**3GPP TSG-SA5 Meeting #162** ***S5-253845***

**Gothenburg, Sweden, 25 – 29 August 2025**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.3* | | | | | | | | |
| **CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  | **28.105** | **CR** | **Input to draftCR** | **rev** | **1** | **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 Rel-19 TS 28.105 Use case, requirements and solution on model confidence requirement | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** | Nokia, Nokia Shanghai Bell | | | | | | | | | |
| ***Source to TSG:*** | SA5 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | AIML\_MGT\_Ph2 | | | | |  | ***Date:*** | | | 2025-05-08 |
|  |  | | | |  | |  | | |  |
| ***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-17 (Release 17) Rel-18 (Release 18) Rel-19 (Release 19)  Rel-20 (Release 20)* | |
|  |  | | | | | | | | | |
| ***Reason for change:*** | | Use case, requirement and solution for supporting the indication of ML model confidence by the ML training MnS consumer to the ML training MnS producer. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | 1. Introduce a new modelConfidenceRequirement attribute in the MLTrainingRequest IoC. 2. Description of the modelConfidenceRequirement attribute in the attribute table. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | Cannot support the indication of ML model confidence requirement to the ML training MNS producer. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | 7.3a.1.2.2, 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:*** | |  | | | | | | | | |

***Start of First change***

6 AI/ML management use cases and requirements

## 6.2b ML model training

### 6.2b.1 Description

Before an ML model is deployed to conduct inference, the ML model algoritm associated with the ML model needs to be trained. The ML model training can be an initial training or the re-training of an already trained ML model.

The ML model is trained by the ML training MnS producer, and the training can be triggered by request(s) from one or more ML training MnS consumer(s), or initiated by the ML training MnS producer (e.g., as a result of model performance evaluation).

The procedures in [X1] for authentication and authorization are applicable for all ML model training use cases.

6.2b.2 Use cases

6.2b.2.X Minimum confidence requirement for ML model training

Confidence is a score, which may be expressed as a percentage (e.g., 0% to 100%). It indicates the model's certainty about the predictions based on the data it has learned from.

High Confidence (e.g., 95%): The model is very certain about its prediction.

Low Confidence (e.g., 30%): The model is uncertain about its prediction.

This confidence score can be valuable. For instance, consumer may decide to only act on the model's predictions if the confidence level is above a certain threshold (e.g., only take preventative action if the overload prediction confidence is higher than 80%).In the ML training step, the ML training function may support the capability to evaluate the confidence value of the trained ML model. The ML model confidence value indicates the average confidence value that the ML model would perform for data with the same distribution as the training data. Therefore, the MnS consumer may prefer to indicate the minimum confidence requirement that the trained ML model should satisfy before the ML model is deployed. The ML training MnS producer should train the ML model to meet the minimum confidence requirement provided by the consumer. If the producer cannot reach the required minimum value, this should be reported to the consumer. This attribute for confidence is optional.

The way of using the confidence score in training is implementation specific, and out of scope of this specification.

***Start of Next change***

6.2b.3 Requirements for ML model training

**Table 6.2b.3-1**

| **Requirement label** | **Description** | **Related use case(s)** |
| --- | --- | --- |
| **REQ-ML\_TRAIN-FUN-01** | The ML training MnS producer shall have a capability allowing an authorized ML training MnS consumer to request ML model training. | ML model training requested by consumer (clause 6.2b.2.1) |
| **REQ- ML\_TRAIN-FUN-02** | The ML training MnS producer shall have a capability allowing the authorized ML training MnS consumer to specify the data sources containing the candidate training data for ML model training. | ML model training requested by consumer (clause 6.2b.2.1) |
| **REQ- ML\_TRAIN-FUN-03** | The ML training MnS producer shall have a capability allowing the authorized ML training MnS consumer to specify the AI/ML inference name of the ML model to be trained. | ML model training requested by consumer (clause 6.2b.2.1) |
| **REQ- ML\_TRAIN-FUN-04** | The ML training MnS producer shall have a capability to provide the training result to the ML training MnS consumer. | ML model training requested by consumer (clause 6.2b.2.1), ML model training initiated by producer (clause 6.2b.2.2) |
| **REQ- ML\_TRAIN-FUN-05** | The ML training MnS producer shall have a capability allowing an authorized ML training MnS consumer to configure the thresholds of the performance measurements and/or KPIs to trigger the re-training of an ML model. (See Note) | ML model training initiated by producer (clause 6.2b.2.2) |
| **REQ- ML\_TRAIN-FUN-06** | The ML training MnS producer shall have a capability to provide the version number of the ML model when it is generated by ML model re-training to the authorized ML training MnS consumer. | ML model training requested by consumer (clause 6.2b.2.1), ML model training initiated by producer (clause 6.2b.2.2) |
| **REQ- ML\_TRAIN-FUN-07** | The ML training MnS producer shall have a capability allowing an authorized ML training MnS consumer to manage the training process, including starting, suspending, or resuming the training process, and configuring the ML context for ML model training. | ML model training requested by consumer (clause 6.2b.2.1), ML model training initiated by producer (clause 6.2b.2.2), ML model joint training (clause 6.2b.2.6) |
| **REQ- ML\_TRAIN-FUN-08** | The ML training MnS producer should have a capability to provide the grouping of ML models to an authorized ML training MnS consumer to enable coordinated inference. | ML model joint training (clause 6.2b.1.2.6) |
| **REQ- ML\_TRAIN-FUN-09** | The ML training MnS producer should have a capability to allow an authorized ML training MnS consumer to request joint training of a group of ML models. | ML model joint training (clause 6.2b.2.6) |
| **REQ- ML\_TRAIN-FUN-10** | The ML training MnS producer should have a capability to jointly train a group of ML models and provide the training results to an authorized consumer. | ML model joint training (clause 6.2b.2.6) |
| **REQ-ML\_SELECT-01** | 3GPP management system shall have a capability to enable an authorized ML training MnS consumer to discover the properties of available ML models including the contexts under which each of the models were trained. | ML model and ML model selection (clause 6.2b.2.3) |
| **REQ-ML\_SELECT-02** | 3GPP management system shall have a capability to enable an authorized ML training MnS consumer to select an ML model to be used for inference. | ML models and ML model selection (clause 6.2b.2.3) |
| **REQ-ML\_SELECT-03** | 3GPP management system shall have a capability to enable an authorized ML training MnS consumer to request for information and be informed about the available alternative ML models of differing complexity and performance. | ML model and ML model selection (clause 6.2b.2.3) |
| **REQ-ML\_SELECT-04** | The 3GPP management system shall have a capability to provide a selected ML model to the authorized ML training MnS consumer. | ML model and ML model selection (clause 6.2b.2.3) |
| **REQ-ML\_TRAIN- MGT-01** | The ML training MnS producer shall have a capability allowing an authorized consumer to manage and configure one or more requests for the specific ML model training, e.g. to modify the request or to delete the request. | ML model training requested by consumer (clause 6.2b.2.1), Managing ML model Training Processes (clause 6.2b.2.4) |
| **REQ-ML\_TRAIN- MGT-02** | The ML training MnS producer shall have a capability allowing an authorized ML training MnS consumer to manage and configure one or more training processes, e.g. to start, suspend or restart the training. | ML model training requested by consumer (clause 6.2b.2.1),  Managing ML model training processes (clause 6.2b.2.4) |
| **REQ-ML\_TRAIN- MGT-03** | 3GPP management system shall have a capability to enable an authorized ML training MnS consumer (e.g. the function/model different from the function that generated a request for ML model training) to request for a report on the outcomes of a specific training instance. | Managing ML model training processes (clause 6.2b.2.4) |
| **REQ-ML\_TRAIN- MGT-04** | 3GPP management system shall have a capability to enable an authorized ML training MnS consumer to define the reporting characteristics related to a specific training request or training instance. | Managing ML model training processes (clause 6.2b.2.4) |
| **REQ-ML\_TRAIN- MGT-05** | 3GPP management system shall have a capability to enable the ML training function to report to any authorized ML training MnS consumer about specific ML model training process and/or report about the outcomes of any such ML model training process. | Managing ML model training processes (clause 6.2b.2.4) |
| **REQ-ML\_ERROR-01** | The 3GPP management system shall enable an authorized consumer of data services (e.g. an ML training function) to request from a producer of data services a Value Quality Score of the data, which is the numerical value that represents the dependability/quality of a given observation and measurement type. | Handling errors in data and ML decisions (clause 6.2b.2.5) |
| **REQ-ML\_ERROR-02** | The 3GPP management system shall enable an authorized consumer of AI/ML decisions (e.g. a controller) to request ML decision confidence score which is the numerical value that represents the dependability/quality of a given decision generated by an AI/ML inference function. | Handling errors in data and ML decisions (clause 6.2b.2.5) |
| **REQ-ML\_ERROR-03** | The 3GPP management system shall have a capability to enable an authorized consumer to provide to the ML Training MnS producer, a training data quality score, which is the numerical value that represents the dependability/quality of a given observation and measurement type. | Handling errors in data and ML decisions (clause 6.2b.2.5) |
| **REQ-ML\_ERROR-04** | The 3GPP management system shall enable a producer of ML decisions (e.g. an AI/ML inference function) to provide to an authorized consumer of ML decisions (e.g. a controller) an AI/ML decision confidence score which is the numerical value that represents the dependability/quality of a given decision generated by the AI/ML inference function. | Handling errors in data and ML decisions (clause 6.2b.2.5) |
| **REQ-ML\_VLD-01** | The ML training MnS producer should have a capability to validate the ML models during the ML model training process and report the performance of the ML models on both the training data and validation data to the authorized consumer. | ML model validation performance reporting (clause 6.2b.2.7) |
| **REQ-ML\_VLD-02** | The ML training MnS producer should have a capability to report the ratio (in terms of quantity of data samples) of the training data and validation data used during the ML model training and validation process. | ML model validation performance reporting (clause 6.2b.2.7) |
| **REQ-TRAIN\_EFF-01** | The 3GPP management system should have the capability to allow an authorized consumer to configure an ML training function to report the effectiveness of data used for model training. | Training data effectiveness reporting (clause 6.2b.2.8) |
| **REQ-ML\_TRAIN\_PM-1** | The ML Training MnS producer should have a capability to allow an authorized consumer to get the capabilities about what kind of ML models the ML training function is able to train. | Performance indicator selection for ML model training (clause 6.2b.2.9.2) |
| **REQ-ML\_TRAIN\_PM-2** | The ML Training MnS producer should have a capability to allow an authorized consumer to query what performance indicators are supported by the ML model training for each ML model. | ML model performance indicators query and selection for ML model training (clause 6.2b.2.9.3) |
| **REQ-ML\_TRAIN\_PM-3** | The ML Training MnS producer should have a capability to allow an authorized consumer to select the performance indicators from those supported by the ML training function for reporting the training performance for each ML model. | ML model performance indicators query and selection for ML model training (clause 6.2b.2.9.3) |
| **REQ-ML\_TRAIN\_PM-4** | The ML Training MnS producer should have a capability to allow an authorized consumer to provide the performance requirements for the ML model training using the selected the performance indicators from those supported by the ML training function. | MnS consumer policy-based selection of ML model performance indicators for ML model training (clause 6.2b.2.9.4) |
| **REQ-MLKTL-1** | The 3GPP management systemshould have a capability to enable an authorized MnS consumer to discover or request all or part of the available shareable knowledge at a given MLKTL MnS producer. | ML-Knowledge-based Transfer Learning (clause 6.2b.2.X1.2.1) |
| **REQ-MLKTL-2** | The 3GPP management systemshould have a capability for an MLKTL MnS producer to provide to an authorized MnS consumer all or part of its available shareable knowledge | ML-Knowledge-based Transfer Learning (clause 6.2b.2.X1.2.1) |
| **REQ-MLKTL-3** | The 3GPP management systemshould have a capability enabling an authorized MnS consumer to request a MLKTL MnS producer to initiate and execute a transfer learning instance to a specified ML model or ML-enabled function | ML-Knowledge-based Transfer Learning (clause 6.2b.2.X1.2.2) |
| **REQ-MLKTL-4** | The 3GPP management systemshould have a capability to enable an authorized MnS consumer to manage or control the knowledge request or the knowledge process or transfer learning process, e.g. to suspend, re-activate or cancel the ML Knowledge Request; or to adjust the description of the desired knowledge  NOTE: An example MnS consumers include an operator or the function that generated the request for available Knowledge | ML-Knowledge-based Transfer Learning (clause 6.2b.2.X1.2.2) |
| **REQ-MLKTL-5** | The 3GPP management systemshould have a capability to enable an ML model or ML training function to register available knowledge to a shared knowledge repository, e.g. through a ML Knowledge Registration process | ML-Knowledge-based Transfer Learning (clause 6.2b.2.X1.2.2) |
| **REQ-MLKTL-6** | The 3GPP management systemshould have a capability enabling an authorized MnS consumer to request the Knowledge Repository to provide some or all the knowledge available for sharing based on specific criteria | ML-Knowledge-based Transfer Learning (clause 6.2b.2.X1.2.2) |
| **REQ-ML\_TRAIN\_CLUSTER-01** | The ML Training MnS producer should have a capability for an authorized MnS consumer to request training of a cluster of ML models as per clustering criteria associated to a set of multiple contexts from a previously trained ML model. | ML model training for multiple contexts (clause 6.2b.2.X2) |
| **REQ-ML\_TRAIN-PRE-01** | The ML training MnS producer should have a capability allowing an authorized ML training MnS consumer to request pre-specialized training of a ML model. | ML Pre-specialised training (clause 6.2b.2.X3) |
| **REQ- ML\_TRAIN-FT-x1** | The ML training MnS producer should have a capability to enable an authorised consumer to request the fine-tuning of a pre-specialised trained ML model. | ML Pre-specialised training (clause 6.2b.2.X3), ML Fine-Tuning (clause 6.2b.2.X4) |
| **REQ- ML\_TRAIN-FT-x2** | The ML training MnS producer should have a capability allowing the consumer to specify the training type of an ML model training request such as pre-specialised training, and fine-tuning. | ML Pre-specialised training (clause 6.2b.2.X3), ML Fine-Tuning (clause 6.2b.2.X4) |
| **REQ-ML\_DIST-TRNG-01** | The ML training MnS producer should have a capability allowing and authorized consumer to provide distributed training requirements to the MnS Producer. | Management of Distributed training (clause 6.2b.2.X5) |
| **REQ-ML\_TRAIN\_FL-1** | The ML training MnS producer should have a capability allowing an authorized consumer to discover the FL roles (FL server or FL client) in Federated Learning. | Management of different roles in Federated Learning (Clause 6.2b.2.X6.2.1) |
| **REQ-ML\_TRAIN\_FL-2** | The ML training MnS producer should have a capability allowing an authorized consumer to provide FL training requirements to the MnS Producer. | Management of different roles in Federated Learning (Clause 6.2b.2.X6.2.1) |
| **REQ-ML\_TRAIN\_FL-3** | The ML training MnS producer should have a capability allowing an authorized consumer to provide requirements for selecting (including adding and removing) FL clients in Federated Learning to the MnS Producer. | Management of different roles in Federated Learning (Clause 6.2b.2.X6.2.1) |
| **REQ-ML\_TRAIN\_FL-4** | The ML training MnS producer should have a capability allowing an authorized consumer to get the performance of the global ML model on each participating FL Client. | Management of different roles in Federated Learning (Clause 6.2b.2.X6.2.1) |
| **REQ-ML\_TRAIN\_FL-5** | The ML training MnS producer should have a capability to report the information about the contribution of each FL client to the FL process to MnS consumer. | Management of different roles in Federated Learning (Clause 6.2b.2.X6.2.1) |
| **REQ-ML\_TRAIN\_FL-6** | The ML training MnS producer should have a capability to report the candidate FL Clients for the FL process. | Management of different roles in Federated Learning (Clause 6.2b.2.X6.2.1) |
| **REQ-RL\_TRAIN\_01** | The ML training MnS producer should have a capability allowing an authorized MnS consumer to query if RL training is supported. | Enabling Reinforcement Learning (6.2b.2.X7.2.1) |
| **REQ-RL\_TRAIN\_02** | The ML Training MnS producer should have a capability to report RL types (i.e., online RL, offline RL) to an authorized consumer. | Enabling Reinforcement Learning (6.2b.2.X7.2.1) |
| **REQ-RL\_TRAIN\_03** | The ML Training MnS producer should have a capability to allow an authorized consumer to get the type and scope of the RL environment for which an RL model has been trained. | Exploration in Reinforcement Learning (6.2b.2.X7.2.2) |
| **REQ-RL\_TRAIN\_04** | The ML Training MnS producer should have a capability to allow an authorized consumer to select the type of the RL environment for which an RL model is to be trained. | Exploration in Reinforcement Learning (6.2b.2.X7.2.2) |
| **REQ-RL\_TRAIN\_05** | The ML Training MnS producer should have a capability to allow an authorized consumer to provide the scope of the RL environment for which an RL model is to be trained. | Exploration in Reinforcement Learning (6.2b.2.X7.2.2) |
| **REQ-RL\_TRAIN\_06** | The ML training MnS producer should have a capability allowing an authorized MnS consumer to provide network performance requirements of performing RL training. | Exploration in Reinforcement Learning (6.2b.2.X7.2.2) |
| **REQ-RL\_TRAIN\_07** | The ML training MnS producer should have a capability to allow an authorized MnS consumer to specify the configuration range that the RL agent is allowed to explore. | Exploration in Reinforcement Learning (6.2b.2.X7.2.2) |
| **REQ-RL\_TRAIN\_08** | The ML Training MnS producer should have a capability to allow an authorized consumer to provide the allowed scope for the entities to be impacted by the RL actions. | Exploration in Reinforcement Learning (6.2b.2.X7  .2) |
| **REQ-ML\_TRAIN\_DST-01** | The 3GPP management system should enable an authorized consumer to provide information on the training dataset distribution. | Training data statistics (clause 6.2b.2.X8) |
| **REQ-ML\_TRAIN\_DST-02** | The 3GPP management system should enable an authorized consumer to provide information on the usage of outliers in the training dataset. | Training data statistics (clause 6.2b.2.X8) |
| **REQ-ML\_TRAIN\_CONF-01** | The ML Training MnS producer should have a capability allowing an authorized consumer to indicate the minimum confidence value that the ML Training MnS producer should meet on the trained ML model. | Minimum confidence requirement for ML model training (clause 6.2b.2.X) |
| NOTE: The performance measurements and KPIs are specific to each type (i.e., the inference type that the ML model supports) of ML model. | | |

***Start of 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 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.

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 |
| modelConfidenceRequirement | O | 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 |
| 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 |
| 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.

***Start of 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: 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 |
| MLTrainingRequest.aIMLInferenceName | It indicates the type of inference that the ML model conducting inference.  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. 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: 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 | |
| mLKnowledge | It indicates an instance of ML Knowledge available at the ML training function. | type: MmLKnowledge  multiplicity: \*  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False | |
| mLKnowledgeName | It identifies the ML Knowledge.  It is unique in each MnS producer. | type: String  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False | |
| KnowledgeType | It identifies the type of ML Knowledge as either a  Statistic, a regression or a Table of input-output value(s)  Allowed values: TABLE , STATISTIC, REGRESSION | type: ENUM  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: 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 regression  NOTE: The nature of the data is not scope of this specification | type: pair<String, String>  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: 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: String  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False | |
| mLTrainingType | It indicates the type of ML training (e.g., initial-training, re-training, pre-specialised training, fine-tuning) requested by the consumer.  allowed values: initial training, pre-specialised training, re-training, fine-tuning. | type: Enum  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False | |
| expectedInferenceScope | It indicates the inference capabilities that the ML model is expected to support, where the inference scope contains a list of aIMLInferenceName that the ML model can be potential adapted to support. | type: AIMLInferenceName  multiplicity: \*  isOrdered: N/A  isUnique: N/A  defaultValue: 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: AIMLInferenceName  multiplicity: \*  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False | |
| distributedTrainingExpectation | It indicates distributed traning expectations provided by MnS consumer.  allowedValues: N/A. | type: DistributedTrainingExpectation  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False | |
| expectedTrainingTime | It indicates the expected training duration provided by MnS consumer, in unit of minites.  allowedValues: Integer | type: Integer  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: 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: Boolean  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: False  isNullable: False | |
| suggestedTrainingNodeList | It indicates a list of suggested training nodes provided by MnS consumer.  allowedValues: Not applicable. | type: DN  multiplicity: 0..\*  isOrdered: N/A  isUnique: True  defaultValue: None  isNullable: False | |
| trainingDataStatisticalProperties | It indicates the training data statistical properties to be considered by the MnS producer when training an ML model. | type: DataStatisticalProperties  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: 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: Boolean  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: FALSE  isNullable: 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: Boolean  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: FALSE  isNullable: False | |
| potentialImpactInfo | This datatype define the potential network impacts due to the inference output results | type: PotentialImpactInfo  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: 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 actions  The 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: ManagedActivationScope  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: 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: ImpactedPM  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: 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: String  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False | |
| supportedLearningTechnology | It identifies the learning technologies including Reinforcement Learning, Federated Learning and Distributed training which supported by the ML training function. | type: SupportedLearningTechnology  multiplicity: 1  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False | |
| rLRequirement | It identifies the expected performanc and performed scope for the ML model training when Reinforcement Learning is supported. | type: RLRequirement  multiplicity: 1  isOrdered: False  isUnique: True  defaultValue: 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, DL  where RL indicates of Reinforcement Learning, FL indicates of Federated Learning and DL indicates of Distributed training. | type: Enum  multiplicity: 1..\*  isOrdered: False  isUnique: True  defaultValue: 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: Enum  multiplicity: 1..\*  isOrdered: False  isUnique: True  defaultValue: 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: AIMLInferenceName  multiplicity: 1..\*  isOrdered: False  isUnique: N/A  defaultValue: 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: Enum  multiplicity: 0..\*  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: 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: EnvironmentScope  multiplicity: 1..\*  isOrdered: False  isUnique: N/A  defaultValue: 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: EnvironmentScope  multiplicity: 1..\*  isOrdered: False  isUnique: N/A  defaultValue: None  isNullable: False | |
| rLPerformanceRequirements | It indicates a list of thresholds for the network performance requirements, when the RL training process(es) is performed. | type: ThresholdInfo  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False | |
| clusteringInfo | It containes information that indicates the clustering criteria for the ML Models that can be grouped together for training | type: ClusteringCriteria  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: 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 model  allowedValues: N/A | type: String  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: 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: String  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: 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: TimeWindow  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: 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: String  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False | |
| modelConfidenceRequirement | It indicates the minimum confidence value that the MnS producer should meet when training an ML model.This is a measure of degree of the convergence of the trained ML model.  allowedValues: {0 ... 100} | type: Integer  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  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. | | |

***End of changes***