XML element matching the XML Schema type as defined in: https://schemas.isotc211.org/19115/-1/mrc/1.3.0/content.xsd

Examples:

```
a)  red
b)  B4
c) <mrc:boundMax>690</mrc:boundMax>
   <mrc:boundMin>630</mrc:boundMin>
   <mrc:boundUnits>nm</mrc:boundUnits>
```

An The EX_Extent is encoded as a text string or an XML element.

Requirement 6	/req/base/isometadatatype/extent Each EX_Extent value SHALL be encoded as a text string or an XML element matching the XML Schema type as defined in: https://schemas.isotc211.org/19115/-1/gex/1.3.0/extent.xsd

Examples:

```
a)  120.0 30.0 130.0 40.0
b)  120.0 30.0 10.0 130.0 40.0 20.0
c) <geographicElement>
       <westBoundLongitude>-171.76409</westBoundLongitude>
       <eastBoundLongitude>-157.86768</eastBoundLongitude>
       <southBoundLatitude>-14.42443</southBoundLatitude>
       <northBoundLatitude>21.31573</northBoundLatitude>
   </geographicElement>
```

The CI_Citation is encoded as a text string or an XML element.

Requirement 7	/req/base/isometadatatype/citation Each CI_Citation value SHALL be encoded as a text string or an XML element matching the XML Schema type as defined in: https://schemas.isotc211.org/19115/-1/cit/1.3.0/citation.xsd

Examples:

```
a)  http://www.opengeospatial.org
b)  <cit:title>Open Geospatial Consortium</cit:title>
   <cit:alternateTitle>OGC</cit:alternateTitle>
   <cit:identifier>
       <cit:code>
           https://portal.ogc.org/files/?artifact_id=104605&version=1
       </cit:code>
   </cit:identifier>
```

The MD_Scope is encoded as an XML element.

Requirement 8	/req/base/isometadatatype/scope Each MD_Scope value SHALL be encoded as an XML element matching the XML Schema type as defined in: https://schemas.isotc211.org/19115/-1/mcc/1.3.0/commonClasses.xsd

Example:

```
<mcc:level>dataset</mcc:level>
<mcc:levelDescription>
    <mcc:dataset>whu_rs19</mcc:dataset>
</mcc:levelDescription>
```

7.1.3. Requirements Class: ISO Quality Type

The ISO Quality Type requirements class defines the requirements for XML encoding of ISO quality types.

Requirements class	
/req/base/isoqualitytype	
Dependency	XML
Requirement 9	/req/base/isoqualitytype/element

The QualityElement object is encoded as an XML element with properties shown in Table 1.

Requirement 9	/req/base/isoqualitytype/element Each QualityElement value SHALL be encoded as an XML element with properties shown in Table 1.

Table 1. QualityElement properties

XML Property	Definition	Data type and values	Obligation
type	The type of the quality element object.	CharacterString [1..1]	Mandatory
measure	Reference to measure used.	MeasureReference [1..1]	Mandatory
evaluationMethod	Evaluation information.	EvaluationMethod [1..1]	Mandatory
result	Value obtained from applying a data quality measure.	QualityResult [1..*]	Mandatory

Example:

```
<type>FormatConsistency</type>
<measure>
   <measureDescription>
      Percentage of training samples with inconsistent image format
   </measureDescription>
</measure>
<evaluationMethod>
   <evaluationMethodDescription>
      Full test method to calculate the percentage of training samples with an
inconsistent format
   </evaluationMethodDescription>
</evaluationMethod>
<result>
   <quantitativeResult>
    <value>
         0
      </value>
    <valueUnit>
         %
      </valueUnit>
   </quantitativeResult>
</result>
```

7.1.4. Requirements Class: Geospatial Type

The Geospatial Type requirements class defines the requirements for XML encoding of geospatial types.

Requirements class	
/req/base/geospatialtype	
Dependency	XML
Dependency	GML
Requirement 10	/req/geospatialtype/feature

The encoding of one or more features follows GML for encoding a Feature element, with members “type”, “geometry” and “properties”. A Feature element represents a spatially bounded thing. Every Feature element is a GML element no matter where it occurs in a GML text.

Requirement 10	/req/geospatialtype/feature Each Feature value SHALL be encoded using the GML feature encoding defined by OpenGIS in GML Section 3.2: http://www.opengis.net/gml/3.2

Examples of Feature encodings are:

```
a)  <gml:metaDataProperty>
    <gml:GenericMetaData>
        <truncated>0.0</truncated>
        <occluded>0</occluded>
        <alpha>-0.2</alpha>
    </gml:GenericMetaData>
   </gml:metaDataProperty>
   <gml:location>
    <gml:GeometricComplex>
        <gml:element>
            <gml:Polygon>
                <gml:exterior>
                    <gml:LinearRing>
                        <gml:posList>
                            51.556272 -0.2803943 51.5562758 -0.2787397 51.5556539
-0.278736 51.5556501 -0.2803906 51.556272 -0.2803943
                        </gml:posList>
                    </gml:LinearRing>
                </gml:exterior>
            </gml:Polygon>
        </gml:element>
    </gml:GeometricComplex>
   </gml:location>
```

```
b)  <gml:metaDataProperty>
       <gml:GenericMetaData>
           <iscrowd>0</iscrowd>
```

```
            <area>2580</area>
        </gml:GenericMetaData>
    </gml:metaDataProperty>
    <gml:location>
        <gml:GeometricComplex>
            <gml:element>
                <gml:Polygon>
                    <gml:exterior>
                        <gml:LinearRing>
                            <gml:posList>
                                274 1602 273 1603 272 1603 271 1603 270 1604 269 1604
268 1604 267 1604 266 1605 265 1605 264 1605 263 1606 262 1606 261 1606 260 1607 259
1607 258 1607 257 1608 256 1608 255 1608 254 1609 253 1610 252 1611 251 1611
                            </gml:posList>
                        </gml:LinearRing>
                    </gml:exterior>
                </gml:Polygon>
            </gml:element>
        </gml:GeometricComplex>
    </gml:location>
    <gml:boundedBy>
        <gml:Envelope>
            244 1602 306 1653
        </gml:Envelope>
    </gml:boundedBy>
```

7.2. Requirements Class: AI_TrainingDataset

The AI_TrainingDataset requirements class defines a XML encoding for the AI_TrainingDataset module, which is based on the UML model specified in the TrainingDML-AI Part 1: Conceptual Model Standard.

Requirements class	
/req/aitrainingdataset	
Dependency	XML
Dependency	/req/base/xmlbasetype
Dependency	/req/base/isometadatatype
Dependency	/req/aitrainingdata
Dependency	/req/aitask
Dependency	/req/ailabeling
Dependency	/req/aidataquality
Dependency	/req/aitdchangeset
Requirement 11	/req/aitrainingdataset/trainingdataset
Requirement 12	/req/aitrainingdataset/metricsinliterature

Requirements class	
Requirement 13	/req/aitrainingdataset/eotrainingdataset

The AI_TrainingDataset object is encoded as an XML element with properties shown in Table 2.

Requirement 11	/req/aitrainingdataset/trainingdataset Each AI_TrainingDataset object SHALL implement the Mandatory properties shown in Table 2.

Table 2. AI_TrainingDataset properties

XML Property	Definition	Data type and values	Obligation
type	Type of the training dataset.	"AI_AbstractTrainingDataset"	Mandatory
id	Identification of the AI training dataset.	CharacterString [1..1]	Mandatory
doi	Digital object identifier of the AI training dataset.	CharacterString [0..1]	Optional
scope	Description of the scope of the training dataset.	MD_Scope [0..1]	Optional
name	Name of the AI training dataset.	CharacterString [1..1]	Mandatory
description	Description of the AI training dataset.	CharacterString [1..1]	Mandatory
version	Version number of the AI training dataset.	CharacterString [0..1]	Optional
amountOfTrainingData	Total number of training samples in the AI training dataset.	Int [0..1]	Optional
createdTime	Time when the AI training dataset was created.	DateTime [0..1]	Optional
updatedTime	Time when the AI training dataset was updated.	DateTime [0..1]	Optional
license	License description of the AI training dataset.	CharacterString [1..1]	Mandatory
providers	People or organizations who provide the AI training dataset.	CharacterString [0..*]	Optional

XML Property	Definition	Data type and values	Obligation
keywords	Keywords of the AI training dataset.	CharacterString [0..*]	Optional
metricsInLIT	Results of performance metrics achieved by AI/ML algorithms in the peer-reviewed literature.	AI_MetricsInLiterature [0..*]	Optional
statisticsInfo	Statistical results for training samples in each class.	NamedValue [0..*]	Optional
dataSources	Citation for the data sources.	CI_Citation [0..*]	Optional
numberOfClasses	Total number of classes in the AI training dataset.	Int [0..1]	Optional
classificationSchema	Classification schema for classes used in the AI training dataset.	CharacterString [0..1]	Optional
classes	Classes used in the AI training dataset.	NamedValue [1..1]	Optional
tasks	Task description of the training dataset.	AI_Task [1..*]	Mandatory
labeling	Provenance information of how the training dataset is labeled.	AI_Labeling [0..*]	Optional
quality	Quality information of the training dataset.	DataQuality [0..*]	Optional
changesets	Changeset between two versions of the training dataset.	AI_TDChangeset [0..*]	Optional
data	Training data in the training dataset.	AI_TrainingData [1..*]	Mandatory

Example:

```
<type>AI_AbstractTrainingDataset</type>
<id>whu_rs19</id>
<name>WHU-RS19</name>
<description>Wuhan University-Remote Sensing 19 Categories (WHU-RS19) has 19 classes
of remote sensing images scenes obtained from Google Earth</description>
<amountOfTrainingData>1013</amountOfTrainingData>
```

```
<createdTime>2010-01-01</createdTime>
<providers>
   <provider>Wuhan University</provider>
</providers>
<keywords>
   <keyword>Remote Sensing</keyword>
   <keyword>Scene Classification</keyword>
</keywords>
<numberOfClasses>19</numberOfClasses>
<classes>
   <class>Airport</class>
   <class>Beach</class>
   <class>Bridge</class>
   <class>Commercial</class>
   <class>Desert</class>
   <class>Farmland</class>
   <class>footballField</class>
   <class>Forest</class>
   <class>Industrial</class>
   <class>Meadow</class>
   <class>Mountain</class>
   <class>Park</class>
   <class>Parking</class>
   <class>Pond</class>
   <class>Port</class>
   <class>railwayStation</class>
   <class>Residential</class>
   <class>River</class>
   <class>Viaduct</class>
</classes>
<tasks>
   <type>EOTask</type>
   <id>whu_rs19-task</id>
   <description>Structural high-resolution satellite image indexing</description>
   <taskType>Scene Classification</taskType>
</tasks>
<data>
   <type>EOTrainingData</type>
   <id>airport_01</id>
   <dataSources>
      <dataSource>googleEarth</dataSource>
   </dataSources>
   <dataURL>image/Airport/airport_01.jpg</dataURL>
   <labels>
      <type>SceneLabel</type>
      <class>Airport</class>
   </labels>
</data>
```

If the optional element AI_MetricsInLiterature is specified, this element is encoded as XML element

with properties as shown in Table 3.

Requirement 12	/req/aitrainingdataset/metricsinliterature Each AI_MetricsInLiterature value SHALL implement the Mandatory properties shown in Table 3.

Table 3. AI_MetricsInLiterature properties

XML Property	Definition	Data type and values	Obligation
doi	Digital object identifier of the peer-reviewed literature.	CharacterString [1..1]	Mandatory
algorithm	AI/ML algorithms used in the peer-reviewed literature.	CharacterString [0..1]	Optional
metrics	Metrics and results of AI/ML algorithms in the peer-reviewed literature.	NamedValue [1..*]	Mandatory

Example:

```
<doi>10.1109/TGRS.2019.2917161</doi>
<algorithm>FACNN</algorithm>
<metrics>
   <key>Overall Accuracy</key>
   <value>0.9881</value>
</metrics>
```

The AI_EOTrainingDataset object is encoded as an XML element with properties shown in Table 2 and Table 4.

Requirement 13	/req/aitrainingdataset/eotrainingdataset Each AI_EOTrainingDataset object SHALL implement the Mandatory properties both shown in Table 2 and Table 4.

Table 4. AI_EOTrainingDataset properties

XML Property	Definition	Data type and values	Obligation
type	Type of the training dataset.	"AI_EOTrainingDataset"	Mandatory
extent	Spatial extent of the EO training dataset.	EX_Extent [0..1]	Optional

XML Property	Definition	Data type and values	Obligation
bands	Bands description of the images used in the EO training dataset.	MD_Band [0..*]	Optional
imageSize	Size of the images used in the EO training dataset.	ChracterString [0..1]	Optional

Example:

```
<type>AI_EOTrainingDataset</type>
<id>whu_rs19</id>
<name>WHU-RS19</name>
<description>Wuhan University-Remote Sensing 19 Categories (WHU-RS19) has 19 classes
of remote sensing images scenes obtained from Google Earth</description>
<amountOfTrainingData>1013</amountOfTrainingData>
<createdTime>2010-01-01</createdTime>
<providers>Wuhan University</providers>
<keywords>Remote Sensing</keywords>
<keywords>Scene Classification</keywords>
<numberOfClasses>19</numberOfClasses>
<extent>-180</extent>
<extent>-90</extent>
<extent>180</extent>
<extent>90</extent>
<bands>red</bands>
<bands>green</bands>
<bands>blue</bands>
<imageSize>6000x7600</imageSize>
<classes>Airport</classes>
<classes>Beach</classes>
<classes>Bridge</classes>
<classes>Commercial</classes>
<classes>Desert</classes>
<classes>Farmland</classes>
<classes>footballField</classes>
<classes>Forest</classes>
<classes>Industrial</classes>
<classes>Meadow</classes>
<classes>Mountain</classes>
<classes>Park</classes>
<classes>Parking</classes>
<classes>Pond</classes>
<classes>Port</classes>
<classes>railwayStation</classes>
<classes>Residential</classes>
<classes>River</classes>
<classes>Viaduct</classes>
<tasks>
```

```
    <type>AI_EOTask</type>
    <id>whu_rs19-task</id>
    <description>Structural high-resolution satellite image indexing</description>
    <taskType>Scene Classification</taskType>
 </tasks>
 <data>
    <type>AI_EOTrainingData</type>
    <id>airport_01</id>
    <dataSources>googleEarth</dataSources>
    <dataURL>image/Airport/airport_01.jpg</dataURL>
    <labels>
       <type>AI_SceneLabel</type>
       <class>Airport</class>
    </labels>
 </data>
```

7.3. Requirements Class: AI_TrainingData

The AI_TrainingData requirements class defines an XML encoding for the AI_TrainingData module, which is based on the UML model specified in the TrainingDML-AI Part 1: Conceptual Model Standard.

Requirements class	
/req/aitrainingdata	
Dependency	XML
Dependency	/req/base/xmlbasetype
Dependency	/req/base/isometadatatype
Dependency	/req/ailabel
Dependency	/req/ailabeling
Dependency	/req/aidataquality
Requirement 14	/req/aitrainingdata/trainingdata
Requirement 15	/req/aitrainingdataset/trainingtypecode
Requirement 16	/req/aitrainingdata/eotrainingdata

The AI_TrainingData object is encoded as an XML element with properties shown in Table 5.

Requirement 14	/req/aitrainingdataset/trainingdata Each AI_TrainingData object SHALL implement the Mandatory properties shown in Table 5.

Table 5. AI_TrainingData properties

XML Property	Definition	Data type and values	Obligation
type	Type of the training data.	"AI_AbstractTrainingData"	Mandatory
id	Identification of the AI training data.	CharacterString [1..1]	Mandatory
datasetId	Identification of the training dataset that the training sample belongs to.	CharacterString [0..1]	Optional
trainingType	Training type of the individual AI training sample.	AI_TrainingTypeCode [0..1]	Optional
numberOfLabels	Total number of labels in the individual AI training sample.	Int [0..1]	Optional
dataSources	Citation of inputs to prepare a training sample.	CI_Citation [0..*]	Optional
labels	Labels in the training data.	AI_Label [1..*]	Mandatory
labeling	Provenance information of how the training data is labeled.	AI_Labeling [0..*]	Optional
quality	Quality information of the training data.	DataQuality [0..*]	Optional

Example:

```
<type>AI_AbstractTrainingData</type>
<id>airport_01</id>
<dataSources>googleEarth</dataSources>
<dataURL>image/Airport/airport_01.jpg</dataURL>
<labels>
   <type>AI_SceneLabel</type>
   <class>Airport</class>
</labels>
```

The AI_TrainingTypeCode is encoded as a text string whose value is one of “training”, “validation” or “test”.

Requirement 15	/req/aitrainingdataset/trainingtypecode Each AI_TrainingTypeCode value SHALL be a text string whose value is one of “training”, “validation” or “test”.

Examples:

```
a)  training
b)  validation
c)  test
```

The AI_EOTrainingData object is encoded as an XML element with properties both shown in Table and Table 6.

Requirement 16	/req/aitrainingdataset/eotrainingdata Each AI_EOTrainingData object SHALL implement the Mandatory properties as defined in Table 5 and Table 6.

Table 6. AI_EOTrainingData properties

XML Property	Definition	Data type and values	Obligation
type	Type of the EO training data.	"AI_EOTrainingData"	Mandatory
extent	Spatial extent of the individual EO training sample.	EX_Extent [0..1]	Optional
dataTime	Data time when the EO data was obtained.	DateTime [0..*]	Optional
dataURL	URL of the EO data.	URL [1..*]	Mandatory

Example:

```
<type>AI_EOTrainingData</type>
<id>airport_01</id>
<dataSources>googleEarth</dataSources>
<dataURL>image/Airport/airport_01.jpg</dataURL>
<labels>
   <type>AI_SceneLabel</type>
   <class>Airport</class>
</labels>
```

7.4. Requirements Class: AI_Task

The AI_Task requirements class defines a XML encoding for the AI_Task module, which is based on

the UML model specified in the TrainingDML-AI Part 1: Conceptual Model Standard.

Requirements class	
/req/aitask	
Dependency	XML
Dependency	/req/base/xmlbasetype
Dependency	/req/base/isometadatatype
Requirement 17	/req/aitask/task
Requirement 18	/req/aitask/eotask

The AI_Task object is encoded as an XML element with properties as shown in Table 7.

Requirement 17	/req/aitask/task Each AI_Task object SHALL implement the Mandatory properties shown in Table 7.

Table 7. AI_Task properties

XML Property	**Definition**	**Data type and values**	**Obligation**
type	Type of the task object.	"AI_AbstractTask"	Mandatory
id	Identification of the task.	CharacterString [1..1]	Mandatory
datasetId	Identification of the training dataset the training sample belongs to.	CharacterString [0..1]	Optional
description	Description of the AI task.	CharacterString [0..1]	Optional

Example:

```
<type>AI_AbstractTask</type>
<id>image-indexing-task</id>
<description>Structural high-resolution satellite image indexing</description>
```

The AI_EOTask object is encoded as an XML element with properties both shown in Table 7 and Table 8.

Requirement 18	/req/aitask/eotask Each AI_EOTask object SHALL implement the Mandatory properties shown in Table 7 and Table 8.

Table 8. AI_EOTask properties

XML Property	Definition	Data type and values	Obligation
type	Type of the task object.	"AI_EOTask"	Mandatory
taskType	Type of the EO task.	CharacterString [1..1]	Mandatory

Example:

```
<type>AI_EOTask</type>
<id>image-indexing-task</id>
<description>Structural high-resolution satellite image indexing</description>
<taskType>Scene Classification</taskType>
```

7.5. Requirements Class: AI_Label

The AI_Label requirements class defines an XML encoding for the AI_Label module, which is based on the UML model specified in the TrainingDML-AI Part 1: Conceptual Model Standard.

Requirements class	
/req/ailabel	
Dependency	XML
Dependency	/req/base/xmlbasetype
Dependency	/req/base/isometadatatype
Dependency	/req/base/geospatialtype
Requirement 19	/req/ailabel/label
Requirement 20	/req/ailabel/scenelabel
Requirement 21	/req/ailabel/objectlabel
Requirement 22	/req/ailabel/pixellabel
Requirement 23	/req/ailabel/imageformatcode

The AI_Label object is encoded as an XML element with properties as shown in Table 9.

Requirement 19	/req/ailabel/label Each AI_Label object SHALL implement the Mandatory properties shown in Table 9.

Table 9. AI_Label properties

XML Property	Definition	Data type and values	Obligation
type	Type of the label object.	"AI_AbstractLabel"	Mandatory

XML Property	Definition	Data type and values	Obligation
isNegative	Whether the training sample related to the label is a positive or negative sample.	Bool [0..1] Default: false	Optional
confidence	Confidence score of the labeler.	Float [0..1] Default: 1.0 Range: [0, 1]	Optional

Example:

```
<type>AI_AbstractLabel</type>
<isNegative>false</isNegative>
```

The AI_SceneLabel object is encoded as an XML element with properties as shown in Table 10.

Requirement 20	/req/ailabel/scenelabel Each AI_SceneLabel object SHALL implement the Mandatory properties shown in Table 10.

Table 10. AI_SceneLabel properties

XML Property	Definition	Data type and values	Obligation
type	Type of the label object at the scene level.	"AI_SceneLabel"	Mandatory
class	Class that records the semantic of the scene of the training sample.	CharacterString [1..1]	Mandatory

Example:

```
<type>AI_SceneLabel</type>
<class>Airport</class>
```

The AI_ObjectLabel object is encoded as an XML element with properties shown in Table 11.

Requirement 21	/req/ailabel/objectlabel Each AI_ObjectLabel object SHALL implement the Mandatory properties shown in Table 11.

Table 11. AI_ObjectLabel properties

XML Property	Definition	Data type and values	Obligation
type	Type of the label object at the object level.	"AI_ObjectLabel"	Mandatory
object	Feature that represents the position and attributes of the object.	Feature [0..1]	Mandatory
bboxType	Type of the bbox.	CharacterString [0..1]	Optional
class	Class that records the semantic of the object type.	CharacterString [1..1]	Mandatory
dateTime	Created time of the object label.	DateTime [0..1]	Optional

Example:

```
<type>AI_ObjectLabel</type>
<class>Pedestrian</class>
<object>
   <gml:metaDataProperty>
      <gml:GenericMetaData>
         <truncated>0.0</truncated>
         <occluded>0</occluded>
         <alpha>-0.2</alpha>
      </gml:GenericMetaData>
   </gml:metaDataProperty>
   <gml:location>
      <gml:GeometricComplex>
         <gml:element>
            <gml:Polygon>
               <gml:exterior>
                  <gml:LinearRing>
                     <gml:posList>
                        51.556272 -0.2803943 51.5562758 -0.2787397 51.5556539
-0.278736 51.5556501 -0.2803906 51.556272 -0.2803943
                     </gml:posList>
                  </gml:LinearRing>
               </gml:exterior>
            </gml:Polygon>
         </gml:element>
      </gml:GeometricComplex>
   </gml:location>
</object>
<bboxType>Horizontal BBox</bboxType>
```

The AI_PixelLabel object is encoded as an XML element with properties as shown in Table 12.

Requirement 22	/req/ailabel/pixellabel Each AI_PixelLabel object SHALL implement the Mandatory properties shown in Table 12.

Table 12. AI_PixelLabel properties

XML Property	Definition	Data type and values	Obligation
type	Type of the label object at the pixel level.	"AI_PixelLabel"	Mandatory
imageURL	URL of the images representing the label information.	URL [1..*]	Mandatory
imageFormat	Image data format.	AI_ImageFormatCode [1..*]	Mandatory

Example:

```
<type>AI_PixelLabel</type>
<imageURL>/label_5classes/GF2_PMS1__L1A0000647767-MSS1_label.tif</imageURL>
<imageFormat>image/tiff; application=geotiff</imageFormat>
```

The AI_ImageFormatCode is encoded as a text string whose value is defined in Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types RFC 2046.

Requirement 23	/req/ailabel/imageformatcode Each AI_ImageFormatCode value SHALL be encoded as a text string defined in Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types RFC 2046.

Examples: a) image/tiff; application=geotiff b) application/x-netcdf c) image/png d) image/jp2

7.6. Requirements Class: AI_Labeling

The AI_Labeling requirements class defines a XML encoding for the AI_Labeling module, which is based on the UML model specified in the TrainingDML-AI Part 1: Conceptual Model Standard.

Requirements class	
/req/ailabeling	
Dependency	XML
Dependency	/req/base/xmlbasetype
Dependency	/req/base/isometadatatype
Requirement 24	/req/ailabeling/labeling

Requirements class	
Requirement 25	/req/ailabeling/labeler
Requirement 26	/req/ailabeling/labelingprocedure
Requirement 27	/req/ailabeling/labelingmethodcode

The AI_Labeling object is encoded as an XML element with properties shown in Table 13.

Requirement 24	/req/ailabeling/labeling Each AI_Labeling object SHALL implement the Mandatory properties shown in Table 13.

Table 13. AI_ Labeling properties

XML Property	Definition	Data type and values	Obligation
type	Type of the labeling object.	"AI_Labeling"	Mandatory
id	Identifier of the labeling.	CharacterString [1..1]	Mandatory
scope	Description of the scope of the labeling.	MD_Scope [1..1]	Mandatory
labelers	Labelers of the labeling activity.	AI_Labeler [0..*]	Optional
procedure	Procedure used in the labeling activity.	AI_LabelingProcedure [0..1]	Optional

Example:

```
<type>AI_Labeling</type>
<id>0</id>
<scope>
   <level>dataset</level>
   <levelDescription>
      <dataset>whu_rs19</dataset>
   </levelDescription>
</scope>
<labelers>...</labelers>
<procedure>...</procedure>
```

The AI_Labeler object is encoded as an XML element with properties as shown in Table 14.

Requirement 25	/req/ailabeling/labeler Each AI_Labeler object SHALL implement the Mandatory properties shown in Table 14.

Table 14. AI_ Labeler properties

XML Property	Definition	Data type and values	Obligation
type	Type of the labeler object.	"AI_Labeler"	Mandatory
id	Identifier of the labeler.	CharacterString [1..1]	Mandatory
name	Name of the labeler.	CharacterString [1..1]	Mandatory

Example:

```
<type>AI_Labeler</type>
<id>0</id>
<name>Tom</name>
```

The AI_LabelingProcedure object is encoded as an XML element with properties as shown in Table 15.

Requirement 26	/req/ailabeling/labelingprocedure Each AI_LabelingProcedure object SHALL implement the Mandatory properties shown in Table 15.

Table 15. AI_ LabelingProcedure properties

XML Property	Definition	Data type and values	Obligation
type	Type of the labeling procedure object.	"AI_LabelingProcedure"	Mandatory
id	Identifier of the labeling procedure.	CharacterString [1..1]	Mandatory
methods	Methods used in the labeling procedure.	AI_LabelingMethodCode [1..*]	Mandatory
tools	Tools or software used in the labeling procedure.	CharacterString [0..*]	Optional

Example:

```
<type>AI_LabelingProcedure</type>
<id>0</id>
<methods>
   <method>manual</method>
</methods>
<tools>
   <tool>ArcGIS</tool>
</tools>
```

The AI_LabelingMethodCode is encoded as a text string whose value is one of “manual”, “semi-automatic” or “automatic”.

Requirement 27	/req/ailabeling/labelingmethodcode Each AI_LabelingMethodCode value SHALL be a text string whose value is one of “manual”, “semi-automatic” or “automatic”.

Examples: a) manual b) semi-automatic c) automatic

7.7. Requirements Class: AI_DataQuality

The AI_ClassBalanceDegree object is encoded as an XML element with properties as shown in Table 16.

Requirement 28	/req/aidataquality/classbalancedegree Each AI_ClassBalanceDegree object SHALL implement the Mandatory properties as shown in Table 16.

Table 16. AI_ ClassBalanceDegree properties

XML Property	Definition	Data type and values	Obligation
type	Type of the class balance degree object.	"AI_ClassBalanceDegree"	Mandatory
measure	Reference to measure used.	MeasureReference [1..1]	Mandatory
evaluationMethod	Evaluation information.	EvaluationMethod [1..1]	Mandatory
result	Value obtained from applying a data quality measure.	QualityResult [1..*]	Mandatory

Example:

```
<type>AI_ClassBalanceDegree</type>
<measure>
  <measureDescription>
    Balance degree of label classes
  </measureDescription>
</measure>
<evaluationMethod>
  <evaluationMethodDescription>
    Counting the number of training samples belonging to each class and calculating
the balance degree
  </evaluationMethodDescription>
</evaluationMethod>
```

```
<result>
  <quantitativeResult>
      <value>
        93.5
      </value>
      <valueUnit>
        %
      </valueUnit>
  </quantitativeResult>
</result>
```

7.8. Requirements Class: AI_TDChangeset

The AI_TDChangeset requirements class defines an XML encoding for the AI_TDChangeset module, which is based on the UML model specified in the TrainingDML-AI Part 1: Conceptual Model Standard.

Requirements class	
/req/aitdchangeset	
Dependency	XML
Dependency	/req/base/xmlbasetype
Dependency	/req/base/isometadatatype
Dependency	/req/tdtrainingdata
Requirement 29	/req/aitdchangeset/tdchangeset

The AI_TDChangeset object is encoded as an XML element with properties shown in Table 17.

Requirement 29	/req/aitdchangeset/tdchangeset Each AI_TDChangeset object SHALL implement the Mandatory properties as shown in Table 17.

Table 17. AI_ TDChangeset properties

XML Property	**Definition**	**Data type and values**	**Obligation**
type	Type of the TD changeset object.	"AI_TDChangeset"	Mandatory
id	Identifier of the changeset.	CharacterString [1..1]	Mandatory
datasetId	Identifier of the training dataset the changeset belongs to.	CharacterString [0..1]	Optional

XML Property	Definition	Data type and values	Obligation
version	Version of the training dataset that the changeset belongs to.	CharacterString [0..1]	Optional
changeCount	Total number of changed training samples.	Int [1..1]	Mandatory
createdTime	The time that the changeset was created.	DateTime [0..1]	Optional
add	Added training samples.	AI_TrainingData [0..*]	Optional
modify	Modified training samples.	AI_TrainingData [0..*]	Optional
delete	Deleted training samples.	AI_TrainingData [0..*]	Optional

Example:

```
<type>AI_TDChangeset</type>
<id>changeset-dota_v1.5</id>
<datasetId>dota_v1.5</datasetId>
<createdTime>2019-01-01</createdTime>
<changeCount>9</changeCount>
<modify>
   <type>EOTrainingData</type>
   <id>P1228</id>
   <dataSources>GF</dataSources>
   <dataURL>train/images/P1228.png</dataURL>
   <numberOfLabels>50</numberOfLabels>
   <trainingType>training</trainingType>
   <labels>...</labels>
</modify>
```

Annex A: Abstract Test Suite (Normative)

A.1. Introduction

Conformance is tested using the XML Schema document which formalize the requirements described above.

A.2. Conformance Class: Base

The Base conformance class tests that occurrences of the basic types are encoded according to the requirements.

Conformance Class	/conf/base	
Requirements	/req/base	
Dependency	An XML Schema Validator	
Test	/conf/base/xml	
	Requirement	/req/base/xmlbasetype/xml
	Test purpose	Verify that the document is well-formed XML.
	Test method	Load the document in an XML validator. Pass if no errors reported. Fail otherwise.
	Test type	Capability
Test	/conf/base/type	
	Requirement	/req/base/xmlbasetype/datatime, /req/base/xmlbasetype/namedvalue, /req/base/xmlbasetype/url, /req/base/isometadatatype, /req/base/isoqualitytype, /req/base/geospatialtype
	Test purpose	Verify that the related values and objects are encoded using the specified property names and structures.
	Test method	Validate the XML instance document using the appropriate object definition from the TrainingDML-AI.xsd XML Schema. Pass if no errors reported. Fail otherwise.
	Test type	Capability

A.3. Conformance Class: AI_TrainingDataset

The AI_TrainingDataset conformance class tests that the training dataset object is encoded

according to the requirements.

Conformance Class	/conf/aitrainingdataset	
Requirements	/req/aitrainingdataset	
Dependency	An XML Schema Validator	
Test	Test purpose	Verify that the training dataset object is encoded using the specified property names and structures.
	Test method	Validate the XML instance document using the appropriate object definition from the TrainingDML-AI.xsd XML Schema. Pass if no errors reported. Fail otherwise.
	Test type	Capability

A.4. Conformance Class: AI_TrainingData

The AI_TrainingData conformance class tests that the training data objects are encoded according to the requirements.

Conformance Class	/conf/aitrainingdata	
Requirements	/req/aitrainingdata	
Dependency	An XML Schema Validator	
Test	Test purpose	Verify that the training data objects are encoded using the specified property names and structures.
	Test method	Validate the XML instance document using the appropriate object definition from the TrainingDML-AI.xsd XML Schema. Pass if no errors reported. Fail otherwise.
	Test type	Capability

A.5. Conformance Class: AI_Task

The AI_Task conformance class tests that the task objects are encoded according to the requirements.

Conformance Class	/conf/aitask
Requirements	/req/aitask
Dependency	An XML Schema Validator

Test	Test purpose	Verify that the task objects are encoded using the specified property names and structures.
	Test method	Validate the XML instance document using the appropriate object definition from the TrainingDML-AI.xsd XML Schema. Pass if no errors reported. Fail otherwise.
	Test type	Capability

A.6. Conformance Class: AI_Label

The AI_Label conformance class tests that the label objects are encoded according to the requirements.

Conformance Class	/conf/ailabel	
Requirements	/req/ailabel	
Dependency	An XML Schema Validator	
Test	Test purpose	Verify that the label objects are encoded using the specified property names and structures.
	Test method	Validate the XML instance document using the appropriate object definition from the TrainingDML-AI.xsd XML Schema. Pass if no errors reported. Fail otherwise.
	Test type	Capability

A.7. Conformance Class: AI_Labeling

The AI_Labeling conformance class tests that the labeling objects are encoded according to the requirements.

Conformance Class	/conf/ailabeling	
Requirements	/req/ailabeling	
Dependency	An XML Schema Validator	
Test	Test purpose	Verify that the labeling objects are encoded using the specified property names and structures.
	Test method	Validate the XML instance document using the appropriate object definition from the TrainingDML-AI.xsd XML Schema. Pass if no errors reported. Fail otherwise.
	Test type	Capability

A.8. Conformance Class: AI_TDChangeset

The AI_TDChangeset conformance class tests that the TD changeset objects are encoded according to the requirements.

Conformance Class	/conf/aitdchangeset	
Requirements	/req/aitdchangeset	
Dependency	An XML Schema Validator	
Test	Test purpose	Verify that the TD changeset objects are encoded using the specified property names and structures.
	Test method	Validate the XML instance document using the appropriate object definition from the TrainingDML-AI.xsd XML Schema. Pass if no errors reported. Fail otherwise.
	Test type	Capability

Annex B: Example (Informative)

B.1. TrainingDataset Encoding Examples

B.1.1. WHU-RS19 Dataset

The WHU-RS19 dataset is widely used in scene classification of remote sensing images. This dataset is collected from Google Earth and has 19 classes including airport, beach, bridge, commercial, desert, farmland, football field, forest, industrial, meadow, mountain, park, parking, pond, port, railway station, residential, river, and viaduct. Each class contains around 50 images, with an image size of 600×600 and a resolution of 0.5 m.

An example of XML encoding of the WHU-RS19 dataset following the TrainingDML-AI UML model can be found in https://github.com/opengeospatial/TrainingDML-AI_SWG/tree/main/use-cases/examples/1.0/WHU-RS19.xml.

B.1.2. DOTA-v1.5 Dataset

The DOTA-v1.5 dataset is a large-scale dataset for object detection in aerial images. The sources for content in the dataset include Google Earth, Gaofen-2, and Jilin-1 imagery provided by China Resources Satellite Data Center. The 16 classes in DOTA-v1.5 are plane, ship, storage tank, baseball diamond, tennis court, basketball court, ground track field, harbor, bridge, large vehicle, small vehicle, helicopter, roundabout, soccer ball field, swimming pool, and container crane. Compared with other aerial image object detection datasets, the dataset has the largest number of classes. The images in the dataset have various image sizes (from 800×800 to 2000×2000) and resolutions (Google Earth/0.1 m-1 m, Gaofen-2/1 m, Jilin-1/0.72 m).

An example of XML encoding of the DOTA-v1.5 dataset following the TrainingDML-AI UML model can be found in https://github.com/opengeospatial/TrainingDML-AI_SWG/tree/main/use-cases/examples/1.0/DOTA-v1.5.xml.

B.1.3. KITTI 2D Object Detection Dataset

The KITTI 2D object detection dataset is a novel open-access dataset and benchmark for road area and ego-lane detection. KITTI 2D consists of 7481 annotated training images of high variability from the KITTI autonomous driving platform by two PointGrey Flea2 color cameras, capturing a broad spectrum of urban street views and road scenes. The eight (8) classes in the KITTI 2D object detection dataset are car, van, truck, pedestrian, person_sitting, cyclist, tram, and misc. Compared with other street view object detection datasets, this dataset compresses diverse scenarios and captures real-world traffic situations, ranging from freeways over rural areas to inner-city scenes with many static and dynamic objects.

An example of XML encoding of the KITTI 2D object detection dataset following the TrainingDML-AI UML model can be found in https://github.com/opengeospatial/TrainingDML-AI_SWG/tree/main/use-cases/examples/1.0/KITTI.xml.

B.1.4. GID Dataset

The GID dataset is one of state-of-art land cover classification datasets. This dataset has a large spatial coverage covering many provinces in China with a relatively high spatial resolution (2 m). GID has two sets. One is the GID-5C. It has 150 images (image size 7200×6800) that are classified into 5 land cover classes. The other set is GID-15C. The images from GID-5C are sliced into 30,000 patches in GID-15C, which have three types of patch sizes (56×56, 112×112, 224×224) and are classified into 15 land cover classes.

An example of XML encoding of the GID-5C dataset following the TrainingDML-AI UML model can be found in https://github.com/opengeospatial/TrainingDML-AI_SWG/tree/main/use-cases/examples/1.0/GID-5C.xml.

B.1.5. Toronto3D Dataset

The Toronto3D dataset is a large urban outdoor point cloud dataset for segmentation collected by the Mobile Laser Scanning System. The dataset covers about 1 km of scene streets in Toronto, including four areas named L001, L002, L003, and L004, with a total of 78.3 million points. Each point in this dataset has 10 attributes representing the 3D position, RGB color, intensity, GPS time, scan angle rank, and category, respectively. This dataset has eight categories, including road, road mark, natural, building, utility line, pole, car, and fence.

An example of XML encoding of the Toronto3D dataset following the TrainingDML-AI UML model can be found in https://github.com/opengeospatial/TrainingDML-AI_SWG/tree/main/use-cases/examples/1.0/Toronto_3D.xml.

B.1.6. WHU-Building Dataset

The WHU-Building dataset is a change detection dataset collected from the Land Information New Zealand Data Service. The dataset is composed of images (with the resolution 0.2 m) in 2012 and 2016, covering 20.5 km2. It includes 12,796 and 16,077 buildings respectively in 2012 and 2016.

An example of XML encoding of the WHU-Building dataset following the TrainingDML-AI UML model can be found in https://github.com/opengeospatial/TrainingDML-AI_SWG/tree/main/use-cases/examples/1.0/WHU-building.xml.

B.1.7. California Change Detection Dataset

The California Change Detection Dataset is composed of two images and a label image. The first image is a Landsat 8 acquisition covering Sacramento County, Yuba County and Sutter County, California, on 5 January 2017. It has nine channels covering the spectrum from deep blue to short-wave infrared, plus two long-wave infrared channels. The second image was acquired on 18 February 2017 by Sentinel-1A over the same area after the occurrence of a flood. The image is recorded in polarizations VV and VH and augmented with the ratio between the two intensities as a third channel. All these channels are log-transformed.

An example of XML encoding of the California change detection dataset following the TrainingDML-AI UML model can be found in https://github.com/opengeospatial/TrainingDML-AI_SWG/tree/main/use-cases/examples/1.0/UiT_HCD_California_2017.xml.

B.1.8. WHU MVS Dataset

The WHU MVS dataset is a synthetic aerial dataset created for large-scale and high-resolution Earth surface reconstruction. The basic training sample of the dataset is a multi-view unit consisting of five aerial images, and their corresponding depth maps are taken as ground truth. There are a total of 5680 pairs of five-view aerial images in the dataset. All the images are simulated from a 3D surface model, which is produced by Smart3D software using Unmanned Aerial Vehicle (UAV) images and refined by manual editing.

An example of XML encoding of the WHU MVS dataset following the TrainingDML-AI UML model can be found in https://github.com/opengeospatial/TrainingDML-AI_SWG/tree/main/use-cases/examples/1.0/WHU_MVS.xml.

B.1.9. iSAID Dataset

The iSAID dataset is the first benchmark dataset for instance segmentation in aerial images. This large-scale and densely annotated dataset contains 655,451 object instances for 15 categories across 2,806 high-resolution images. The images of iSAID is the same as the DOTA-v1.0 dataset, which are manily collected from the Google Earth, some are taken by satellite JL-1, the others are taken by satellite GF-2 of the China Centre for Resources Satellite Data and Application. The object categories in iSAID include: plane, ship, storage tank, baseball diamond, tennis court, basketball court, ground track field, harbor, bridge, large vehicle, small vehicle, helicopter, roundabout, soccer ball field and swimming pool.

An example of XML encoding of the iSAID dataset following the TrainingDML-AI UML model can be found in https://github.com/opengeospatial/TrainingDML-AI_SWG/blob/main/use-cases/examples/1.0/iSAID.xml.

B.2. DataQuality Encoding Example

B.2.1. WHU-RS19 Data Quality

An encoded data quality example of the WHU-RS19 datasets following the TrainingDML-AI UML model can be found in https://github.com/opengeospatial/TrainingDML-AI_SWG/tree/main/use-cases/examples/1.0/WHU-RS19-quality.xml.

B.3. TDChangeset Encoding Example

B.3.1. DOTA-v1.5 Changeset

DOTA-v1.5 uses the same images as DOTA-v1.0, but the extremely small instances (less than 10 pixels) are also annotated. Moreover, a new category "container crane" is added. It contains 403,318 instances in total. The number of images and dataset splits are the same as DOTA-v1.0. This version was released for the DOAI Challenge 2019 on Object Detection in Aerial Images in conjunction with IEEE CVPR 2019.

An encoded changeset example between the DOTA-v1.0 and DOTA-v1.5 datasets following the TrainingDML-AI UML model can be found in https://github.com/opengeospatial/TrainingDML-

AI_SWG/tree/main/use-cases/examples/1.0/DOTA-v1.5-changeset.xml.

B.4. Non-EO Imagery TrainingDataset Encoding Examples

B.4.1. ERA5 Dataset

The ERA5 dataset is derived from in-situ observational data (Copernicus product), and we limit its usage scenario to the autoregression problem of time series data. Therefore, its label is the data itself. Similar to unsupervised learning, the autoregression task for time series data does not require additional labeled data. For this dataset, inheritance classes for AI_AbstractLabel are not defined, although this class is required in the existing standard (please note that these test cases are for future versions of the standard). In addition, additional attributes to support the complete representation of dataset information were added.

An example of XML encoding of the ERA5 dataset following the TrainingDML-AI UML model can be found in https://github.com/opengeospatial/TrainingDML-AI_SWG/blob/main/use-cases/examples/1.0/ERA5_hourly_data.xml.

B.4.2. SCIERC Dataset

The SCIERC dataset is derived from textual data, and its labels are the classification of the text. This dataset is a text classification problem, with the goal of information extraction and entity recognition. For this textual dataset, the Abstract class is inherited and AI_TextTrainingDataset, AI_TextTrainingData, AI_TextTask, and AI_EntityLabel respectively are defined. In addition, additional attributes to support the complete representation of dataset information were added.

An example of XML encoding of the SCIERC dataset following the TrainingDML-AI UML model can be found in https://github.com/opengeospatial/TrainingDML-AI_SWG/blob/main/use-cases/examples/1.0/SCIRec.xml.

B.4.3. nuScenes Dataset

The nuScenes dataset is a public large-scale dataset for autonomous driving developed by the team at Motional (formerly nuTonomy). The full dataset includes approximately 1.4M camera images, 390k LIDAR sweeps, 1.4M RADAR sweeps and 1.4M object bounding boxes in 40k keyframes. Although the training data may come from different domains, the 3D annotation boxes captured by numerous sensors in the same keyframe are targeted at the same object and are unique. Based on this, a 3D annotation box is used to organize each 3D object using AI_ObjectLabel. Since each training data and each 3D object require many additional attributes to be fully described, many additional attributes to provide a detailed description of the training dataset, training data, labels, etc. were added.

An example of XML encoding of the nuScenes dataset following the TrainingDML-AI UML model can be found in https://github.com/opengeospatial/TrainingDML-AI_SWG/blob/main/use-cases/examples/1.0/nuScenes.xml.

Annex C: Revision History (Informative)

Date	Release	Author	Paragraph modified	Description

Annex D: Bibliography

[1] Yue, P., ed., 2023. OGC Training Data Markup Language for Artificial Intelligence (TrainingDML-AI) Part1: Conceptual Model Standard, OGC 23-008r3. Wayland, MA: Open Geospatial Consortium Inc. https://docs.ogc.org/is/23-008r3/23-008r3.html

[2] Freed, N., 1996. RFC 2046. Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. https://www.ietf.org/rfc/rfc2046.txt

[3] Portele, C., 2007. OpenGIS Geography Markup Language (GML) Encoding Standard, OGC 07-036. Wayland, MA: Open Geospatial Consortium Inc. https://portal.ogc.org/files/?artifact_id=20509

[4] World Wide Web Consortium. Extensible Markup Language (XML). https://www.w3.org/XML/

[5] World Wide Web Consortium. XML Schema. https://www.w3.org/XML/Schema

[6] ISO, 2019. ISO 19107: 2019. Geographic information — Spatial schema. https://www.iso.org/standard/66175.html

[7] ISO, 2022. ISO 19157-1: 2022. Geographic information — Data quality. https://www.iso.org/standard/78900.html

[8] ISO, 2014. 19115-1:2014, Geographic information — Metadata — Part 1: Fundamentals. https://www.iso.org/standard/53798.html

[9] Landry, T., ed., 2018. OGC Testbed-14: Machine Learning Engineering Report, OGC 18-038r2. Wayland, MA: Open Geospatial Consortium Inc. https://docs.ogc.org/per/18-038r2.html

[10] Meek, S., ed., 2019. OGC Testbed-15: Machine Learning Engineering Report, OGC 19-027r2. Wayland, MA: Open Geospatial Consortium Inc. https://docs.ogc.org/per/19-027r2.html

[11] Schumann, G., ed., 2020. OGC Testbed-16: Machine Learning Training Data Engineering Report, OGC 20-018. Wayland, MA: Open Geospatial Consortium Inc. https://docs.ogc.org/per/20-015r2.html

[12] Yue, P., Shangguan, B., Hu, L., Jiang, L., Zhang, C., Cao, Z., Pan, Y., 2022. Towards a training data model for artificial intelligence in earth observation. International Journal of Geographical Information Science, 1-25. https://doi.org/10.1080/13658816.2022.2087223