**TSG-SA5 Meeting #162 *S5-253976d1***

**, , – Revision of S5-253474**

|  |
| --- |
| *CR-Form-v12.3* |
| **CHANGE REQUEST** |
|  |
|  |  | **CR** | **draftCR** | **rev** | **-** | **Current version:** | **19.2.0** |  |
|  |
| *For* [***HE******LP***](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: compr**ehensive 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 Rel-19 TS 28.105 Training types in Training NRM fragment |
|  |  |
| ***Source to WG:*** |  |
| ***Source to TSG:*** | SA5 |
|  |  |
| ***Work item code:*** | AIML\_MGT\_Ph2 |  | ***Date:*** | 2025-08-14 |
|  |  |  |  |  |
| ***Category:*** | **F** |  | ***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 canbe 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-17 (Release 17)Rel-18 (Release 18)Rel-19 (Release 19) Rel-20 (Release 20)* |
|  |  |
| ***Reason for change:*** | **First Issue**: **MLTrainingRequest IOC*** ExpectedInferenceScope: Having both expectedInferenceScope (can be present) and aimlInferenceName (must always be present, i..e cardinality =1) in this IOC is problematic. If a MnS producer supports pre-specialized training, there are no design-time conditions that prevents the MOI two have values for both expectedInferenceScope and aimlInferenceName. This causes problem on MnS producer behavior.
* Attribute constraints: The conditions captured in the attribute constraints table are wrong for most of attributes.
* Attribute properties: The following following attributes need attention: “aimlInferenceName”, “mlModelRef” and “mlModelCoordinationGroupRef”. According to the use case description and requirements that drove their definition, these attributes:
* can be configured by the consumer. This means that “isWritable” property should be “T”. However, currently “isWritable” = “F”.
* cannot be modified after the creation of the MLTrainingRequest instance. This means that “isInvariant” property should be “T”. However, currently “isInvariant” = “F”.

**Third Issue**: **“mlTrainingType” attribute**The “mLTrainingType” attribute is currently defined in three IOCs: MLModel, MLTrainingFunction and MLTrainingRequest. However, it only needs to be present in the MLModel. On the one hand, “mLTrainingType” is a property associated to an ML model, but not to the training function – therefore, this attribute cannot represent a discoverable property of MLTrainingFunction.The reason is that the training type that can be invoked over an ML model depends on the model internals (e.g. architecture), not on the logic/implementation of the training function as such.On the other hand, the current definition of MLTrainingRequest does not require “mLTrainingType” attribute, as the information conveyed by this attribute is already specified with either “mLModelRef” (the MLModel instance already has “mlTrainingType”) and “aIMLInferenceName” (used for pre-specialized training, fine-tuning and initial training). Including “mlTrainingType” together with any of these attributes will cause redundancies and may cause inconsistencies. |
|  |  |
| ***Summary of change:*** | First Issue:* Remove “expectedInferenceScope”.
* Update “aimlInferenceName” in the TrainingRequest IOC, updating attribute cardinality to 0..\* and clarifying its usage for different training types.
* Update constraints and properties of impacted attributes.

Second Issue:* Remove the restriction that prevents having in the system a MLModel instance representing an initial ML model.
* Make “MLTrainingRequest.mlModelRef” attribute M.

Third Issue:* Remove “mlTrainingType” attribute from MLTrainingFunction and MLTrainingRequest.
* Update “mlTrainingType” attribute definition in the big table, to reflect on the above changes.
 |
|  |  |
| ***Consequences if not approved:*** | Wrong implementation of training functionality which wrong/uncertain behaviour on system/MnS producer when processing consumer-initiated training requests.Limitations on the initial training use case for operators. |
|  |  |
| ***Clauses affected:*** | 7.3a.1.2.1.2, 7.3a.1.2.2.1, 7.3a.1.2.2.2, 7.3a.1.2.2.3, 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:*** |  |

|  |
| --- |
| **1st Change** |

##### 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 |
| mLKnowledge | O | T | F | F | T |
|  |  |  |  |  |  |
| **Attribute related to role** |  |  |  |  |  |
| mLModelRepositoryRef | M | T | F | F | T |

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

None.

###### 7.3a.1.2.1.4 Notifications

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

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

7.3a.1.2.2 MLTrainingRequest

7.3a.1.2.2.1 Definition

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

To trigger the ML model training process, ML training MnS consumer needs to create MLTrainingRequestinstances 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 instance represents the request for one of the following training types:: (1) initial training, where the MnS consumer requests to train an initial version of 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. For re-training, the MLTrainingRequest is always associated to one MLModel or one MLModelCoordinationGroup. For fine-tuning, the MLTrainingRequest is always associated to one MLModel representing a pre-specialised ML model. The aIMLInferenceName attribute specifies the inference type(s) This attribute can be used by the consumer to request for initial training, pre-specialized training and fine-tuning. For pre-specialized training, aIMLInferenceName attribute can have two or more values. For fine-tuning, aIMLInferenceName attribute has one single value that is specified together with mlModelRef,to adapt the inference scope of that pre-specialized ML model.

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.

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 | T | T | 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 |
| 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 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| clusteringInfo | O | T | T | F | T |
| **Attribute related to role** |  |  |  |  |  |
| mLModelRef | M | T | T | T | T |
| mLModelCoordinationGroupRef | CM | T | T | T | T |

7.3a.1.2.2.3 Attribute constraints

**Table 7.3a.1.2.2.3-1**

|  |  |
| --- | --- |
| **Name** | **Definition** |
| aIMLInferenceName | Condition: Any of the following training types are supported: Consumer-initiated Initial training, pre-specialized training, fine-tuning. |
|  |  |
| mLModelCoordinationGroupRef | Condition: ML model joint training is supported. |
| mLKnowledgeName | Condition: ML-knowledge-based transfer learning is supported.  |
| rLRequirement | Condition: Reinforcement learning is supported |
|  |  |

7.3a.1.2.2.4 Notifications

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

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

## 7.5 Attribute definitions

### 7.5.1 Attribute properties

| **Attribute Name** | **Documentation and Allowed Values** | **Properties** |
| --- | --- | --- |
| mLModelId | It identifies the ML model.It is unique in each MnS producer.allowedValues: N/A. | type: Stringmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: 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: Stringmultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| aIMLInferenceName | It indicates the type of inference that the ML model supports. allowedValues: see clause 7.4.10 | type: AIMLInferenceNamemultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| MLTrainingRequest.aIMLInferenceName | It indicates the type(s) of inference that the ML model is trained for. For initial training, re-traning and fine-tuning, one single type of inference is supported.For pre-specialized training, two or more types of inference are supported.allowedValues: see clause 7.4.10 | type: AIMLInferenceNamemultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: 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: 1isOrdered: N/AisUnique: N/AdefaultValue: 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: 1isOrdered: N/AisUnique: N/AdefaultValue: 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: NgRanInferenceTypemultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| vSExtensionType | It indicates the type of inference that is vendor's specific extension.allowedValues: N/A. | type: Stringmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: 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. It may include the information about the effectiveness of training data, which indicates the consumer-provided training data is useful or not.allowedValues: N/A. | type: Stringmultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| trainingRequestRef | It is the DN(s) of the related MLTrainingRequest MOI(s). | type: DN multiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| trainingProcessRef | It is the DN(s) of the related MLTrainingProcess MOI(s) that produced the MLTrainingReport. | type: DN multiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: 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..1isOrdered: N/AisUnique: N/AdefaultValue: 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..1isOrdered: N/AisUnique: N/AdefaultValue: 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: Integermultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: 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: 1isOrdered: N/AisUnique: N/AdefaultValue: 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: Enummultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| mLTrainingProcessId | It identifies the training process.It is unique in each instantiated process in the MnS producer.allowedValues: N/A. | type: Stringmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: 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: Integermultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: 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: Stringmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| progressStatus | It indicates the status of the process.allowedValues: N/A. | type: ProcessMonitor multiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: 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: Booleanmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: 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: Booleanmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: False |
| mLModelVersion | It indicates the version number of the ML model.allowedValues: N/A. | type: Stringmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| performanceRequirements | It indicates the expected performance for a trained ML model when performing on the training data.allowedValues: N/A. | type: ModelPerformancemultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| modelPerformanceTraining | It indicates the performance score of the ML model when performing on the training data.allowedValues: N/A. | type: ModelPerformancemultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: 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 trainedThe allowed values for " mLTrainingProcess.progressStatus.status " = "CANCELLING" are vendor specific.The allowed values for " mLTrainingProcess.progressStatus.status " = "NOT\_STARTED" are vendor specific. | type: Stringmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: 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: Stringmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: 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: Stringmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: 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: Realmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: 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: Booleanmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: 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: Booleanmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: 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: Booleanmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: 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: Booleanmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: False |
| inferenceEntityRef | It describes the target entities that will use the ML model for inference. | type: DN multiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: 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: FalseisUnique: TruedefaultValue: None isNullable: False |
| areNewTrainingDataUsed | It indicates whether new training data are used for the ML model training.allowedValues: TRUE, FALSE. | type: Booleanmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: 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: Realmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: 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: Realmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| expectedRuntimeContext | This describes the context where an MLModel is expected to be applied.allowedValues: N/A | type: MLContextmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| trainingContext | This specifies the context under which the MLModel has been trained.allowedValues: N/A | type: MLContextmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| runTimeContext | This specifies the context where the MLmodel or model is being applied.allowedValues: N/A | type: MLContextmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| MLTrainingRequest.mLModelRef | It identifies the DN of the MLModel requested to be trained. | type: DNmultiplicity: 0..1isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| MLTrainingReport.mLModelGeneratedRef | It identifies the DN of the MLModel generated by the ML model training. | type: DNmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| mLModelRepositoryRef | It identifies the DN of the MLModelRepository. | type: DNmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| mLRepositoryId | It indicates the unique ID of the ML repository. | type: Stringmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| modelPerformanceValidation | It indicates the performance score of the ML model when performing on the validation data.allowedValues: N/A | type: ModelPerformancemultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: 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: Integermultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: 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: Enummultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: 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: Booleanmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: 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: Booleanmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: False |
| MLTestingRequest.mLModelRef | It identifies the DN of the MLModel requested to be tested. | type: DNMultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| modelPerformanceTesting | It indicates the performance score of the ML model when performing on the testing data.allowedValues: N/A. | type: ModelPerformancemultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: 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: Stringmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| testingRequestRef | It identifies the DN of the MLTestingRequest MOI. | type: DNmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| supportedPerformanceIndicators | This parameter lists specific PerformanceIndicator(s) of an ML model.allowedValues: N/A. | type: SupportedPerfIndicator multiplicity: 1..\*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| performanceIndicatorName | It indicates the identifier of the specific performance indicator.allowedValues: N/A | type: Stringmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: 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: Booleanmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: 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: Booleanmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: False |
| mLUpdateProcessRef | It is the DN of the mLUpdateProcess MOI that represents the process of updating an ML model. | type: DNmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| mLUpdateRequestRefList | It is the list of DN of the MLUpdateRequest MOI that represents an ML update request. | type: DNmultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| mLUpdateReportRef | It is the DN of the MLUpdateReport MOI that represents an ML update report. | type: DNmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| mLUpdateReportingPeriod | It specifies the time duration upon which the MnS consumer expects the ML update is reported. | type: TimeWindowmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| availMLCapabilityReport | It represents the available ML capabilities.allowedValues: N/A. | type: AvailMLCapabilityReport multiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| updatedMLCapability | It represents the updated ML capabilities.allowedValues: N/A. | type: AvailMLCapabilityReport multiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| availMLCapabilityReportID | It identifies the available ML capability report.allowedValues: N/A. | type: Stringmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: 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: Stringmultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| mlCapabilityVersionId | It indicates the version of ML capabilities that is available for the update.  | type: Stringmultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: 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: ModelPerformancemultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: 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: ModelPerformancemultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: NoneisNullable: False |
| updateTimeDeadline | It indicates the maximum as stated in the MLUpdate request that should be taken to complete the update | type: TimeWindowmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| MLUpdateReport.mLModelRefList | It indicates the DN of MLModel instances that can be updated. | type: DNmultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: NoneisNullable: False |
| MLUpdateRequest.requestStatus | It describes the status of a particular ML update request.allowedValues: NOT\_STARTED, IN\_PROGRESS, CANCELLING, SUSPENDED, FINISHED, and CANCELLED. | type: Enummultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: 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: Booleanmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: 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: Booleanmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: False |
| memberMLModelRefList | It identifies the list of member ML models within an ML model coordination group. | type: DNmultiplicity: 2..\*isOrdered: TrueisUnique: TruedefaultValue: None isNullable: False |
| MLTrainingRequest.mLModelCoordinationGroupRef | It identifies the DN of the MLModelCoordinationGroup requested to be trained. | type: DNmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| MLTrainingReport.mLModelCoordinationGroupGeneratedRef | It identifies the DN of the MLModelCoordinationGroup generated by ML model joint training. | type: DNmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| MLTestingRequest.mLModelCoordinationGroupRef | It identifies the DN of the MLModelCoordinationGroup requested to be tested. | type: DNmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: 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: DNmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: 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: Enummultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: 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: Booleanmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: 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: Booleanmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: False |
| mLModelToLoadRef | It identifies the DN of a trained MLModel requested to be loaded to the target inference function(s). | type: DNmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: 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: AIMLManagementPolicymultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| thresholdList | It provides the list of threshold.  | type: ThresholdInfomultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: 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: Stringmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: 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: Booleanmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: 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: Booleanmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: False |
| mLModelLoadingRequestRef | It identifies the DN of the associated MLModelLoadingRequest. | type: DNmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| mLModelLoadingPolicyRef | It identifies the DN of the associated MLModelLoadingPolicyRef. | type: DNmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| loadedMLModelRef | It identifies the DN of the MLModel that has been loaded to the inference function.  | type: DNmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| activationStatus | It describes the activation status.allowedValues: ACTIVATED, DEACTIVATED. | type: Enummultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: 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: ManagedActivationScopemultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: 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: AIMLManagementPolicymultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: 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: DNmultiplicity: \*isOrdered: TrueisUnique: TruedefaultValue: 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: TimeWindowmultiplicity: \*isOrdered: TrueisUnique: TruedefaultValue: 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: GeoAreamultiplicity: \*isOrdered: TrueisUnique: TruedefaultValue: None isNullable: False |
| usedByFunctionRefList | It provides the DNs of the functions supported by the AIMLInferenceFunction.allowedValues: N/A | type: DNmultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| inferenceOutputId  | It identifies an inference output within an AIMLinferenceReport. | type: Stringmultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: 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: InferenceOutputmultiplicity: 1..\*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| inferencePerformance | It indicates the performance score of the ML model during Inference.allowedValues: N/A. | type: ModelPerformancemultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| inferenceOutputTime | It indicates the time at which the inference output is generated.allowedValues: N/A | type: DateTimemultiplicity: \*isOrdered: TrueisUnique: TruedefaultValue: None isNullable: False |
| outputResult | It indicates the result of an inference. | type: AttributeValuePairmultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: NullisNullable: False |
| mLCapabilitiesInfoList | It indicates information about what an ML model can generate inference for. allowedValues: N/A. | type: MLCapabilityInfomultiplicity: 1..\*isOrdered: FalseisUnique: TruedefaultValue: 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: Stringmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| mLCapabilityParameters | It indicates a set of optional parameters that apply for an aIMLInferenceName capabilityName. allowedValues: N/A | type: AttributeValuePair multiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| aIMLInferenceReportRefList | It indicates a list of DN of the AIMLInferenceReport MOI that represents an AIML inference report. | type: DNmultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| mLModelRefList | It identifies the list of MLModel DN. | type: DNmultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| mLKnowledge | It indicates an instance of ML Knowledge available at the ML training function. | type: MmLKnowledgemultiplicity: \*isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| mLKnowledgeName | It identifies the ML Knowledge.It is unique in each MnS producer. | type: Stringmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| KnowledgeType | It identifies the type of ML Knowledge as either aStatistic, a regression or a Table of input-output value(s)Allowed values: TABLE , STATISTIC, REGRESSION | type: ENUMmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| PredictorResponseArray | It identifies the predictor and corresponding response data for a piece of ML Knowledge. For exapme, it represents one of the following:- the input and output data for a table - the predictor and response for a statistic, - the input and output data for a regressionNOTE: The nature of the data is not scope of this specification | type: pair<String, String>multiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| inferenceExplanationInfo | It indicates the inference explanation information of the ML model Inference results. E.g. the critical features in the training or inference data. | type: Stringmultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| mLTrainingType | It indicates the type of training that can be applied to an ML modelallowed values: initial training, pre-specialised training, re-training, fine-tuning. | type: Enummultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
|  |  |  |
| inferenceScope | It indicates the inference capabilities that the ML model after pre-specialized training can be fine-tuned to support, where the inference scope contains a list of aIMLInferenceName that the ML model can be potentially adapted to support. | type: AIMLInferenceNamemultiplicity: \*isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| distributedTrainingExpectation | It indicates distributed traning expectations provided by MnS consumer.allowedValues: N/A. | type: DistributedTrainingExpectationmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| expectedTrainingTime | It indicates the expected training duration provided by MnS consumer, in unit of minites.allowedValues: Integer | type: Integermultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| dataSplitIndication | This is a Boolean attribute specifying whether the provided training data should be split or not. The value FALSE specify that the training data shall not be spilt.allowedValues: TRUE, FALSE. | type: Booleanmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: False isNullable: False |
| suggestedTrainingNodeList | It indicates a list of suggested training nodes provided by MnS consumer.allowedValues: Not applicable. | type: DNmultiplicity: 0..\*isOrdered: N/AisUnique: TruedefaultValue: NoneisNullable: False |
| trainingDataStatisticalProperties | It indicates the training data statistical properties to be considered by the MnS producer when training an ML model. | type: DataStatisticalPropertiesmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| 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: Booleanmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: False |
| 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: Booleanmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: False |
| potentialImpactInfo | This datatype define the potential network impacts due to the inference output results | type: PotentialImpactInfomultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| impactedScope | This will specify the scope of affect, the inference output may have on the network including entities performing the recommended actions in the inference output and entities impacted due to implementation of the recommended actionsThe choice attribuite dNList defines Identifier of the network functions that may be affected by the output result of the inference function.The choice attribute timeWindow defines a time duration indicating that the related network function(s) may be affected during this time duration by the inference output result.The choice attribute geoPolygon defines a Geographical location indicating that the network function(s) in that location may be affected by the inference output result. | type: ManagedActivationScopemultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| impactedPM | This will identify the potential performance metrics that may be degraded/improved due to the implementation of recommendations provided as part of inference output. | type: ImpactedPMmultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| pMIdentifier | This indicates the performance measurement or the KPI that may be impacted by the ML Model. This will be the name of PM and KPI as defined in 3GPP TS 28.552 and 28.554 respectively (e.g. for Managing NG-RAN AI/ML-based distributed Load Balancing function, the PM can be measurements related to MLB, UE throughput and Radio resource utilization etc). | type: Stringmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| supportedLearningTechnology | It identifies the learning technologies including Reinforcement Learning, Federated Learning and Distributed training which supported by the ML training function. | type: SupportedLearningTechnologymultiplicity: 1isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| rLRequirement | It identifies the expected performanc and performed scope for the ML model training when Reinforcement Learning is supported. | type: RLRequirementmultiplicity: 1isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| learningTechnologyName | It indicates a list of learning technology names used to represent the learning technics supported by the ML training function.allowedValues: RL, FL, DLwhere RL indicates of Reinforcement Learning, FL indicates of Federated Learning and DL indicates of Distributed training. | type: Enummultiplicity: 1..\*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| supportedEnvironment | It indicates the supported RL environments. When the ML training MnS producer supports RL, this attribute is included in the SupportedLearningTechnology datatype, which indicates the supported environment of the ML training function for ML model training.allowedValues: SIMULATION ENVIONMENTS, REAL NETWORK ENVIONMENTS. | type: Enummultiplicity: 1..\*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| supportedInferenceNameList | It indicates a list of inference name that the learning technologies can be applied. allowedValues: see clause 7.4.10 | type: AIMLInferenceNamemultiplicity: 1..\*isOrdered: FalseisUnique: N/AdefaultValue: None isNullable: False |
| rLEnvironmentType | It indicates the simulated environment or real network where the ML model should be traind.allowedValues: SIMULATION ENVIONMENTS, REAL NETWORK ENVIONMENTS | type: Enummultiplicity: 0..\*isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| rLEnvironmentScope | It indicates the specific environment scope for the entities that the RL process should be performed, i.e, where the RL agent is located. | type: EnvironmentScopemultiplicity: 1..\*isOrdered: FalseisUnique: N/AdefaultValue: None isNullable: False |
| rLImpactedScope | It indicates the specific environment scope for the entities that may be impacted by the RL process, i.e., scope may be impacted by actions of the RL agent. | type: EnvironmentScopemultiplicity: 1..\*isOrdered: FalseisUnique: N/AdefaultValue: None isNullable: False |
| rLPerformanceRequirements | It indicates a list of thresholds for the network performance requirements, when the RL training process(es) is performed. | type: ThresholdInfomultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| clusteringInfo | It containes information that indicates the clustering criteria for the ML Models that can be grouped together for training | type: ClusteringCriteriamultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| ClusteringCriteria.performanceMetric | This defines clustering criteria based on the performance metric for which the ML model is mainly evaluated. That is, the models, which intend to achieve same performance characteristic (e.g. accuracy, precision, F1 score etc) can be clustered together for training. It indicates the performance metric used to evaluate the performance of an ML modelallowedValues: N/A | type: Stringmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| taskType | This defines grouping criteria based on the task the ML model is trained for. For example, this can be aIMLInferenceName or capabilityName as defined in 3GPP TS 28.105.Note: Whether the taskType can be aIMLInferenceName here is FFS. | type: Stringmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| allowedClusterTrainingTime | This defines the combined time limit within which the training of ML models cluster shall be completed. A cluster of ML models takes more time to train together as compared to time taken for training an individual ML model. The criteria allows accommodating only those ML models whose training time does not exceed the set combined time limit | type: TimeWindowmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: True |
| preferredModelDiversity | This defines the consumer preferred model diversity types that is to be considered for models clustering. For example, decision trees, neural networks, linear regression and like so | type: Stringmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| clusteringInfo | It containes information that indicates the clustering criteria for the ML Models that can be grouped together for training | type: ClusteringCriteriamultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: 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. |

|  |
| --- |
| **End of Changes** |