**3GPP TSG SA WG5 Meeting #156 S5-244879d1**

**Maastricht, The Netherlands 19 - 23 August 2024**

|  |
| --- |
| *CR-Form-v12.1* |
| **CHANGE REQUEST** |
|  |
|  | **28.105** | **CR** |  **0176** | **rev** |  | **Current version:** | **18.4.0** |  |
|  |
| *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:***  | Rel-18 CR TS 28.105 Corrections including editorial fixes  |
|  |  |
| ***Source to WG:*** | NEC, Intel, Ericsson |
| ***Source to TSG:*** | S5 |
|  |  |
| ***Work item code:*** |  |  | ***Date:*** | 2024-08-05 |
|  |  |  |  |  |
| ***Category:*** |  |  | ***Release:*** | Rel-18 |
|  | *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-15 (Release 15)Rel-16 (Release 16)Rel-17 (Release 17)Rel-18 (Release 18)* |
|  |  |
| ***Reason for change:*** | Some corrections were missed from previous Draft CR implementations. New corrections and editorials are also needed.  |
|  |  |
| ***Summary of change:*** | Correction of figure 6.2x.2.1-1: ML entity to ML modelCorrect figure 6.2x.2.1-1 caption: x to bCorrections of attributes in Table 7.3a.1.2.2.3-1Various editorials including insertion or removal of speces where required. Chnage attribute properties “Type” to “type” to align with the MnS template in TS32.160. |
|  |  |
| ***Consequences if not approved:*** | Unclear and confusing spec. |
|  |  |
| ***Clauses affected:*** | 6.2b.2.1, 6.2b.3, 6.2c.1, 6.2c.2.3, 6.2c.2.4.2, 6.2c.3, 6.5.2.3, 6.5.3.1, 7.3a.1.2.2.2, 7.3a.1.2.2.3, 7.3a.1.2.2.2, 7.3a.1.2.2.3, 7.5.1 |
|  |  |
|  | **Y** | **N** |  |  |
| ***Other specs*** |  |  |  Other core specifications  | TS/TR ... CR ...  |
| ***affected:*** |  |  |  Test specifications | TS/TR ... CR ...  |
| ***(show related CRs)*** |  |  |  O&M Specifications | TS/TR ... CR ...  |
|  |  |
| ***Other comments:*** |  |
|  |  |
| ***This CR's revision history:*** |  |

***Start of change***

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).

6.2b.2 Use cases

6.2b.2.1 ML model training requested by consumer



**Figure 6.bx.2.1-1: ML model training requested by ML training MnS consumer**

The ML model training may be triggered by the request(s) from one or more ML training MnS consumer(s). The consumer may be for example a network function, a management function, an operator, or another functional differentiation.

To trigger an initial ML model training, the MnS consumer needs to specify in the ML training request the inference type which indicates the function or purpose of the ML model, e.g. CoverageProblemAnalysis [see TS 28.104 [2]]. The ML training MnS producer can perform the initial training according to the designated inference type. To trigger an ML model re-training, the MnS consumer needs to specify in the ML training request the identifier of the ML model to be re-trained.

The consumer may provide the data source(s) that contain(s) the training data which are considered as inputs candidates for training. To obtain the valid training outcomes, consumers may also designate their requirements for model performance (e.g. accuracy, etc) in the training request.

The performance of the ML model depends on the degree of commonality between the distribution of the data used for training and the distribution of the data used for inference. As time progresses, the distribution of the input data used for inference might change as compared to the distribution of the data used for training. In such a scenario, the performance of the ML model degrades over time. The ML training MnS producer may re-train the ML model if the inference performance of the ML model falls below a certain threshold, which needs to be configurable by the MnS consumer.

Following the ML training request by the ML training MnS consumer, the ML training MnS producer provides a response to the consumer indicating whether the request was accepted.

If the request is accepted, the ML training MnS producer decides when to start the ML model training with consideration of the request(s) from the consumer(s). Once the training is decided, the producer performs the following:

- selects the training data, with consideration of the consumer provided candidate training data. Since the training data directly influences the algorithm and performance of the trained ML model, the ML training MnS producer may examine the consumer's provided training data and decide to select none, some or all of them. In addition, the ML training MnS producer may select some other training data that are available;

- trains the ML model using the selected training data;

- validate the trained model using validation set of the training data;

- provides the training results (including the identifier of the ML model generated from the initially trained ML model or the version number of the ML model associated with the re-trained model, training performance results, etc.) to the ML training MnS consumer(s).

***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. | Performance indicator selection for ML model training (clause 6.2b.2.9.2) |
| **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. | Performance indicator selection for ML model training (clause 6.2b.2.9.2) |
| **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. | Performance indicator selection for ML model training (clause 6.2b.2.9.2) |
| NOTE: The performance measurements and KPIs are specific to each type (i.e., the inference type that the ML model supports) of ML model. |

***Next change***

6.2c ML model testing

6.2c.1 Description

After the training and validation, the ML model needs to be tested to evaluate the performance of it when it conducts inference using testing data.

If the testing performance is not acceptable or does not meet the pre-defined requirements, the consumer may request the ML training producer to re-train the ML model with specific training data and/or performance requirements.

6.2c.2 Use cases

6.2c.2.1 Consumer-requested ML model testing

After receiving an ML training report about a trained ML model from the ML training MnS producer, the consumer may request the ML testing MnS producer to test the ML model before applying it to the target inference function.

The ML model testing is to conduct inference on the tested ML model using the testing data as inference inputs and produce the inference output for each testing dataset example.

The ML testing MnS producer may be the same as or different from the ML training MnS producer.

After completing the ML model testing, the ML testing MnS producer provides the testing report indicating the success or failure of the ML model testing to the consumer. For a successful ML model testing, the testing report contains the testing results, i.e., the inference output for each testing dataset example.

6.2c.2.2 Producer-initiated ML model testing

The ML model testing may also be initiated by the MnS producer, after the ML model is trained and validated. A consumer (e.g., an operator) may still need to define the policies (e.g., allowed time window, maximum number of testing iterations, etc.) for the testing of a given ML model. The consumer may pre-define performance requirements for the ML model testing and allow the MnS producer to decide on whether ML model re-training/validation need to be triggered. ML model re-training may be triggered by the testing MnS producer itself based on the performance requirements supplied by the MnS consumer.

6.2c.2.2 Producer-initiated ML model testing

The ML model testing may also be initiated by the MnS producer, after the ML model is trained and validated. A consumer (e.g., an operator) may still need to define the policies (e.g., allowed time window, maximum number of testing iterations, etc.) for the testing of a given ML model. The consumer may pre-define performance requirements for the ML model testing and allow the MnS producer to decide on whether ML model re-training/validation need to be triggered. ML model re-training may be triggered by the testing MnS producer itself based on the performance requirements supplied by the MnS consumer.

6.2c.2.3 Joint testing of multiple ML models

A group of ML models may work in a coordinated manner for complex use cases.

The group of ML models is generated by the ML training function. The group, including all contained ML models, needs to be tested. After the ML model testing of the group, the MnS producer provides the testing results to the consumer.

NOTE: This use case is about the ML models testing before deployment.

6.2c.2.4 Performance management for ML model testing

6.2c.2.4.1 Overview

During ML model testing, the performance of ML model needs to be evaluated on testing data. The performance is the degree to which the ML models fulfil the objectives for which they were trained. The related performance indicators need to be collected and analyzed.

***Next change***

6.2c.2.4.2 Performance indicator selection for ML model testing

The ML model testing function may support testing for a single or several ML model algorithms and may support the capability to evaluate each ML model by one or more performance indicators.

The MnS consumer may prefer to use some performance indicator(s) over others to evaluate one kind of ML model. The performance indicators for testing mainly include the following aspects:

- ML model testing performance indicators: performance indicators of the ML model itself, including but not limited to:

- Accuracy indicator,

- Precision indicator,

- Recall indicator,

- F1 score indicator,

- MSE (Mean Squared Error) indicator,

- MAE (Mean Absolute Error) indicator, and

- RMSE (Root Mean Square Error) indicator.

In a similar way as for training, the MnS producer for ML model testing needs to provide the name(s) of supported performance indicator(s) for the MnS consumer to query and select for ML model performance evaluation. The MnS consumer may also need to provide the performance requirements of the ML model using the selected performance indicators.

The MnS producer for ML model testing uses the selected performance indicators for evaluating ML model testing, and reports with the corresponding performance score in the ML testing report when testing is completed.

***Next change***

6.2c.3 Requirements for ML model testing

**Table 6.2c.3-1**

| **Requirement label** | **Description** | **Related use case(s)** |
| --- | --- | --- |
| **REQ-ML\_TEST-1** | The ML testing MnS producer shall have a capability to allow an authorized consumer to request the testing of a specific ML model.  | Consumer-requested ML model testing (clause 6.2c.2.1) |
| **REQ-ML\_TEST-2** | The ML testing MnS producer shall have a capability to trigger the testing of an ML model and allow the MnS consumer to set the policy for the testing. | Producer-initiated ML model testing (6.2c.2.2) |
| **REQ-ML\_TEST-3** | The ML testing MnS producer shall have a capability to report the performance of the ML model when it performs inference on the testing data. | Consumer-requested ML model testing (clause 6.2c.2.1), and Producer-initiated ML model testing (clause 6.2c.2.2) |
| **REQ-ML\_TEST-4** | The ML testing MnS producer shall have a capability allowing an authorized consumer to request the testing of a group of ML models. | Joint testing of multiple ML models (clause 6.2c.2.3) |
| **REQ-ML\_TEST\_PM-1** | The ML testing MnS producer should have a capability to allow an authorized consumer to get the capabilities about what kind of ML models the ML testing function is able to test. | Performance indicator selection for ML model testing (clause 6.2c.2.4.2) |
| **REQ-ML\_TEST\_PM-2** | The ML testing MnS producer should have a capability to allow an authorized consumer to query what performance indicators are supported by the ML testing function for each ML model. | Performance indicator selection for ML model testing (clause 6.2c.2.4.2) |
| **REQ-ML\_TEST\_PM-3** | The ML testing MnS producer should have a capability to allow an authorized consumer to select the performance indicators from those supported by the ML testing function for reporting the testing performance for each ML model. | Performance indicator selection for ML tra (clause 6.2c.2.4.2) |

***Next change***

6.5.2.3 Requirements for AIML update control

**Table 6.5.2.3-1**

| **Requirement label** | **Description** | **Related use case(s)** |
| --- | --- | --- |
| **REQ-AIML\_UPDATE-1** | The AI/ML Inference MnS producer should have a capability to inform an authorized MnS consumer of the availability of AI/ML capabilities or ML models or versions thereof (e.g., as learned through a training process or as provided via a software update) and the readiness to update the AI/ML capabilities of the respective network function when triggered | Availability of new capabilities or ML models (clause 6.5.2.2.1) |
| **REQ-AIML\_UPDATE-2** | The AI/ML Inference MnS producer should have a capability to inform an authorized MnS consumer of the expected performance gain if/when the AI/ML capabilities or ML models of the respective network function are updated with/to the specific set of newly available AI/ML capabilities | Availability of new capabilities or ML models (clause 6.5.2.2.1) |
| **REQ-AIML\_UPDATE-3** | The AI/ML Inference MnS producer should have a capability to allow an authorized MnS consumer to request the AI/ML MnS producer to update its ML models using a specific version of newly available AI/ML capabilities or ML models or using AI/ML capabilities or ML models with requirements (e.g., the minimum achievable performance after updating, the maximum time taken to complete the update, etc) | Triggering ML model update (clause 6.5.2.2.2) |
| **REQ-AIML\_UPDATE-4** | The AI/ML Inference MnS producer should have a capability for the AI/ML MnS producer to inform an authorized MnS consumer about of the process or outcomes related to any request for updating the AI/ML capabilities or ML models | Triggering ML model update (clause 6.5.2.2.2) |
| **REQ-AIML\_UPDATE-5** | The AI/ML Inference MnS producer should have a capability for the AI/ML MnS producer to inform an authorized MnS consumer about of the achieved performance gain following the update of the AI/ML capabilities of a network function with/to the specific newly available ML models or set of AI/ML capabilities | Triggering ML model update (clause 6.5.2.2.2) |
| **REQ-AIML\_UPDATE-6** | The AI/ML Inference MnS producer should have a capability for an authorized MnS consumer (e.g., an operator or the function/ model that generated the request for updating the AI/ML capabilities) to manage the request and subsequent process, e.g. to suspend, re-activate or cancel the request or process; or to adjust the characteristics of the capability update; or to define how often the update may occur, suspend, restart or cancel the request or to further adjust the requirements of the update | Triggering ML model update (clause 6.5.2.2.2) |

6.5.3 AI/ML inference capabilities management

6.5.3.1 Description

A network or management function that applies AI/ML to accomplish specific tasks may be considered to have one or more ML models, each having specific capabilities.

Different network functions, e.g., MDA Functions, may need to rely on existing AI/ML capabilities to accomplish the desired inference. However, the details of such ML-based solutions (i.e., which ML models are applied and how) for accomplishing those inference functionalities is not obvious. The management services are required to identify the capabilities of the involved ML models and to map those capabilities to the desired logic.

***Next change***

7.3a.1.2.2.2 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 |
| cancelRequest | O | T | T | F | T |
| suspendRequest | 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** |
| inferenceType Support Qualifier | Condition: MLTrainingRequest MOI represents the request for initial ML model training.  |
| mLEntityToTrainRef Support Qualifier | Condition: MLTrainingRequest MOI represents the request for ML model re-training. |
| mLEntityCoordinationGroupToTrainRef Support Qualifier | Condition: MLTrainingRequest MOI represents the request for joint training of a group of ML models. |

***Next change***

7.3a.1.2.2.2 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 |
| cancelRequest | O | T | T | F | T |
| suspendRequest | 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 Support Qualifier | Condition: MLTrainingRequest MOI represents the request for initial ML model training.  |
| mLModelRef Support Qualifier | Condition: MLTrainingRequest MOI represents the request for ML model re-training. |
| mLModelCoordinationGroupRef Support Qualifier | Condition: MLTrainingRequest MOI represents the request for joint training of a group of ML models. |

***Next change***

7.5 Attribute definitions

7.5.1 Attribute properties

**Table 7.5.1-1**

| **Attribute Name** | **Documentation and Allowed Values** | **Properties** |
| --- | --- | --- |
| mLModelId | It identifies the ML model.It is unique in each MnS producer.allowedValues: N/A. | type: 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: the values of the MDA type (see 3GPP TS 28.104 [2]), Analytics ID(s) of NWDAF (see 3GPP TS 23.288 [3]), types of inference for RAN, and vendor's specific extensions. | type: Stringmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| areConsumerTrainingDataUsed | It indicates whether the consumer provided training data have been used for the ML model training.allowedValues: ALL, PARTIALLY, NONE. | type: Enummultiplicity: 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.allowedValues: N/A. | type: Stringmultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| trainingRequestRef | It is the DN(s) of the related MLTrainingRequest MOI(s).allowedValues: DN. | 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.allowedValues: DN. | 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.allowedValues: DN. | 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.allowedValues: DN. | type: DN multiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: True |
| 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 describes the model that requested to instantiate the MLTrainingRequest MOI.This attribute can be of type String or DN. | 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. Default value is set to "FALSE". 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. Default value is set to "FALSE".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.Default value is set to "FALSE".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. Default value is set to "FALSE".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.Default value is set to “FALSE“.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. Default value is set to "FALSE". allowedValues: TRUE, FALSE. | type: Booleanmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: False |
| inferenceModelRef | 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 the other new training data have been 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.allowedValues: DN | type: DNmultiplicity: 0..1isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| MLTrainingReport. mLModelGeneratedRef | It identifies the DN of the MLModel generated by the ML training.allowedValues: DN | type: DNmultiplicity: 1isOrdered: FalseisUnique: TruedefaultValue: 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.Default value is set to "FALSE". 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. Default value is set to "FALSE". 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.AllowedValues: DN | type: DNMultiplicity: 0..1isOrdered: FalsoisUnique: TruedefaultValue: NoneisNullable: True |
| 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 (including the inference result for each testing data example) is provided.The detailed testing result format is vendor specific.allowedValues: N/A. | type: Stringmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: True |
| testingRequestRef | It identifies the DN of the MLTestingRequest MOI.allowedValues: DN | type: DNmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: True |
| 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 Default value is set to "FALSE". 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. Default value is set to "FALSE". 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.allowedValues: DN. | type: DNmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| mLUpdateRequestRef | It is the DN of the MLUpdateRequest MOI that represents an ML update request.allowedValues: DN. | type: DNmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| mLUpdateReportRef | It is the DN of the MLUpdateReport MOI that represents an ML update report.allowedValues: DN. | 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.mLModelModelRef | It indicates the DN of MLModel instances that can be updated. | type: DNmultiplicity: 1 .. \*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.Default value is set to "FALSE". 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. Default value is set to "FALSE". allowedValues: TRUE, FALSE. | type: Booleanmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: False |
| memberMLModelRefList | It identifies the list of member ML models within a level of an ML model coordination group.allowedValues: DN list | type: DNmultiplicity: 2..\*isOrdered: TrueisUnique: TruedefaultValue: None isNullable: False |
| MLTrainingRequest.mLModelCoordinationGroupRef | It identifies the DN of the MLModelCoordinationGroup requested to be trained.allowedValues: DN | type: DNmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| MLTrainingReport.mLModelCoordinationGroupGeneratedRef | It identifies the DN of the MLModelCoordinationGroup generated by ML training.AllowedValues: DN | type: DNmultiplicity: 1..\*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| MLTestingRequest.mLModelCoordinationGroupRef | It identifies the DN of the MLEntityCoordinationGroup requested to be tested.AllowedValues: DN | type: DNmultiplicity: 1..\*isOrdered: FalseisUnique: TruedefaultValue: 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 |
| sourceTrainedMLModelRef | It identifies the DN of the source trained MLModel whose copy has been loaded from the ML model repository to the inference function. allowedValues: DN | type: DNmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: True |
| 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.Default value is set to "FALSE". 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. Default value is set to "FALSE". 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: True |
| 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: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: True |
| 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. Default value is set to "FALSE". 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. Default value is set to "FALSE". allowedValues: TRUE, FALSE. | type: Booleanmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: False |
| MLModelLoadingRequestRef | It identifies the DN of the associated MLModelLoadingRequest.allowedValues: DN. | type: DNmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: True |
| MLModelLoadingPolicyRef | It identifies the DN of the associated MLModelLoadingPolicyRef.allowedValues: DN. | type: DNmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: True |
| LoadedMLModelRef | It identifies the DN of the MLModel that has been loaded to the inference function. allowedValues: DN | type: DNmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: True |
| 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 and capabilityName. allowedValues: N/A | type: AttributeValuePair multiplicity: \*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***