**3GPP TSG-SA5 Meeting #162 *S5-253841***

Goteborg, Sweden, 25 - 29 August 2025

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.1* | | | | | | | | |
| **CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  | **28.105** | **CR** | **DraftCR** | **rev** | **-** | **Current version:** | **19.2.0** |  |
|  | | | | | | | | |
| *For* [***HE******LP***](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: comprehensive instructions can be found at* [*http://www.3gpp.org/Change-Requests*](http://www.3gpp.org/Change-Requests)*.* | | | | | | | | |
|  | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Proposed change affects:*** | UICC apps |  | ME |  | Radio Access Network | **X** | Core Network | **X** |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | |
| ***Title:*** | Input to draftCR Add NRMs on LCM for Federated Learning | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** | ZTE Corporation | | | | | | | | | |
| ***Source to TSG:*** | S5 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | AIML\_MGT\_Ph2 | | | | |  | ***Date:*** | | | 2025-05-09 |
|  |  | | | |  | |  | | |  |
| ***Category:*** | B |  | | | | | ***Release:*** | | | Rel-19 |
|  | *Use one of the following categories:* ***F*** *(correction)* ***A*** *(mirror corresponding to a change in an earlier release)* ***B*** *(addition of feature),* ***C*** *(functional modification of feature)* ***D*** *(editorial modification)*  Detailed explanations of the above categories can be found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | | | | | | | | *Use one of the following releases: Rel-8 (Release 8) Rel-9 (Release 9) Rel-10 (Release 10) Rel-11 (Release 11) … Rel-15 (Release 15) Rel-16 (Release 16) Rel-17 (Release 17) Rel-18 (Release 18)* | |
|  |  | | | | | | | | | |
| ***Reason for change:*** | | Add solution to support FL use case and requirements | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | Add solution to support FL use case and requirements | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | Solution of FL is missing | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | 7.3a.1.2.1, 7.3a.1.2.2, 7.3a.1.2.3, 7.3a.1.2.4, 7.4.w, 7.4.x, 7.4.y, 7.4.z, 7.5.1 | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **Y** | **N** |  | | | |  | | |
| ***Other specs*** | |  | **X** | Other core specifications | | | | TS/TR ... CR ... | | |
| ***affected:*** | |  | **X** | Test specifications | | | | TS/TR ... CR ... | | |
| ***(show related CRs)*** | |  | **X** | O&M Specifications | | | | TS/TR ... CR ... | | |
|  | |  | | | | | | | | |
| ***Other comments:*** | |  | | | | | | | | |
|  | |  | | | | | | | | |
| ***This CR's revision history:*** | |  | | | | | | | | |

|  |
| --- |
| **First modified section** |

##### 7.3a.1.2.1 MLTrainingFunction

###### 7.3a.1.2.1.1 Definition

The IOC MLTrainingFunction represents the model that undertakes ML model training. The MOI of MLTrainingFunction is also the container of the MLTrainingRequest, MLTrainingReport, MLTrainingProcess and ThresholdMonitor MOI(s).

This MLTrainingFunction instance is created by the system (MnS producer) or pre-installed, it can only be deleted by the system.

The ThresholdMonitor contains the list of performance measurements and the corresponding thresholds that are monitored and used to identify the need for ML model re-training by the MnS Producer.

TheML training function represented by MLTrainingFunction MOI supports training of one or more MLModel(s).

The MLTrainingFunction includes information about its applicable type of training, which includes pre-specialised training, fine-tuning, or re-training.

The MLTrainingFunction MOI have a supportedLearningTechnology attribute to indicate the supported learning technology including Reinforcement Learning, Federated Learning and Distributed training. This attribute can enable the ML training MnS producer allowing ML training MnS consumer to query if RL/FL/DL is supported.

An MLTrainingFunction instance may contain a set of ML knowledge instances associated with a set of ML models that have been trained. An MnS consumer can find available ML knowledge by reading the mLKnowledge attribute on the ML MLTrainingFunction. Relatedly, the MnS consumer can find the characteristics of a specific ML knowledge instance by reading the related mLKnowledge. The request for training using MLknowledge is not to be combined with training using collected data – the training function should not provide ML knowledge along side the raw data used for creating the ML knowledge.

###### 7.3a.1.2.1.2 Attributes

The MLTrainingFunction IOC includes attributes inherited from ManagedFunction IOC (defined in TS 28.622 [12]) and the following attributes:

Table 7.3a.1.2.1.2-1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
| supportedLearningTechnology | M | T | F | F | T |
| fLParticipatingInfo | CM | T | F | F | T |
| mLKnowledge | O | T | F | F | T |
| mLTrainingType | M | T | F | F | T |
| **Attribute related to role** |  |  |  |  |  |
| mLModelRepositoryRef | M | T | F | F | T |

###### 7.3a.1.2.1.3 Attribute constraints

**Table 7.3a.1.2.1.3-1**

|  |  |
| --- | --- |
| **Name** | **Definition** |
| fLParticipatingInfo | Condition: FL is supported. |

###### 7.3a.1.2.1.4 Notifications

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

7.3a.1.2.2 MLTrainingRequest

7.3a.1.2.2.1 Definition

The IOC MLTrainingRequest represents the ML model training request that is trigered by the ML training MnS consumer.

To trigger the ML model training process, ML training MnS consumer needs create MLTrainingRequest object instances on the ML training MnS producer. The MLTrainingRequest MOI is contained under one MLTrainingFunction MOI.

The MLTrainingRequest MOI may represent the request for initial ML model training or re-training. For ML model re-training, the MLTrainingRequest is associated to one MLModel for re-training a single ML model, or associated to one MLModelCoordinationGroup.

The MLTrainingRequest includes information about a ML training type to define the type of training requested by the MnS consumer. The training type can be one of the following: (1) initial training, where the MnS consumer requests to train an ML model of which the instance does not exist yet, (2) pre-specialised training, where the ML model is trained on a dataset that is not specific to any particular type of inference, (3) re-training, where the ML model is re-trained on the same type of dataset on which it was previously trained to support the same type of inference, and (4) fine-tuning, where the ML model is trained to adapt it to support a new single type of inference.

The aIMLInferenceName means the inference type will be used for conducting inference.

The MLTrainingRequest has a source to identify where it is coming from, which is represented with trainingRequestSource attribute. This attribute may be used by an ML Training MnS producer to prioritize the training resources for different sources.

Each MLTrainingRequest indicates the expectedRunTimeContext that describes the specific conditions for which the MLModel should be trained.

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

- 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 required training data, with consideration of the consumer’s request provided candidate training data if any. The ML training MnS producer may examine the consumer's provided candidate training data and select none, some or all of them for training. In addition, the ML training MnS producer may select some other training data that are available in order to meet the consumer’s requirements for the ML model training;

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

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

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

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

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

The ML training MnS prodcuer shall delete the corresponding MLTrainingRequest instance in case of the status value turns to "FINISHED" or "CANCELLED". The MnS producer may notify the status of the request to MnS consumer after deleting MLTrainingRequest instance.

For the MLTrainingRequest used to trigger the ML model training of RL, the MLTrainingRequest MOI has an rLRequirement attribute to indicate the requirements of the RL.

For the MLTrainingRequest used to trigger the ML model training of FL, the MLTrainingRequest MOI has an fLRequirement attribute to indicate the requirements of the FL, which represents the requirements for the ML Training Function playing the role of FL Server to train an ML Model using FL.

The MLTrainingRequest can be used to trigger ML-knowledge-based transfer learning. The source ML knowledge should be indicated using the mLKnowledgeName, where the source does not want to reveal the source MLModel. The request for training using ML knowledge is not to be combined with training using collected data – the request cannot be for both mLKnowledgeName and candidateTrainingDataSource.

For the MLTrainingRequest to include clustering criteria, indicating which ML models with multiple contexts belonging to the same producer can form the cluster and trained together, the MLTrainingRequest MOI is enhanced with attribute clusteringInfo containing information that provides the clustering criteria for the ML Models to be trained together.

7.3a.1.2.2.2 Attributes

The MLTrainingRequest IOC includes attributes inherited from Top IOC (defined in TS 28.622 [12]) and the following attributes:

**Table 7.3a.1.2.2.1-1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute name** | **Support Qualifier** | **isReadable** | **isWritable** | **isInvariant** | **isNotifyable** |
| aIMLInferenceName | CM | T | F | F | T |
| candidateTrainingDataSource | O | T | T | F | T |
| trainingDataQualityScore | O | T | T | F | T |
| trainingRequestSource | M | T | T | F | T |
| requestStatus | M | T | F | F | T |
| expectedRuntimeContext | M | T | T | F | T |
| performanceRequirements | M | T | T | F | T |
| rLRequirement | CM | T | T | F | T |
| fLRequirement | CM | T | T | F | T |
| cancelRequest | O | T | T | F | T |
| suspendRequest | O | T | T | F | T |
| trainingDataStatisticalProperties | O | T | T | F | T |
| distributedTrainingExpectation | O | T | T | F | T |
| mLKnowledgeName | CM | T | T | F | T |
| mLTrainingType | M | T | T | F | T |
| expectedInferenceScope | CM | T | T | F | T |
| clusteringInfo | O | T | T | F | T |
| **Attribute related to role** |  |  |  |  |  |
| mLModelRef | CM | T | F | F | T |
| mLModelCoordinationGroupRef | CM | T | F | F | T |

7.3a.1.2.2.3 Attribute constraints

**Table 7.3a.1.2.2.3-1**

|  |  |
| --- | --- |
| **Name** | **Definition** |
| aIMLInferenceName | Condition: MLTrainingRequest MOI represents the request for ML model initial training. |
| mLModelRef | Condition: MLTrainingRequest MOI represents the request for ML model re-training. |
| mLModelCoordinationGroupRef | Condition: ML model joint training is supported. |
| mLKnowledgeName | Condition: Knowledge is indicated only if candidateTrainingDataSource is not indicated |
| rLRequirement | Condition: MLTrainingRequest MOI represents the request for reinforcement learning |
| fLRequirement | Condition: FL is supported |
| expectedInferenceScope | Condition: The MLTrainingRequest is for an ML model pre-specialised training. |

7.3a.1.2.2.4 Notifications

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

##### 7.3a.1.2.3 MLTrainingReport

###### 7.3a.1.2.3.1 Definition

The IOC MLTrainingReport represents the ML model training report that is provided by the training MnS producer. The MLTrainingReport is associated with one MLModel or one MLModelCoordinationGroup.

The MLTrainingReport instance is created by the training MnS producer automatically when creating an MLTrainingRequest instance.

The MLTrainingReport MOI is contained under one MLTrainingFunction MOI.

###### 7.3a.1.2.3.2 Attributes

The MLTrainingReport IOC includes attributes inherited from Top IOC (defined in TS 28.622 [12]) and the following attributes:

Table 7.3a.1.2.3.2-1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
| usedConsumerTrainingData | CM | T | F | F | T |
| modelConfidenceIndication | O | T | F | F | T |
| modelPerformanceTraining | M | T | F | F | T |
| areNewTrainingDataUsed | M | T | F | F | T |
| modelPerformanceValidation | O | T | F | F | T |
| flReportPerClient | CM | 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 |
| mLModelGeneratedRef | M | T | F | F | T |
| mLModelCoordinationGroupGeneratedRef | CM | T | F | F | T |
| mLModelRef | M | T | F | F | T |

###### 7.3a.1.2.3.3 Attribute constraints

Table 7.3a.1.2.3.3-1

|  |  |
| --- | --- |
| Name | Definition |
| trainingRequestRef | Condition: The MLTrainingReport MOI represents the report for the ML model training that was requested by the MnS consumer (via MLTrainingRequest MOI). |
| lastTrainingRef | Condition: The MLTrainingReport MOI represents the report for the ML model training that was not ML model initial training (i.e. the model has been trained before). |
| mLModelCoordinationGroupGeneratedRef | Condition: The MLTrainingReport MOI represents the report for ML model joint training. |
| flReportPerClient | Condition: FL is supported. |

###### 7.3a.1.2.3.4 Notifications

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

##### 7.3a.1.2.4 MLTrainingProcess

###### 7.3a.1.2.4.1 Definition

The IOC MLTrainingProcess represents the ML model training process. When a ML model training process starts, an instance of the MLTrainingProcess is created by the MnS Producer and notification is sent to MnS consumer who has subscribed to it.The MnS producer can delete the MLTrainingProcess instance whose attribute status equals to "FINISHED" or "CANCELLED" automatically.

One MLTrainingProcess MOI may be instantiated for each MLTrainingRequest MOI or a set of MLTrainingRequest MOIs.

For each MLModel under training, a MLTrainingProcess is instantiated, i.e. an MLTrainingProcess is associated with one MLModel or one MLModelCoordinationGroup.The MLTrainingProcess may be associated with one or more MLTrainingRequest MOI.

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

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

Each MLTrainingProcess has a priority that may be used to prioritize the execution of different MLTrainingProcess instances.

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

The "progressStatus" attribute represents the status of the ML model training and includes information the ML training MnS consumer can use to monitor the progress and results. The data type of this attribute is "ProcessMonitor" (see 3GPP TS 28.622 [12]). The following specializations are provided for this data type for the ML model 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 MLT MnS producer provides the training report, by creating an MLTrainingReport MOI, to the MLT MnS consumer.

###### 7.3a.1.2.4.2 Attributes

The MLTrainingProcess IOC includes attributes inherited from Top IOC (defined in TS 28.622 [12]) and the following attributes:

Table 7.3a.1.2.4.2-1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
| 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 |
| mLModelGeneratedRef | CM | T | F | F | T |
| mLModelRef | M | T | F | F | T |
| participatingFLClientRefList | CM | T | F | F | T |

###### 7.3a.1.2.4.3 Attribute constraints

Table 7.3a.1.2.4.3-1

|  |  |
| --- | --- |
| Name | Definition |
| trainingRequestRef | Condition: The MLTrainingReport MOI represents the report for the ML model training that was requested by the training MnS consumer (via MLTrainingRequest MOI). |
| mLModelGeneratedRef | Condition: The MLTrainingProcess MOI is instantiated to retrain an existing MLModel. |
| participatingFLClientRefList | Condition: FL is supported. |

|  |
| --- |
| **Next modified section** |

### 7.4.w FLParticipatingInfo <<dataType>>

#### 7.4.w.1 Definition

This <<dataType>> defines the FL capability that an ML training function supports.

#### 7.4.w.2 Attributes

Table 7.4.x.2-1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
| fLRole | M | T | F | F | T |
| fLUsageState | O | T | F | F | T |
| candidateFLClientRefList | CO | T | F | F | T |

Note: FL capability should be aligned with RL NRM

#### 7.4.w.3 Attribute constraints

**Table 7.4.w.3-1**

|  |  |
| --- | --- |
| **Name** | **Definition** |
| candidateFLClientRefList | Condition: The ML training function plays the role of “FL\_Server”. |

#### 7.4.w.4 Notifications

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

### 7.4.x FLRequirement <<dataType>>

#### 7.4.x.1 Definition

The <<datatype>> FLRequirement represents the requirement for the ML training function that plays the role of FL server to train an ML model using FL.

#### 7.4.x.2 Attributes

The FLRequirement <<datatype>>includes attributes inherited from Top IOC (defined in TS 28.622 [12]) and the following attributes:

Table 7.3a.1.2.x.2-1

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

#### 7.4.x.3 Attribute constraints

*None*

#### 7.4.x.4 Notifications

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

### 7.4.y FLClientSelectionCriteria <<dataType>>

#### 7.4.y.1 Definition

This data type specifies the criteria for selecting the FL Clients by the FL server.It defines the conditions that FL Clients should meet to participate in FL.

#### 7.4.y.2 Attributes

Table 7.4.y.2-1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
| minimumAvailableDataSamples | M | T | T | F | T |
| minimumAvailableTimeDuration | M | T | T | F | T |
| minimumInterimModelPerformance | O | T | T | F | T |
| servingGeoArea | O | T | T | F | T |
| isRenewableSource | O | T | T | F | T |
| renewableSourceType | O | T | T | F | T |
| carbonEmissionInfo | O | T | T | F | T |
| RenewableEnergyInfo | O | T | T | F | T |
| clientRedundancy | O | T | T | F | T |
| trainingDataWithOrWithoutOutliers | O | T | T | F | T |
| uniformlyDistributedTrainingData | O | T | T | F | T |

#### 7.4.y.3 Attribute constraints

None.

#### 7.4.y.4 Notifications

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

### 7.4.z FLReportPerClient <<dataType>>

#### 7.4.z.1 Definition

This data type specifies the detailed report for an FL from the FL Server for each participating FL Client.

#### 7.4.z.2 Attributes

Table 7.4.z.2-1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
| ClientRef | M | T | F | F | T |
| numberOfDataSamplesUsed | O | T | F | F | T |
| trainingTimeDuration | O | T | F | F | T |
| ModelPerformanceOnClient | O | T | F | F | T |

#### 7.4.z.3 Attribute constraints

None.

#### 7.4.z.4 Notifications

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

|  |
| --- |
| **Next modified section** |

### 7.5.1 Attribute properties

Table 7.5.1-1

| Attribute Name | Documentation and Allowed Values | Properties |
| --- | --- | --- |
| mLModelId | It identifies the ML model.  It is unique in each MnS producer.  allowedValues: N/A. | type: String  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| candidateTrainingDataSource | 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: False |
| aIMLInferenceName | It indicates the type of inference that the ML model supports.  allowedValues: see clause 7.4.10 | type: AIMLInferenceName  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| mDAType | It indicates the type of inference that the ML model for MDA supports.  The detailed definition and corresponding allowed values for mDAType see TS 28.104 [2]. | type: MDAType (TS 28.104 [2])  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| nwdafAnalyticsType | It indicates the type of inference that the ML model for NWDAF supports.  The detailed definition and corresponding allowed values for nwdafAnalyticsID see NwdafEvent in TS 29.520 [20]. | type: NwdafEvent (TS 29.520 [20])  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| ngRanInferenceType | It indicates the type of inference that the ML model for NG-RAN supports.  The detailed definition and corresponding allowed values for ngRanInferenceType see clause 7.4a.1 | type: NgRanInferenceType  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| vSExtensionType | It indicates the type of inference that is vendor's specific extension.  allowedValues: N/A. | type: String  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| usedConsumerTrainingData | It provides the address(es) where lists of the consumer-provided training data are located, which have been used for the ML model training.  allowedValues: N/A. | type: String  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |
| trainingRequestRef | It is the DN(s) of the related MLTrainingRequest MOI(s). | type: DN  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |
| trainingProcessRef | It is the DN(s) of the related MLTrainingProcess MOI(s) that produced the MLTrainingReport. | type: DN  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| trainingReportRef | It is the DN of the MLTrainingReport MOI that represents the reports of the ML model training. | type: DN  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| lastTrainingRef | It is the DN of the MLTrainingReport MOI that represents the reports for the last training of the ML model(s). | type: DN  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| modelConfidenceIndication | It indicates the average confidence value (in unit of percentage) that the ML model would perform for inference on the data with the same distribution as training data.  Essentially, this is a measure of degree of the convergence of the trained ML model.  allowedValues: { 0..100 }. | type: Integer  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| trainingRequestSource | It identifies the entity that requested to instantiate the MLTrainingRequest MOI.  This attribute is the DN of a managed entity, otherwise, it is a String. | type: <<Choice>>  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| MLTrainingRequest.requestStatus | It describes the status of a particular ML model training request.  allowedValues: NOT\_STARTED, IN\_PROGRESS, CANCELLING, SUSPENDED, FINISHED, and CANCELLED. | type: Enum  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| mLTrainingProcessId | 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: False |
| priority | It indicates the priority of the training process.  The priority may be used by the ML model training to schedule the training processes. Lower value indicates a higher priority.  allowedValues: { 0..65535 }. | type: Integer  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: 0  isNullable: False |
| terminationConditions | It indicates the conditions to be considered by the ML training MnS producer to terminate a specific training process.  allowedValues: N/A. | type: String  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| progressStatus | It indicates the status of the process.  allowedValues: N/A. | type: ProcessMonitor  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| MLUpdateProcess.cancelProcess | It allows the ML update MnS consumer to cancel the ML update process.  Setting this attribute to "TRUE" cancels the ML update process. Setting the attribute to "FALSE" has no observable result.  allowedValues: TRUE, FALSE. | type: Boolean  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: FALSE  isNullable: False |
| MLUpdateProcess.suspendProcess | It allows the ML update MnS consumer to suspend the ML update process.  Setting this attribute to "TRUE" suspends the ML update process. The process can be resumed by setting this attribute to “FALSE” when it is suspended. Setting the attribute to "FALSE" has no observable result.  allowedValues: TRUE, FALSE. | type: Boolean  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: FALSE  isNullable: False |
| mLModelVersion | It indicates the version number of the ML model.  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 ML model when performing on the training data.  allowedValues: N/A. | type: ModelPerformance  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |
| modelPerformanceTraining | It indicates the performance score of the ML model when performing on the training data.  allowedValues: N/A. | type: ModelPerformance  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |
| MLTrainingProcess.progressStatus.progressStateInfo | It provides the following specialization for the “progressStateInfo“ attribute of the “ProcessMonitor“ data type for the “MLTrainingProcess.progressStatus“.  When the ML model training is in progress, and the " mLTrainingProcess.progressStatus.status " is equal to "RUNNING", it provides the more detailed progress information.  allowedValues for " mLTrainingProcess.progressStatus.status " = "RUNNING":  - “COLLECTING\_DATA”  - “PREPARING\_TRAINING\_DATA”  - “TRAINING” + DN of the MLModel being trained  The allowed values for " mLTrainingProcess.progressStatus.status " = "CANCELLING" are vendor specific.  The allowed values for " mLTrainingProcess.progressStatus.status " = "NOT\_STARTED" 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 ML model.  allowedValues: the name of the MDA output IEs (see 3GPP TS 28.104 [2]), name of analytics output IEs of NWDAF (see TS 23.288 [3]), RAN inference output IE name(s), and vendor's specific extensions. | type: String  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| performanceMetric | It indicates the performance metric used to evaluate the performance of an ML model, e.g. "accuracy", "precision", "F1 score", etc.  allowedValues: N/A. | type: String  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| performanceScore | It indicates the performance score (in unit of percentage) of an ML model when performing inference on a specific data set (Note).  The performance metrics may be different for different kinds of 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: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| MLTrainingRequest.cancelRequest | It allows the ML training MnS consumer to cancel the ML model training request.  Setting this attribute to "TRUE" cancels the ML model training request. The request can be resumed by setting this attribute to "FALSE" when it is suspended. Cancellation is possible when the requestStatus is the "NOT\_STARTED", " IN\_PROGRESS", and "SUSPENDED" state. Setting the attribute to "FALSE" has no observable result.  allowedValues: TRUE, FALSE. | type: Boolean  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: FALSE  isNullable: False |
| MLTrainingRequest.suspendRequest | It allows the ML training MnS consumer to suspend the ML model training request.  Setting this attribute to "TRUE" suspends the ML model training process. Suspension is possible when the requestStatus is not the "FINISHED" state. Setting the attribute to "FALSE" has no observable result.  allowedValues: TRUE, FALSE. | type: Boolean  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: FALSE  isNullable: False |
| MLTrainingProcess.cancelProcess | It allows the ML training MnS consumer to cancel the ML model training process.  Setting this attribute to “TRUE“ cancels the ML model training process. Cancellation is possible when the “mLTrainingProcess.progressStatus.status“ is not the “FINISHED“ state. Setting the attribute to “FALSE“ has no observable result.  allowedValues: TRUE, FALSE. | type: Boolean  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: FALSE  isNullable: False |
| MLTrainingProcess.suspendProcess | It allows the ML training MnS consumer to suspend the ML model training process.  Setting this attribute to "TRUE" suspends the ML model training process. The process can be resumed by setting this attribute to “FALSE” when it is suspended. Suspension is possible when the " mLTrainingProcess.progressStatus.status" is not the "FINISHED", "CANCELLING" or "CANCELLED" state. Setting the attribute to "FALSE" has no observable result.  allowedValues: TRUE, FALSE. | type: Boolean  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: FALSE  isNullable: False |
| inferenceEntityRef | It describes the target entities that will use the ML model for inference. | type: DN  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |
| dataProviderRef | It describes the entities that have provided or should provide data needed by the ML model e.g. for training or inference | type: DN  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |
| areNewTrainingDataUsed | It indicates whether new training data are used for the ML model training.  allowedValues: TRUE, FALSE. | type: Boolean  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| trainingDataQualityScore | 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 | It is the numerical value that represents the dependability/quality of a given decision generated by the AI/ML inference 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 |
| expectedRuntimeContext | This describes the context where an MLModel is expected to be applied.  allowedValues: N/A | type: MLContext  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| trainingContext | This specifies the context under which the MLModel has been trained.  allowedValues: N/A | type: MLContext  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| runTimeContext | This specifies the context where the MLmodel or model is being applied.  allowedValues: N/A | type: MLContext  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| MLTrainingRequest.mLModelRef | It identifies the DN of the MLModel requested to be trained. | type: DN  multiplicity: 0..1  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |
| MLTrainingReport.mLModelGeneratedRef | It identifies the DN of the MLModel generated by the ML model training. | type: DN  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| mLModelRepositoryRef | It identifies the DN of the MLModelRepository. | type: DN  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| mLRepositoryId | It indicates the unique ID of the ML repository. | type: String  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| modelPerformanceValidation | It indicates the performance score of the ML model when performing on the validation data.  allowedValues: N/A | type: ModelPerformance  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |
| dataRatioTrainingAndValidation | It indicates the ratio (in terms of quantity of data samples) of the training data and validation data used during the training and validation process. It is represented by the percentage of the validation data samples in the total training data set (including both training data samples and validation data samples). The value is an integer reflecting the rounded number of percent \* 100.    allowedValues: { 0 .. 100 }. | type: Integer  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| MLTestingRequest.requestStatus | It describes the status of a particular ML testing request.  allowedValues: NOT\_STARTED, IN\_PROGRESS, CANCELLING, SUSPENDED, FINISHED, and CANCELLED. | type: Enum  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| MLTestingRequest.cancelRequest | It allows the ML testing MnS consumer to cancel the ML testing request.  Setting this attribute to "TRUE" cancels the ML testing request. Cancellation is possible when the requestStatus is the "NOT\_STARTED", " IN\_PROGRESS", and "SUSPENDED" state. Setting the attribute to "FALSE" has no observable result.  allowedValues: TRUE, FALSE. | type: Boolean  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: FALSE  isNullable: False |
| MLTestingRequest.suspendRequest | It allows the ML testing MnS consumer to suspend the ML testing request.  Setting this attribute to "TRUE" suspends the ML testing request. The request can be resumed by setting this attribute to “FALSE” when it is suspended. Suspension is possible when the requestStatus is not the "FINISHED" state. Setting the attribute to "FALSE" has no observable result.  allowedValues: TRUE, FALSE. | type: Boolean  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: FALSE  isNullable: False |
| MLTestingRequest.mLModelRef | It identifies the DN of the MLModel requested to be tested. | type: DN  Multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| modelPerformanceTesting | It indicates the performance score of the ML model when performing on the testing data.  allowedValues: N/A. | type: ModelPerformance  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |
| mLTestingResult | It provides the address where the testing result is provided.  The detailed testing result format is vendor specific.  allowedValues: N/A. | type: String  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| testingRequestRef | It identifies the DN of the MLTestingRequest MOI. | type: DN  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| supportedPerformanceIndicators | This parameter lists specific PerformanceIndicator(s) of an ML model.  allowedValues: N/A. | type: SupportedPerfIndicator  multiplicity: 1..\*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |
| performanceIndicatorName | It indicates the identifier of the specific performance indicator.  allowedValues: N/A | type: String  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| isSupportedForTraining | It indicates whether the specific performance indicator is supported a performance metric of ML model training for the ML model.  allowedValues: TRUE, FALSE. | type: Boolean  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: FALSE  isNullable: False |
| isSupportedForTesting | It indicates whether the specific performance indicator is supported a performance metric of ML model testing for the ML model.  allowedValues: TRUE, FALSE. | type: Boolean  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: FALSE  isNullable: False |
| mLUpdateProcessRef | It is the DN of the mLUpdateProcess MOI that represents the process of updating an ML model. | type: DN  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| mLUpdateRequestRefList | It is the list of DN of the MLUpdateRequest MOI that represents an  ML update request. | type: DN  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |
| mLUpdateReportRef | It is the DN of the MLUpdateReport MOI that represents an ML update report. | type: DN  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| mLUpdateReportingPeriod | It specifies the time duration upon which the MnS consumer expects the ML update is reported. | type: TimeWindow  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| availMLCapabilityReport | It represents the available ML capabilities.  allowedValues: N/A. | type: AvailMLCapabilityReport multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| updatedMLCapability | It represents the updated ML capabilities.  allowedValues: N/A. | type: AvailMLCapabilityReport multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| availMLCapabilityReportID | It identifies the available ML capability report.  allowedValues: N/A. | type: String  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| newCapabilityVersionId | It indicates the specific version of AI/ML capabilities to be applied for the update. It is typically the one indicated by the MLCapabilityVersionID in a newCapabilityVersion | type: String  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |
| mlCapabilityVersionId | It indicates the version of ML capabilities that is available for the update. | type: String  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |
| performanceGainThreshold | It defines the minimum performance gain as a percentage that shall be achieved with the capability update, i.e., the difference in the performances between the existing capabilities and the new capabilities should be at least performanceGainThreshold otherwise the new capabilities should not be applied.  Allowed value: float between 0.0 and 100.0 | type: ModelPerformance  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |
| expectedPerformanceGains | It indicates the expected performance gain if/when the AI/ML capabilities of the respective network function are updated with/to the specific set of newly available AI/ML capabilities. | type: ModelPerformance  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |
| updateTimeDeadline | It indicates the maximum as stated in the MLUpdate request that should be taken to complete the update | type: TimeWindow  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| MLUpdateReport.mLModelRefList | It indicates the DN of MLModel instances that can be updated. | type: DN  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |
| MLUpdateRequest.requestStatus | It describes the status of a particular ML update request.  allowedValues: NOT\_STARTED, IN\_PROGRESS, CANCELLING, SUSPENDED, FINISHED, and CANCELLED. | type: Enum  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| MLUpdateRequest.cancelRequest | It allows the MnS consumer to cancel the ML update request.  Setting this attribute to "TRUE" cancels the ML update request. Cancellation is possible when the requestStatus is the "NOT\_STARTED", " IN\_PROGRESS", and "SUSPENDED" state. Setting the attribute to "FALSE" has no observable result.  allowedValues: TRUE, FALSE. | type: Boolean  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: FALSE  isNullable: False |
| MLUpdateRequest.suspendRequest | It allows the MnS consumer to suspend the ML update request.  Setting this attribute to "TRUE" suspends the ML update request. The request can be resumed by setting this attribute to “FALSE” when it is suspended. Suspension is possible when the requestStatus is not the "FINISHED" state. Setting the attribute to "FALSE" has no observable result.  allowedValues: TRUE, FALSE. | type: Boolean  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: FALSE  isNullable: False |
| memberMLModelRefList | It identifies the list of member ML models within an ML model coordination group. | type: DN  multiplicity: 2..\*  isOrdered: True  isUnique: True  defaultValue: None  isNullable: False |
| MLTrainingRequest.mLModelCoordinationGroupRef | It identifies the DN of the MLModelCoordinationGroup requested to be trained. | type: DN  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| MLTrainingReport.mLModelCoordinationGroupGeneratedRef | It identifies the DN of the MLModelCoordinationGroup generated by ML model joint training. | type: DN  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| MLTestingRequest.mLModelCoordinationGroupRef | It identifies the DN of the MLModelCoordinationGroup requested to be tested. | type: DN  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| retrainingEventsMonitorRef | It indicates the DN of the ThresholdMonitor MOI that indicates the performance measurements and its corresponding thresholds to be used by MnS producer to initiate the re-training of the MLModel. | type: DN  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| MLModelLoadingRequest.requestStatus | It describes the status of a particular ML model loading request.  allowedValues: NOT\_STARTED, IN\_PROGRESS, CANCELLING, SUSPENDED, FINISHED, and CANCELLED. | type: Enum  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| MLModelLoadingRequest.cancelRequest | It allows the MnS consumer to cancel the ML model loading request.  Setting this attribute to "TRUE" cancels the ML model loading. Cancellation is possible when the requestStatus is the "NOT\_STARTED", " IN\_PROGRESS", and "SUSPENDED" state. Setting the attribute to "FALSE" has no observable result.  allowedValues: TRUE, FALSE. | type: Boolean  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: FALSE  isNullable: False |
| MLModelLoadingRequest.suspendRequest | It allows the MnS consumer to suspend the ML model loading request.  Setting this attribute to "TRUE" suspends the ML model loading request. The request can be resumed by setting this attribute to “FALSE” when it is suspended. Suspension is possible when the requestStatus is not the "FINISHED" state. Setting the attribute to "FALSE" has no observable result.  allowedValues: TRUE, FALSE. | type: Boolean  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: FALSE  isNullable: False |
| mLModelToLoadRef | It identifies the DN of a trained MLModel requested to be loaded to the target inference function(s). | type: DN  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| policyForLoading | It provides the policy for controlling ML model loading triggered by the MnS producer.  This policy contains two thresholds in the thresholdList attribute. The first threshold is related to the ML model to be loaded, and the second threshold is related to the existing ML model being used for inference. | type: AIMLManagementPolicy  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| thresholdList | It provides the list of threshold. | type: ThresholdInfo  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |
| MLModelLoadingProcess.progressStatus.progressStateInfo | It provides the following specialization for the "progressStateInfo" attribute of the "ProcessMonitor" data type for the "MLModelLoadingProcess.progressStatus".  When the ML model loading is in progress, and the " MLModelLoadingProcess.progressStatus.status " is equal to "RUNNING", it provides the more detailed progress information.  allowedValues for " MLModelLoadingProcess.progressStatus.status " = "RUNNING":  The allowed values for " MLModelLoadingProcess.progressStatus.status " = "CANCELLING" are vendor specific.  The allowed values for " MLModelLoadingProcess.progressStatus.status " = "NOT\_STARTED" are vendor specific. | type: String  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| MLModelLoadingProcess.cancelProcess | It allows the MnS consumer to cancel the ML model loading process.  Setting this attribute to "TRUE" cancels the process. Cancellation is possible when the "MLModelLoadingProcess.progressStatus.status" is not the "FINISHED" state. Setting the attribute to "FALSE" has no observable result.  allowedValues: TRUE, FALSE. | type: Boolean  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: FALSE  isNullable: False |
| MLModelLoadingProcess.suspendProcess | It allows the MnS consumer to suspend the ML model loading process.  Setting this attribute to "TRUE" suspends the process. The process can be resumed by setting this attribute to "FALSE" when it is suspended. Suspension is possible when the "MLModelLoadingProcess.progressStatus.status" is not the "FINISHED", "CANCELLING" or "CANCELLED" state. Setting the attribute to "FALSE" has no observable result.  allowedValues: TRUE, FALSE. | type: Boolean  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: FALSE  isNullable: False |
| mLModelLoadingRequestRef | It identifies the DN of the associated MLModelLoadingRequest. | type: DN  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| mLModelLoadingPolicyRef | It identifies the DN of the associated MLModelLoadingPolicyRef. | type: DN  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| loadedMLModelRef | It identifies the DN of the MLModel that has been loaded to the inference function. | type: DN  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| activationStatus | It describes the activation status.  allowedValues: ACTIVATED, DEACTIVATED. | type: Enum  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| AIMLManagementPolicy.managedActivationScope | It provides a list of sub scopes for which ML inference is activated as triggered by a policy on the MnS producer. For example, the sub scopes may be a list of cells or of geographical areas. The list is an ordered list indicating the inference is activated for the first sub scope and gradually extended to the next sub scope if the policy evaluates to true.  allowedValues: N/A | type: ManagedActivationScope  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| AIMLInferenceFunction.managedActivationScope | It provides a list of sub scopes for which ML inference is activated as triggered by a policy on the MnS producer. For example, the sub scopes may be a list of cells or of geographical areas. The list is an ordered list indicating the inference is activated for the first sub scope and gradually extended to the next sub scope if the policy evaluates to true.  allowedValues: N/A | type: AIMLManagementPolicy  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| ManagedActivationScope.dNList | It indicates the list of DN, the list is an ordered list indicating the inference is activated for the first sub scope and gradually extended to the next sub scope.  allowedValues: N/A | type: DN  multiplicity: \*  isOrdered: True  isUnique: True  defaultValue: None  isNullable: False |
| ManagedActivationScope.timeWindow | It indicates the list of time window; the list is an ordered list indicating the inference is activated for the first sub scope and gradually extended to the next sub scope.  allowedValues: N/A | type: TimeWindow  multiplicity: \*  isOrdered: True  isUnique: True  defaultValue: None  isNullable: False |
| ManagedActivationScope.geoPolygon | It indicates the list of GeoArea, the list is an ordered list indicating the inference is activated for the first sub scope and gradually extended to the next sub scope.  allowedValues: N/A | type: GeoArea  multiplicity: \*  isOrdered: True  isUnique: True  defaultValue: None  isNullable: False |
| usedByFunctionRefList | It provides the DNs of the functions supported by the AIMLInferenceFunction.  allowedValues: N/A | type: DN  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |
| inferenceOutputId | It identifies an inference output within an AIMLinferenceReport. | type: String  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |
| inferenceOutputs | It indicates the Outputs that have been derived by the AIMLInferenceFunction instance from a specific ML model.  Each ML model, inferenceOutputs may be a set of values.  allowedValues: N/A. | type: InferenceOutput  multiplicity: 1..\*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |
| inferencePerformance | It indicates the performance score of the ML model during Inference.  allowedValues: N/A. | type: ModelPerformance  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |
| inferenceOutputTime | It indicates the time at which the inference output is generated.  allowedValues: N/A | type: DateTime  multiplicity: \*  isOrdered: True  isUnique: True  defaultValue: None  isNullable: False |
| outputResult | It indicates the result of an inference. | type: AttributeValuePair  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: Null  isNullable: False |
| mLCapabilitiesInfoList | It indicates information about what an ML model can generate inference for.  allowedValues: N/A. | type: MLCapabilityInfo  multiplicity: 1..\*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |
| capabilityName | It indicates the name of a capability for which an ML model can generate inference. The capability is defined by Mns producer which can be traffic analysis capability, coverage analysis capability, mobility analysis capability or vendor specific extensions.  allowedValues: N/A. | type: String  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| mLCapabilityParameters | It indicates a set of optional parameters that apply for an aIMLInferenceName capabilityName.  allowedValues: N/A | type: AttributeValuePair  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |
| aIMLInferenceReportRefList | It indicates a list of DN of the AIMLInferenceReport MOI that represents an AIML inference report. | type: DN  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False | |
| mLModelRefList | It identifies the list of MLModel DN. | type: DN  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False | |
| fLParticipatingInfo | It indicates the information of the ML Training Function participating in the FL process. | type: FLParticipatingInfo  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False | |
| FLParticipatingInfo.fLRole | It indicates the role that an ML training function plays in FL.  allowedValues: FL\_Server, FL\_Client. | type: Enum  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False | |
| FLParticipatingInfo.fLUsageState | This attribute defines the FL state of the MLTrainingFunction for training a given ML model.  allowedValues:  “TRUE” indicates that the MLTrainingFunction is available for starting a new FL process for the ML model;  “FALSE” indicates that the MLTrainingFunction is unavailable for a starting new FL process for the ML model. | type: Boolean  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: False  isNullable: False | |
| FLParticipatingInfo.candidateFLClientRefList | It identifies the DNs of the MLTrainingFunction instances that are capable of acting as the FL client. | type: DN  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False | |
| fLRequirement | It indicates the requirements of FL training. | type: FLRequirement  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False | |
| fLClientSelectionCriteria | It provides the criteria for selecting the FL Clients for an FL. | type: FLClientSelectionCriteria  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False | |
| FLClientSelectionCriteria.minimumAvailableDataSamples | It indicates the minimum number of data samples can be used for training. | type: Integer  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False | |
| FLClientSelectionCriteria.minimumAvailableTimeDuration | It indicates the minimum time length that the FL Client is available to participate into an FL, in unit of minutes. | type: Integer  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False | |
| FLClientSelectionCriteria.minimumInterimModelPerformance | It indicates the minimum training performance score for an interim model on an FL Client. | type: ModelPerformance  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False | |
| FLClientSelectionCriteria.uniformlyDistributedTrainingData | It indicates the need for using training data that are uniformly distributed according to the different aspects (e.g., equivalent data samples for each UE in the training data, equivalent data samples for each type of slice in the training data, equivalent data samples from each GeoArea in the training data) of the aIMLinferenceName.  allowedValues: TRUE, FALSE. | type: Boolean  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: FALSE  isNullable: False | |
| FLClientSelectionCriteria.trainingDataWithOrWithoutOutliers | It indicates that the training data samples should consider or disregard data samples that are at the extreme boundaries of the value range.  allowedValues: TRUE, FALSE. | type: Boolean  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: FALSE  isNullable: False | |
| FLClientSelectionCriteria.servingGeoArea | It indicates the serving geographical area of an FL Client. | type: GeoArea  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False | |
| FLClientSelectionCriteria.isRenewableEnergySource | This defines that a FL client can be selected by FL server only if the client is using any renewable energy source. Its values can be TRUE or FALSE, where TRUE means the FL client must be using a renewable energy source and FALSE means the energy source of FL client does not matter | type: Boolean  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False | |
| FLClientSelectionCriteria.renewableEnergySourceType | This defines that a FL client can be selected by FL server only if it is using a particular type of renewable energy source.  allowedValues: WIND, SOLAR, BIOGAS, BIOFUEL, AEROTHERMAL, GEOTHERMAL, HYDROPOWER, BIOMASS, LANDFILLGAS. | type: ENUM  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False | |
| FLClientSelectionCriteria.carbonEmissionInfo | This defines the maximum carbon emission value, below which the FL client should emit in order to be chosen by FL server for training a ML model as FL client. Its value can in terms of kilogram of CO2 equivalent to consumed energy (kWh). | type: Float  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False | |
| RenewableEnergyInfo | It defines that a certain amount of renewable sourced energy, which at least, the FL client should be using for its operation in order to be selected as FL client. It can be defined either in terms of percentage or in terms of a ratio (e.g. ratio of renewable energy consumed by total energy consumed) | type: Real  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False | |
| FLClientSelectionCriteria.clientRedundancy | This defines that if the FL client needs to have some type of redundancy to handle client dropouts gracefully in order to be selected by FL server to train a ML model. Its values can be TRUE or FALSE, where TRUE means the FL client must have some type of redundancy and FALSE means the redundancy of FL client does not matter for its selection | type: Boolen  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False | |
| flReportPerClient | This report is provide by the server to the consumer. | type: FLReportPerClient  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False | |
| FLReportPerClient.ClientRef | It identifies the DN of MLTrainingFunction that plays the role of FL Client. | type: DN  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False | |
| FLReportPerClient.numberOfDataSamplesUsed | It indicates the number of data samples that have been used in the ML training. | type: Integer  multiplicity: 1  isOrdered: N/A  isUnique: N/ATrue  defaultValue: None  isNullable: False | |
| FLReportPerClient.trainingTimeDuration | It indicates the time window that the FL Client/FL Server can participate into an FL process. | type: TimeWindow  multiplicity: 1..\*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False | |
| FLReportPerClient. ModelPerformanceOnClient | It indicates the performance score of the final global ML model running on the local data set of the FL Client. | type: ModelPerformance  multiplicity: 2..\*  isOrdered: False  isUnique: False  defaultValue: None  isNullable: False | |
| participatingFLClientRefList | It identifies the list of DN of the MLTrainingFunction that participated the FL process as FL Clients. | type: DN  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False | |
| NOTE: When the performanceScore is to indicate the performance score for ML model training, the data set is the training data set. When the performanceScore is to indicate the performance score for ML validation, the data set is the validation data set. When the performanceScore is to indicate the performance score for ML model testing, the data set is the testing data set. | | |

|  |
| --- |
| **Next modified section** |