**3GPP TSG-SA5 Meeting #143eS5-223394**

**09 - 17 May 2022, E-meeting**

**Source: Nokia**

**Title: pCR 28.105 Add requirements for handling erroneous data & decisions**

**Document for: Approval**

**Agenda Item: 6.6.5**

# 1 Decision/action requested

***The group is asked to discuss and agree on the proposal.***

# 2 References

[1] 3GPP TS 28.105-110 “Management and orchestration; AI/ML management”.

# 3 Rationale

The AI/ML training needs to support the capabilities for handling errors in input data or in the learned decisions. This pCR presents the corresponding requirements.

# 4 Detailed proposal

|  |
| --- |
| **Start of modifications** |

## 6.N Handling errors in data and ML decisions

### 6.N.1 Description

Traditionally, the machine-learning-enabled Functions (e.g., AIML Entity 1 and AIML Entity 2) are trained on good quality data, i.e., data that was collected when the network was working correctly, to represent the expected context in which the AIML Entity is meant to operate. Good quality data is void of errors, such as:

* Imprecise measurements, with added noise (such as RSRP, SINR, or QoE estimations).
* Missing values or entire records, e.g. because of communication link failures.
* Records which are communicated with a significant delay (in case of online measurements).

Without errors, an AIML Entity can depend on a few precise inputs, and don’t need to exploit the redundancy present in the training data. However, during inference, the AIML Entity is very likely to come across these inconsistencies. When this happens, the AIML Entity shows high error in the inference outputs, even if redundant and uncorrupted data is available from other sources.

AIML Entity1

AIML Entity2



Network Resources

ML Consumer

p

KPIs

**Error**

p

Fig 3. The propagation of erroneous information

As such the system needs to account for errors and inconsistencies in the input data and how the consumers of ML decisions should deal with decisions that are made based on such erroneous and inconsistent data. The system should: 1) enable functions to undertake the training in a way that prepares the AIML Entity s to deal with the errors, i.e. to identify the errors in the data during training; and 2) enable the ML consumers to account for the possibility of erroneous input data into the ML decision makers.

### 6.N.2 Requirements

|  |  |  |
| --- | --- | --- |
| **Requirement label** | **Description** | **Related use case(s)** |
| **ML\_Error\_Req\_1** | The 3GPP management system shall enable an authorized consumer of data services (e.g., an ML-enabled function) to request from a producer of data services a **Value Quality Score of the data, which is t**he numerical value that represents the dependability/quality of a given observation and measurement type. | Handling errors in data and ML decisions |
| **ML\_Error\_Req\_2** | The 3GPP management system shall enable an authorized consumer of ML decisions (e.g. a controller) to request **ML decision confidence score which is t**he numerical value that represents the dependability/quality of a given decision generated by the ML-based function. | Handling errors in data and ML decisions |
| **ML\_Error\_Req\_3** | The 3GPP management system shall enable a producer of data services (e.g., a gNB) to provide to an authorized consumer (e.g., an ML-enabled function) a **Value Quality Score of the data, which is t**he numerical value that represents the dependability/quality of a given observation and measurement type. | Handling errors in data and ML decisions |
| **ML\_Error\_Req\_4** | The 3GPP management system shall enable a producer of ML decisions (e.g. an ML-enabled function) to provide to an authorized consumer of ML decisions (e.g. a controller) an **ML decision confidence score which is t**he numerical value that represents the dependability/quality of a given decision generated by the ML-based function. | Handling errors in data and ML decisions |

|  |
| --- |
| **Next change** |

## 7.3 Class definitions

### 7.3.1 AIMLTrainingFunction

#### 7.3.1.1 Definition

The IOC AIMLTrainingFunction represents the entity that undertakes AI/ML training and is also the container of the AIMLTrainingRequest IOC(s).

The entity represented by AIMLTrainingFunction MOI supports training of one or more AIMLEntity(s).

The AIMLTrainingFunction contains one or more AIMLTrainingRequest MOIs.

#### 7.3.1.2 Attributes

The AIMLTrainingFunction IOC includes attributes inherited fromTOP IOC (defined in TS 28.622) and the following attributes:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
| aIMLEntityList | M | T | F | F | F |

#### 7.3.1.3 Attribute constraints

None.

#### 7.3.1.4 Notifications

The common notifications defined in clause 7.6 are valid for this IOC, without exceptions or additions.

### 7.3.2 AIMLTrainingRequest

#### 7.3.2.1 Definition

The IOC AIMLTrainingRequest represents the AI/ML model training request that is created by the MnS consumer.

The AIMLTrainingRequest MOI is contained under one AIMLTrainingFunction MOI. Each AIMLTrainingRequest is associated to at least one AIMLEntity.

The AIMLTrainingRequest may have a source to identify where it is coming from and which may be used to prioritize the resources for different sources. The sources may be for example the network functions, operator roles, or other functional differentiations.

Each AIMLTrainingRequest may indicate the expectedRunTimeContext that describes the specific conditions for which the AIMLEntity (either AIML Model or AIML-enabled function) should be trained for.

In case the request is accepted, the MnS producer decides when to start the AI/ML training. Once the MnS producer decides to start the training based on the request, the MnS producer instantiates one or more AI/MLTrainingProcess MOI(s) that are responsible to:

- collects (more) data for training, if the training data are not available or the data are available but not sufficient for the training,

- prepares and selects the training data, with consideration of the consumer provided candidate training data if any. The MnS producer may examine the consumer provided candidate training data and select none, some or all of them for training. In addition, the MnS producer may select some other training data that are available,

- trains the AIMLEntity using the selected and prepared training data.

The AIMLTrainingRequest may have a requestStatus field to represent the status of the specific AIMLTrainingRequest:

- The attribute values are "NOT\_STARTED", "TRAINING\_IN\_PROGRESS", "SUSPENDED", "FINISHED", and "CANCELLED".

- When value turns to "TrainingInProcess", the MnS producer instantiates one or more AIMLTrainingProcess MOI(s) representing the training process(es) being performed per the request and notifies the MnS consumer(s) who subscribed the notification.

When all of the training process associated to this request are completed, the value turns to "FINISHED.

#### 7.3.2.2 Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
| aIMLEntityId | M | T | T | F | T |
| candidateTraingDataSource | O | T | T | F | T |
| traingDataQualityScore | O | T | T | F | T |
| trainingRequestSource | M | T | T | F | T |
| requestStatus | M | T | T | F | T |
| expectedRuntimeContext |  |  |  |  |  |
| peformanceRequirements | M | T | T | F | T |
| cancelRequest | O | T | T | F | T |
| suspendRequest | O | T | T | F | T |
| **Attribute related to role** |  |  |  |  |  |
|  |  |  |  |  |  |

#### 7.3.2.3 Attribute constraints

|  |  |
| --- | --- |
| Name | Definition |
|  |  |
|  |  |

#### 7.3.2.4 Notifications

The common notifications defined in clause 7.6 are valid for this IOC, without exceptions or additions.

### 7.3.3 AIMLTrainingReporting

#### 7.3.3.1 Definition

The IOC AIMLTrainingReporting represents the container of the AIMLTrainingReport IOC(s).

#### 7.3.3.2 Attributes

No additional attributes other than the ones inherited from the parent class.

#### 7.3.3.3 Attribute constraints

None.

#### 7.3.3.4 Notifications

The common notifications defined in clause 7.6 are valid for this IOC, without exceptions or additions.

### 7.3.4 AIMLTrainingReport

#### 7.3.4.1 Definition

The IOC AIMLTrainingReport represents the AI/ML model training report that is provided by the MnS producer.

The AIMLTrainingReport MOI is contained under one AIMLTrainingReports MOI.

#### 7.3.4.2 Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
| aIMLEntityId | M | T | F | F | T |
| areConsumerTrainingDataUsed | M | T | F | F | T |
| usedConsumerTrainingData | CM | T | F | F | T |
| confidenceIndication | O | T | F | F | T |
| modelPeformanceTraining | M | T | F | F | T |
| **Attribute related to role** |  |  |  |  |  |
| trainingRequestRef | CM | T | F | F | T |
| trainingProcessRef | M | T | F | F | T |
| lastTrainingRef | CM | T | F | F | T |

#### 7.3.4.3 Attribute constraints

|  |  |
| --- | --- |
| Name | Definition |
| usedConsumerTrainingData Support Qualifier | Condition: The value of areConsumerTrainingDataUsed attribute is ALL or PARTIALLY. |
| trainingRequestRef Support Qualifier | Condition: The AIMLTrainingReport MOI represents the report for the AI/ML model training that was requested by the MnS consumer (via AIMLTrainingRequest MOI). |
| lastTrainingRef Support Qualifier | Condition: The AIMLTrainingReport MOI represents the report for the AI/ML model training that was not initial training (i.e., the model has been trained before). |

#### 7.3.4.4 Notifications

The common notifications defined in clause 7.6 are valid for this IOC, without exceptions or additions.

### 7.3.5 AIMLTrainingProcess

#### 7.3.5.1 Definition

The IOC AIMLTrainingProcess represents the AI/ML training process.

One AIMLTrainingProcess MOI may be instantied for each AIMLTrainingRequest MOI or a set of AIMLTrainingRequest MOIs.

For each AIMLEntity under training, a AIMLTrainingProcess is instantiated, i.e., an AIMLTrainingProcess is associated with exactly one AIMLEntity.The AIMLTrainingProcess may be associated with one or more AIMLTrainingRequest MOI.

The AIMLTrainingProcess does not have to correspond to a specific AIMLTrainingRequest, i.e., a AIMLTrainingRequest does not have to be associated to a specific AIMLTrainingProcess. The AIMLTrainingProcess may be managed separately from the AIMLTrainingRequest MOIs, e.g., the AIMLTrainingRequest MOI may come from consumers which are network functions while the operator may wish to manage the AIMLTrainingProcess that is instantiated following the requests. Thus, the AIMLTrainingProcess may be associated to either one or more AIMLTrainingRequest MOI.

Each AIMLTrainingProcess instance needs to be managed differently from the related AIMLEntity, although the AIMLTrainingProcess may be associated to only one AIMLEntity. For example, the AIMLTrainingProcess may be triggered to start with a specific version of the AIMLEntity and multiple AIMLTrainingProcesse instances may be triggered for different versions of the AIMLEntitys. In either case the AIMLTrainingProcesses are still associated with the same AIMLEntity but are managed separately from the AIMLEntity

Each AIMLTrainingProcess has a priority that may be used to prioritize the execution of different AIMLTrainingProcesses. By default, the priority of the AIMLTrainingProcess may be related in a 1:1 manner with the priority of the AIMLTrainingRequest for which the AIMLTrainingProcess is instantiated.

Each AIMLTrainingProcess may have one or more termination conditions used to define the points at which the AIMLTrainingProcess may terminate .

The "ProgressStatus" attribute represents the status of the AI/ML model training and includes information the MnS consumer can use to monitor the progress and results. The data type of this attribute is "ProcessMonitor" (see TS 28.622 [11]). The following specialisations are provided for this data type for the AI/ML training process:

- The "status" attribute values are "RUNNING", "CANCELLING", "SUSPENDED", "FINISHED", and "CANCELLED". The other values are not used;

- The "timer" attribute is not used;

- When the "status" is equal to "RUNNING" the "progressStateInfo" attribute shall indicate one of the following states: "COLLECTING\_DATA", "PREPARING\_TRAINING\_DATA", "TRAINING".

- No specifications are provided for the "resultStateInfo" attribute. Vendor specific information may be provided though.

When the training is completed with "status" equal to "FINISHED", the MnS producer provides the training report, by creating an AIMLTrainingReport MOI, to the MnS consumer.

7.3.5.2 Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
| aIMLTrainingProcessId | M | T | T | F | T |
| priority | M | T | T | F | T |
| terminationConditions | M | T | T | F | T |
| progressStatus | M | T | F | F | T |
| cancelProcess | O | T | T | F | T |
| suspendProcess | O | T | T | F | T |
| **Attribute related to role** |  |  |  |  |  |
| trainingRequestRef | CM | T | F | F | T |
| trainingReportRef | M | T | F | F | T |

Editor’s note: the relation of progressStateInfo and ProgressStatus is FFS.

Editor's Note: The nature of the terminationConditions is FFS

#### 7.3.5.3 Attribute constraints

|  |  |
| --- | --- |
| Name | Definition |
| trainingRequestRef Support Qualifier | Condition: The AIMLTrainingReport MOI represents the report for the AI/ML model training that was requested by the MnS consumer (via AIMLTrainingRequest MOI). |

#### 7.3.5.4 Notifications

The common notifications defined in clause 7.6 are valid for this IOC, without exceptions or additions.

## 7.4 Data type definitions

### 7.4.1 ModelPerformance <<dataType>>

#### 7.4.1.1 Definition

This data type specifies the performance score of an AI/ML entity when performing inference. The performance score is provided for each inference output.

#### 7.4.1.2 Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
| inferenceOutputName | M | T | F | F | T |
| performanceScore | M | T | F | F | T |
| decisionConfidenceScore | O | T | F | F | T |
| **Attribute related to role** |  |  |  |  |  |
|  |  |  |  |  |  |

#### 7.4.1.3 Attribute constraints

None.

#### 7.4.1.4 Notifications

The notifications specified for the IOC using this <<dataType>> for its attribute(s), shall be applicable.

### 7.4.2 AIMLEntity <<dataType>>

#### 7.4.2.1 Definition

This data type represents the properties of an AI/ML entity which could be either an AI/ML Model or AI/ML-enabled function containing the AI/ML model. AIMLTraining may be requested for either an AI/ML Model or AI/ML-enabled function. The algorithm of AI/ML Model or AI/ML-enbaled function is not to be standardized.

For each AIMLEntity under training, one or more AIMLTrainingProcess are instantiated.

The AIMLEntity may contain 3 types of contexts - TrainingContext which is the context under which the AIMLEntity has been trained, the ExpectedRunTimeContext which is the context where an AIMLEntity is expected to be applied or and the RunTimeContext which is the context where the model is being applied.

#### 7.4.2.2 Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
| aIMLEntityId | M | T | F | F | T |
| inferenceType | M | T | F | F | T |
| aIMLEntityVersion | M | T | F | F | T |
| ExpectedRunTimeContext | O | T | T | F | T |
| TrainingContext | CM | T | F | F | T |
| RunTimeContext | O | T | F | F | T |
| **Attribute related to role** |  |  |  |  |  |
|  |  |  |  |  |  |

#### 7.4.3.3 Attribute constraints

None.

|  |  |
| --- | --- |
| Name | Definition |
| TrainingContext Support Qualifier | Condition: The TrainingContext represents the status and conditions related to training and should be added when training is completed |

#### 7.4.3.4 Notifications

The notifications specified for the IOC using this <<dataType>> for its attribute(s), shall be applicable.

### 7.4.3 AIMLContext <<dataType>>

#### 7.4.3.1 Definition

The AIMLContext represents the status and conditions related to the AIMLEntity. Specially it may be one of three types of context - the ExpectedRunTimeContext, the TrainingContext, and the RunTimeContext.

#### 7.4.3.2 Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
| managedEntityRef | M | T | F | F | F |
| dataProviderRef | M | T | F | F | F |
| **Attribute related to role** |  |  |  |  |  |
|  |  |  |  |  |  |

Editor’s note: Whether other attributes are needed for the context is FFS.

#### 7.4.3.3 Attribute constraints

None.

#### 7.4.1.4 Notifications

The notifications specified for the IOC using this <<dataType>> for its attribute(s), shall be applicable.

## 7.5 Attribute definitions

### 7.5.1 Attribute properties

| Attribute Name | Documentation and Allowed Values | Properties |
| --- | --- | --- |
| aIMLEntityId | It identifies the AI/ML entity.  It is unique in each MnS producer.  allowedValues: N/A. | type: String  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: True |
| candidateTraingDataSource | It provides the address(es) of the candidate training data source provided by MnS consumer. The detailed training data format is vendor specific.  allowedValues: N/A. | type: String  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: True |
| inferenceType | It indicates the type of inference that the AI/ML model supports.  allowedValues: the values of the MDA type (see TS 28.104 [2]), Analytics ID(s) of NWDAF (see TS 23.288 [3]), types of inference for RAN-intelligence, and vendor’s specific extensions. | type: String  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: True |
| areConsumerTrainingDataUsed | It indicates whether the consumer provided training data have been used for the AI/ML model training.  allowedValues: ALL, PARTIALLY, NONE. | type: Enum  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: True |
| usedConsumerTrainingData | It provides the address(es) where lists of the consumer-provided training data are located, which have been used for the AI/ML model training.  allowedValues: N/A. | type: String  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: True |
| trainingRequestRef | It is the DN(s) of the related AIMLTrainingRequest MOI(s).  allowedValues: DN. | type: DN (see TS 32.156 [12])  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: True |
| lastTrainingRef | It is the DN of the AIMLTrainingReport MOI that represents the reports for the last training of the AI/ML model.  allowedValues: DN. | type: DN (see TS 32.156 [12])  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: True |
| confidenceIndication | It indicates the confidence (in unit of percentage) that the AI/ML model would perform for inference on the data with the same distribution as training data.  allowedValues: { 0..100 }. | type: integer  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| aIMLEntityList | It describes the list of aIMLEntity. | type: AIMLEntity  multiplicity: \*  isOrdered: False  isUnique: N/True  defaultValue: None  isNullable: False |
| trainingRequestSource | It descriobes the entity that requested to instantiatethe AIMLTrainingRequest MOI. | type: integer  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| requestStatus | It describes the status of a particular AI/ML training request. T.  allowedValues: NOT\_STARTED, TRAINING\_IN\_PROGRESS, CANCELLING, SUSPENDED, FINISHED, and CANCELLED. | type: Enum  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| aIMLTrainingProcessId | It identifies the training process.  It is unique in each instantiated process in the MnS producer.  allowedValues: N/A. | type: String  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: True |
| priority | It indicates the priority of the training process.  The priority may be used by the AI/ML training to schedule the training processes.  allowedValues: { 0..100 }. | type: ENUM  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| terminationConditions | It indicates the conditions to be considered by the AIMLTraining to terminate a specific training process.  Editor's Note: The specific nature of the termination conditions is FFS  allowedValues: FFS. | type: FFS  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| progressStatus | It indicates the status of the AI/ML training process.  allowedValues: N/A. | type: ProcessMonitor (see TS 28.622 [11])  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| aIMLEntityVersion | It indicates the version number of the AI/ML entity.  allowedValues: N/A. | type: String  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| performanceRequirements | It indicates the expected performance for a trained AI/ML entity when performing on the training data.  allowedValues: N/A. | type: ModelPeformance  multiplicity: \*  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: True |
| performanceTraining | It indicates the performance score of the AI/ML entity when performing on the training data.  allowedValues: N/A. | type: ModelPeformance  multiplicity: \*  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| AIMLTrainingProcess.progressStatus.progressStateInfo | It provides the following specialisation for the "progressStateInfo" attribute of the "ProcessMonitor" data type for the "AIMLTrainingProcess".  When the AI/ML training is in progress, and the "status" is equal to " RUNNING" it provides the more detailed progress information.  allowedValues for "status" = "RUNNING":  - COLLECTING\_DATA  - PREPARING\_TRAINING\_DATA  - TRAINING  The allowed values for "status" = "CANCELLED" are vendor specific. | Type: String  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| inferenceOutputName | It indicates the name of an inference output of an AI/ML entity.  allowedValues: the name of the MDA output IEs (see TS 28.104 [2]), name of analytics output IEs of NWDAF (see TS 23.288 [3]), RAN-intelligence inference output IE name(s), and vendor’s specific extensions. | Type: String  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| performanceScore | It indicates the performance score (in unit of percentage) of an AI/ML entity when performing inference on a specific data set (Note).  The performance metrics may be different for different kinds of AI/ML models depending on the nature of the model. For instance, for numeric prediction, the metric may be accuracy; for classification, the metric may be a combination of precision and recall, like the “F1 score”.  allowedValues: { 0..100 }. | Type: Real  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| cancelRequest | It indicates whether the MnS consumer cancels the AI/ML training request.  Setting this attribute to "TRUE" cancels the AI/ML training request. Cancellation is possible when the requestStatus is the "NOT\_STARTED", " TRAINING\_IN\_PROGRESS", and "SUSPENDED" state. Setting the attribute to "FALSE" has no observable result.  Default value is set to "FALSE".  allowedValues: TRUE, FALSE. | Type: ENUM  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: FALSE  isNullable: False |
| suspendRequest | It indicates whether the MnS consumer suspends the AI/ML training request.  Setting this attribute to "TRUE" suspends the AI/ML training request. Suspension is possible when the requestStatus is the not “FINISHED" state. Setting the attribute to "FALSE" has no observable result.  Default value is set to "FALSE".  allowedValues: TRUE, FALSE. | Type: ENUM  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: FALSE  isNullable: False |
| cancelProcess | It indicates whether the MnS consumer cancels the AI/ML training process.  Setting this attribute to "TRUE" cancels the AI/ML training request. Cancellation is possible when the progressStateInfo is the not “FINISHED" state. Setting the attribute to "FALSE" has no observable result.  Default value is set to "FALSE".  allowedValues: TRUE, FALSE. | Type: ENUM  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: FALSE  isNullable: False |
| suspendProcess | It indicates whether the MnS consumer suspends the AI/ML training process.  Setting this attribute to "TRUE" suspends the AI/ML training request. Suspension is possible when the progressStateInfo is the not “FINISHED", “CANCELLING” or “CANCELLED” state. Setting the attribute to "FALSE" has no observable result.  Default value is set to "FALSE".  allowedValues: TRUE, FALSE. | Type: ENUM  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: FALSE  isNullable: False |
| managedEntityRef | It describes the entities that the MLApp is responsible for managing or optmimizing | Type: DN (see TS 32.156 [12])  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: True |
| dataProviderRef | It describes the entities that have provided or should provide data needed by the MLApp, say for training or inference | Type: DN (see TS 32.156 [12])  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: True |
| traingDataQualityScore | It indicates numerical value that represents the dependability/quality of a given observation and measurement type. The lowest value indicates the lowest level of dependability of the data, i.e., that the data is not usable at all.  allowedValues: { 0..100 }. | Type: Real  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| decisionConfidenceScore | **t**he numerical value that represents the dependability/quality of a given decision generated by the ML-based function. The lowest value indicates the lowest level of dependability of the decisions, i.e., that the data is not usable at all.  allowedValues: { 0..100 }. | Type: Real  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| NOTE: when the performanceScore is to indicate the performance score for AI/ML training, the data set is the training data set. | | |

### 7.5.2 Constraints

|  |  |  |
| --- | --- | --- |
| Name | Affected attribute(s) | Definition |
|  |  |  |

#### 7.3.2.2 Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
| aIMLEntityId | M | T | T | F | T |
| candidateTraingDataSource | O | T | T | F | T |
| trainingRequestSource | M | T | T | F | T |
| requestStatus | M | T | T | F | T |
| expectedRuntimeContext |  |  |  |  |  |
| peformanceRequirements | M | T | T | F | T |
| cancelRequest | O | T | T | F | T |
| suspendRequest | O | T | T | F | T |
| **Attribute related to role** |  |  |  |  |  |
|  |  |  |  |  |  |

|  |
| --- |
| **End of modifications** |