**3GPP TSG-SA5 Meeting #162 *S5-253329***

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

|  |
| --- |
| *CR-Form-v12.3* |
| **CHANGE REQUEST** |
|  |
|  |  | **CR** | **Input-to-draftCR** | **rev** |  | **Current version:** |  |  |
|  |
| *For* [***HELP***](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 TS28.105 update ML inference emulation  |
|  |  |
| ***Source to WG:*** | Nokia |
| ***Source to TSG:*** | SA5 |
|  |  |
| ***Work item code:*** | AIML\_MGT\_Ph2 |  | ***Date:*** | 2025-08-13 |
|  |  |  |  |  |
| ***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 canbe found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | *Use one of the following releases:Rel-8 (Release 8)Rel-9 (Release 9)Rel-10 (Release 10)Rel-11 (Release 11)…Rel-17 (Release 17)Rel-18 (Release 18)Rel-19 (Release 19) Rel-20 (Release 20)* |
|  |  |
| ***Reason for change:*** | to clarify description and NRM for the agreed use case on ML inference emulation environment selection. |
|  |  |
| ***Summary of change:*** | Changes to description and NRM for ML inference emulation environment selection .  |
|  |  |
| ***Consequences if not approved:*** | The information model for the agreed use case and requirement for ML inference emulation environment selection would be incomplete. |
|  |  |
| ***Clauses affected:*** | 6.3.2, 7.3a.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***

## 6.3 AI/ML inference emulation

### 6.3.1 Description

Before the ML model is applied in the production network, the MnS inference consumer may want to receive results of inference in one or more environments that emulate (to different extents) the expected inference characteristics, in a process that may be termed as Inference emulation. The Inference emulation phase enables this.

### 6.3.2 Use cases

#### 6.3.2.1 AI/ML inference emulation

After the ML model is validated and tested during development, the MnS consumer may wish to receive information from an inference emulation process that indicates if the ML model or the associated ML inference function is working correctly under certain runtime context.

The management system should have the capabilities enabling an MnS consumer:

- to request an inference emulation function to provide emulation reports on ML inference emulation of ML model in a given ML inference emulation environment; and

- to receive the results from running inference through an AI/ML inference emulation environment available at the emulation MnS producer.

- to configure an ML inference emulation producer and execute ML models in a controlled way. The managed function (instance) configured for emulation is not in normal operation, i.e., taking traffic.

6.3.2.2 ML inference emulation environment selection

Although an ML model may be well-trained, its performance in the production network can be difficult to predict and guarantee because the training environment and production network are not identical. If a trained or tested ML model is directly applied to the production network, it may negatively impact the production network.

ML emulation involves applying the ML model in an emulation environment to verify whether its performance meets the expected inference characteristics. Considering the diversity of ML inference scenarios, one or more emulation environments are provided, e.g., simulation environments, a digital twin of the network (based on simulation or emulation), a test network or even the real network under certain constrained conditions, each differing in terms of emulation scope, emulation performance, and impact on user experience, e.g., unlike emulaiting via live netwok network, a digital twin will not have impact on users. .

The management system should have the capability to enable an MnS consumer to determine and select the appropriate emulation environment to be triggered and to provide the necessary configuration properties related to that environment. The configuration information may include defining the scope of the emulation, the emulation time, and other relevant parameters. The inference emulation should indicate the environment for ML model emulation.

### 6.3.3 Requirements for Managing AI/ML inference emulation

Table 6.3.3-1

| Requirement label | Description | Related use case(s) |
| --- | --- | --- |
| **REQ-AI/ML\_EMUL-1** | The MnS producer for AI/ML inference emulation should have a capability enabling an authorized MnS consumer to request emulation of an ML model and receive reporting about the ML inference emulation and environment for which the ML model has been emulated.  | AI/ML Inference emulation (clause 6.3.2.1)ML inference emulation environment selection (clause 6.3.2.2) |
| **REQ-AI/ML\_EMUL-2** | The MnS producer for AI/ML inference emulation should have a capability enabling an authorized MnS consumer to request an inference emulation function and provide inference emulation reports on an ML model or inference Function. | AI/ML Inference emulation (clause 6.3.2.1) |
| **REQ-EMUL\_SEL-1** | The MnS producer for AI/ML inference emulation should have a capability enabling an authorized MnS consumer to read information on the available emulation environment(s) or select the emulation environment. | ML inference emulation environment selection (clause 6.3.2.2) |
| **REQ-AI/ML\_ SEL-2** | The MnS producer for AI/ML inference emulation should have a capability to allow an authorized MnS consumer to manage or control the ML inference emulation and inference emulation environment. | ML inference emulation environment selection (clause 6.3.2.2) |

***Start of First change***

# 7 Information model definitions for AI/ML management

## 7.3a Information model definitions for AI/ML operational phases

### 7.3a.2 Information model definitions for AI/ML inference emulation

#### 7.3a.2.1 Class diagram

##### 7.3a.2.1.1 Relationships



Figure 7.3a.2.1.1-1: NRM fragment for AI/ML inference emulation control

##### 7.3a.2.1.2 Inheritance



Figure 7.3a.2.1.2-1: AI/ML inference emulation Inheritance Relations

#### 7.3a.2.2 Class definitions

##### 7.3a.2.2.1 AIMLInferenceEmulationFunction

###### 7.3a.2.2.1.1 Definition

This IOC represents the properties of a function that undertakes AI/ML Inference Emulation.

This AIMLInferenceEmulationFunction instance is created by the system (AI/ML inference emulation MnS producer) or pre-installed, it can only be deleted by the system.

An AIMLInferenceEmulationFunction may be associated with one or more MLModel(s). AIMLInferenceFunction is name contained with AIMLInferenceEmulationReport(s) that delivers the outcomes of the emulation processes.

An AIMLInferenceEmulationFunction is associated with one or more AI/ML Inference Emulation Environment(s) identified by a type and enabling the MnS consumer to execute, manage and control the AIML inference on several environments,. An example environment may be a network digital twin, a test network or the live network under certain conditions, e.g. during the maintenance window.

NOTE: The way of triggering of an AI/ML inference emulation and the instantiation of the related AI/ML inference emulation process is not in the scope of the present document.

An AIMLInferenceEmulationFunction output reuses the AIMLinferenceReport

###### 7.3a.2.2.1.2 Attributes

The AIMLInferenceEmulationFunction IOC inherited from ManagedFunction IOC (defined in TS 28.622 [12]).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable  | isWritable | isInvariant | isNotifyable |
| EnvironmentType | M | T | F | F | T |
| simScope | M | T | F | T | T |
| **Attribute related to role** |  |  |  |  |  |
| mLModelRef | M | T | F | T | T |

###### 7.3a.2.2.1.3 Attribute constraints

None.

###### 7.3a.2.2.1.4 Notifications

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

***Next change***

## 7.5 Attribute definitions

### 7.5.1 Attribute properties

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: see clause 7.4.10 | type: Stringmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| mDAType | It indicates the type of inference that the ML model for MDA supports. The detailed definition and corresponding allowed values for mDAType see TS 28.104 [2]. | type: MDAType (TS 28.104 [2])multiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| nwdafAnalyticsType | It indicates the type of inference that the ML model for NWDAF supports. The detailed definition and corresponding allowed values for nwdafAnalyticsID see NwdafEvent in TS 29.520 [20]. | type: NwdafEvent (TS 29.520 [20])multiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| ngRanInferenceType | It indicates the type of inference that the ML model for NG-RAN supports. The detailed definition and corresponding allowed values for ngRanInferenceType see clause 7.4a.1 | type: NgRanInferenceTypemultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| vSExtensionType | It indicates the type of inference that is vendor's specific extension.allowedValues: N/A. | type: Stringmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| usedConsumerTrainingData | It provides the address(es) where lists of the consumer-provided training data are located, which have been used for the ML model training.allowedValues: N/A. | type: Stringmultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| trainingRequestRef | It is the DN(s) of the related MLTrainingRequest MOI(s). | type: DN multiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| trainingProcessRef | It is the DN(s) of the related MLTrainingProcess MOI(s) that produced the MLTrainingReport. | type: DN multiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| trainingReportRef | It is the DN of the MLTrainingReport MOI that represents the reports of the ML model training. | type: DN multiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| lastTrainingRef | It is the DN of the MLTrainingReport MOI that represents the reports for the last training of the ML model(s). | type: DN multiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| modelConfidenceIndication | It indicates the average confidence value (in unit of percentage) that the ML model would perform for inference on the data with the same distribution as training data.Essentially, this is a measure of degree of the convergence of the trained ML model.allowedValues: { 0..100 }. | type: Integermultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| trainingRequestSource | It identifies the entity that requested to instantiate the MLTrainingRequest MOI.This attribute is the DN of a managed entity, otherwise, it is a String. | type: <<Choice>>multiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| MLTrainingRequest.requestStatus | It describes the status of a particular ML model training request.allowedValues: NOT\_STARTED, IN\_PROGRESS, CANCELLING, SUSPENDED, FINISHED, and CANCELLED. | type: Enummultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| mLTrainingProcessId | It identifies the training process.It is unique in each instantiated process in the MnS producer.allowedValues: N/A. | type: Stringmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| priority | It indicates the priority of the training process.The priority may be used by the ML model training to schedule the training processes. Lower value indicates a higher priority.allowedValues: { 0..65535 }. | type: Integermultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: 0 isNullable: False |
| terminationConditions | It indicates the conditions to be considered by the ML training MnS producer to terminate a specific training process.allowedValues: N/A. | type: Stringmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| progressStatus | It indicates the status of the process.allowedValues: N/A. | type: ProcessMonitor multiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| MLUpdateProcess.cancelProcess | It allows the ML update MnS consumer to cancel the ML update process.Setting this attribute to "TRUE" cancels the ML update process. Setting the attribute to "FALSE" has no observable result. allowedValues: TRUE, FALSE. | type: Booleanmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: False |
| MLUpdateProcess.suspendProcess | It allows the ML update MnS consumer to suspend the ML update process.Setting this attribute to "TRUE" suspends the ML update process. The process can be resumed by setting this attribute to “FALSE” when it is suspended. Setting the attribute to "FALSE" has no observable result.allowedValues: TRUE, FALSE. | type: Booleanmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: False |
| mLModelVersion | It indicates the version number of the ML model.allowedValues: N/A. | type: Stringmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| performanceRequirements | It indicates the expected performance for a trained ML model when performing on the training data.allowedValues: N/A. | type: ModelPerformancemultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| modelPerformanceTraining | It indicates the performance score of the ML model when performing on the training data.allowedValues: N/A. | type: ModelPerformancemultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| MLTrainingProcess.progressStatus.progressStateInfo | It provides the following specialization for the “progressStateInfo“ attribute of the “ProcessMonitor“ data type for the “MLTrainingProcess.progressStatus“.When the ML model training is in progress, and the " mLTrainingProcess.progressStatus.status " is equal to "RUNNING", it provides the more detailed progress information.allowedValues for " mLTrainingProcess.progressStatus.status " = "RUNNING":- “COLLECTING\_DATA”- “PREPARING\_TRAINING\_DATA”- “TRAINING” + DN of the MLModel being trainedThe allowed values for " mLTrainingProcess.progressStatus.status " = "CANCELLING" are vendor specific.The allowed values for " mLTrainingProcess.progressStatus.status " = "NOT\_STARTED" are vendor specific. | type: Stringmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| inferenceOutputName | It indicates the name of an inference output of an ML model.allowedValues: the name of the MDA output IEs (see 3GPP TS 28.104 [2]), name of analytics output IEs of NWDAF (see TS 23.288 [3]), RAN inference output IE name(s), and vendor's specific extensions. | type: Stringmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| performanceMetric | It indicates the performance metric used to evaluate the performance of an ML model, e.g. "accuracy", "precision", "F1 score", etc.allowedValues: N/A. | type: Stringmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| performanceScore | It indicates the performance score (in unit of percentage) of an ML model when performing inference on a specific data set (Note).The performance metrics may be different for different kinds of ML models depending on the nature of the model. For instance, for numeric prediction, the metric may be accuracy; for classification, the metric may be a combination of precision and recall, like the "F1 score".allowedValues: { 0..100 }. | type: Realmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| MLTrainingRequest.cancelRequest | It allows the ML training MnS consumer to cancel the ML model training request.Setting this attribute to "TRUE" cancels the ML model training request. The request can be resumed by setting this attribute to "FALSE" when it is suspended. Cancellation is possible when the requestStatus is the "NOT\_STARTED", " IN\_PROGRESS", and "SUSPENDED" state. Setting the attribute to "FALSE" has no observable result.allowedValues: TRUE, FALSE. | type: Booleanmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: False |
| MLTrainingRequest.suspendRequest | It allows the ML training MnS consumer to suspend the ML model training request.Setting this attribute to "TRUE" suspends the ML model training process. Suspension is possible when the requestStatus is not the "FINISHED" state. Setting the attribute to "FALSE" has no observable result. allowedValues: TRUE, FALSE. | type: Booleanmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: False |
| MLTrainingProcess.cancelProcess | It allows the ML training MnS consumer to cancel the ML model training process.Setting this attribute to “TRUE“ cancels the ML model training process. Cancellation is possible when the “mLTrainingProcess.progressStatus.status“ is not the “FINISHED“ state. Setting the attribute to “FALSE“ has no observable result.allowedValues: TRUE, FALSE. | type: Booleanmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: False |
| MLTrainingProcess.suspendProcess | It allows the ML training MnS consumer to suspend the ML model training process.Setting this attribute to "TRUE" suspends the ML model training process. The process can be resumed by setting this attribute to “FALSE” when it is suspended. Suspension is possible when the " mLTrainingProcess.progressStatus.status" is not the "FINISHED", "CANCELLING" or "CANCELLED" state. Setting the attribute to "FALSE" has no observable result. allowedValues: TRUE, FALSE. | type: Booleanmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: False |
| inferenceEntityRef | It describes the target entities that will use the ML model for inference. | type: DN multiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| dataProviderRef | It describes the entities that have provided or should provide data needed by the ML model e.g. for training or inference | type: DN multiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| areNewTrainingDataUsed | It indicates whether new training data are used for the ML model training.allowedValues: TRUE, FALSE. | type: Booleanmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| trainingDataQualityScore | It indicates numerical value that represents the dependability/quality of a given observation and measurement type. The lowest value indicates the lowest level of dependability of the data, i.e. that the data is not usable at all. allowedValues: { 0..100 }. | type: Realmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| decisionConfidenceScore | It is the numerical value that represents the dependability/quality of a given decision generated by the AI/ML inference function. The lowest value indicates the lowest level of dependability of the decisions, i.e. that the data is not usable at all.allowedValues: { 0..100 }. | type: Realmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| expectedRuntimeContext | This describes the context where an MLModel is expected to be applied.allowedValues: N/A | type: MLContextmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| trainingContext | This specifies the context under which the MLModel has been trained.allowedValues: N/A | type: MLContextmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| runTimeContext | This specifies the context where the MLmodel or model is being applied.allowedValues: N/A | type: MLContextmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| MLTrainingRequest.mLModelRef | It identifies the DN of the MLModel requested to be trained. | type: DNmultiplicity: 0..1isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| MLTrainingReport.mLModelGeneratedRef | It identifies the DN of the MLModel generated by the ML model training. | type: DNmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| mLModelRepositoryRef | It identifies the DN of the MLModelRepository. | type: DNmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| mLRepositoryId | It indicates the unique ID of the ML repository. | type: Stringmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| modelPerformanceValidation | It indicates the performance score of the ML model when performing on the validation data.allowedValues: N/A | type: ModelPerformancemultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| dataRatioTrainingAndValidation | It indicates the ratio (in terms of quantity of data samples) of the training data and validation data used during the training and validation process. It is represented by the percentage of the validation data samples in the total training data set (including both training data samples and validation data samples). The value is an integer reflecting the rounded number of percent \* 100. allowedValues: { 0 .. 100 }. | type: Integermultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| MLTestingRequest.requestStatus | It describes the status of a particular ML testing request.allowedValues: NOT\_STARTED, IN\_PROGRESS, CANCELLING, SUSPENDED, FINISHED, and CANCELLED. | type: Enummultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| MLTestingRequest.cancelRequest | It allows the ML testing MnS consumer to cancel the ML testing request.Setting this attribute to "TRUE" cancels the ML testing request. Cancellation is possible when the requestStatus is the "NOT\_STARTED", " IN\_PROGRESS", and "SUSPENDED" state. Setting the attribute to "FALSE" has no observable result.allowedValues: TRUE, FALSE. | type: Booleanmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: False |
| MLTestingRequest.suspendRequest | It allows the ML testing MnS consumer to suspend the ML testing request.Setting this attribute to "TRUE" suspends the ML testing request. The request can be resumed by setting this attribute to “FALSE” when it is suspended. Suspension is possible when the requestStatus is not the "FINISHED" state. Setting the attribute to "FALSE" has no observable result. allowedValues: TRUE, FALSE. | type: Booleanmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: False |
| MLTestingRequest.mLModelRef | It identifies the DN of the MLModel requested to be tested. | type: DNMultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| modelPerformanceTesting | It indicates the performance score of the ML model when performing on the testing data.allowedValues: N/A. | type: ModelPerformancemultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| mLTestingResult | It provides the address where the testing result (including the inference result for each testing data example) is provided.The detailed testing result format is vendor specific.allowedValues: N/A. | type: Stringmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| testingRequestRef | It identifies the DN of the MLTestingRequest MOI. | type: DNmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| supportedPerformanceIndicators | This parameter lists specific PerformanceIndicator(s) of an ML model.allowedValues: N/A. | type: SupportedPerfIndicator multiplicity: 1..\*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| performanceIndicatorName | It indicates the identifier of the specific performance indicator.allowedValues: N/A | type: Stringmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| isSupportedForTraining | It indicates whether the specific performance indicator is supported a performance metric of ML model training for the ML model. allowedValues: TRUE, FALSE. | type: Booleanmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: False |
| isSupportedForTesting | It indicates whether the specific performance indicator is supported a performance metric of ML model testing for the ML model. allowedValues: TRUE, FALSE. | type: Booleanmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: False |
| mLUpdateProcessRef | It is the DN of the mLUpdateProcess MOI that represents the process of updating an ML model. | type: DNmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| mLUpdateRequestRefList | It is the list of DN of the MLUpdateRequest MOI that represents an ML update request. | type: DNmultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| mLUpdateReportRef | It is the DN of the MLUpdateReport MOI that represents an ML update report. | type: DNmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| mLUpdateReportingPeriod | It specifies the time duration upon which the MnS consumer expects the ML update is reported. | type: TimeWindowmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| availMLCapabilityReport | It represents the available ML capabilities.allowedValues: N/A. | type: AvailMLCapabilityReport multiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| updatedMLCapability | It represents the updated ML capabilities.allowedValues: N/A. | type: AvailMLCapabilityReport multiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| availMLCapabilityReportID | It identifies the available ML capability report.allowedValues: N/A. | type: Stringmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| newCapabilityVersionId | It indicates the specific version of AI/ML capabilities to be applied for the update. It is typically the one indicated by the MLCapabilityVersionID in a newCapabilityVersion | type: Stringmultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| mlCapabilityVersionId | It indicates the version of ML capabilities that is available for the update.  | type: Stringmultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| performanceGainThreshold | It defines the minimum performance gain as a percentage that shall be achieved with the capability update, i.e., the difference in the performances between the existing capabilities and the new capabilities should be at least performanceGainThreshold otherwise the new capabilities should not be applied.Allowed value: float between 0.0 and 100.0 | type: ModelPerformancemultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| expectedPerformanceGains | It indicates the expected performance gain if/when the AI/ML capabilities of the respective network function are updated with/to the specific set of newly available AI/ML capabilities. | type: ModelPerformancemultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: NoneisNullable: False |
| updateTimeDeadline | It indicates the maximum as stated in the MLUpdate request that should be taken to complete the update | type: TimeWindowmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| MLUpdateReport.mLModelRefList | It indicates the DN of MLModel instances that can be updated. | type: DNmultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: NoneisNullable: False |
| MLUpdateRequest.requestStatus | It describes the status of a particular ML update request.allowedValues: NOT\_STARTED, IN\_PROGRESS, CANCELLING, SUSPENDED, FINISHED, and CANCELLED. | type: Enummultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| MLUpdateRequest.cancelRequest | It allows the MnS consumer to cancel the ML update request.Setting this attribute to "TRUE" cancels the ML update request. Cancellation is possible when the requestStatus is the "NOT\_STARTED", " IN\_PROGRESS", and "SUSPENDED" state. Setting the attribute to "FALSE" has no observable result.allowedValues: TRUE, FALSE. | type: Booleanmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: False |
| MLUpdateRequest.suspendRequest | It allows the MnS consumer to suspend the ML update request.Setting this attribute to "TRUE" suspends the ML update request. The request can be resumed by setting this attribute to “FALSE” when it is suspended. Suspension is possible when the requestStatus is not the "FINISHED" state. Setting the attribute to "FALSE" has no observable result. allowedValues: TRUE, FALSE. | type: Booleanmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: False |
| memberMLModelRefList | It identifies the list of member ML models within an ML model coordination group. | type: DNmultiplicity: 2..\*isOrdered: TrueisUnique: TruedefaultValue: None isNullable: False |
| MLTrainingRequest.mLModelCoordinationGroupRef | It identifies the DN of the MLModelCoordinationGroup requested to be trained. | type: DNmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| MLTrainingReport.mLModelCoordinationGroupGeneratedRef | It identifies the DN of the MLModelCoordinationGroup generated by ML model joint training. | type: DNmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| MLTestingRequest.mLModelCoordinationGroupRef | It identifies the DN of the MLModelCoordinationGroup requested to be tested. | type: DNmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| retrainingEventsMonitorRef | It indicates the DN of the ThresholdMonitor MOI that indicates the performance measurements and its corresponding thresholds to be used by MnS producer to initiate the re-training of the MLModel. | type: DNmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| MLModelLoadingRequest.requestStatus | It describes the status of a particular ML model loading request.allowedValues: NOT\_STARTED, IN\_PROGRESS, CANCELLING, SUSPENDED, FINISHED, and CANCELLED. | type: Enummultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| MLModelLoadingRequest.cancelRequest | It allows the MnS consumer to cancel the ML model loading request.Setting this attribute to "TRUE" cancels the ML model loading. Cancellation is possible when the requestStatus is the "NOT\_STARTED", " IN\_PROGRESS", and "SUSPENDED" state. Setting the attribute to "FALSE" has no observable result.allowedValues: TRUE, FALSE. | type: Booleanmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: False |
| MLModelLoadingRequest.suspendRequest | It allows the MnS consumer to suspend the ML model loading request.Setting this attribute to "TRUE" suspends the ML model loading request. The request can be resumed by setting this attribute to “FALSE” when it is suspended. Suspension is possible when the requestStatus is not the "FINISHED" state. Setting the attribute to "FALSE" has no observable result. allowedValues: TRUE, FALSE. | type: Booleanmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: False |
| mLModelToLoadRef | It identifies the DN of a trained MLModel requested to be loaded to the target inference function(s). | type: DNmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| policyForLoading | It provides the policy for controlling ML model loading triggered by the MnS producer.This policy contains two thresholds in the thresholdList attribute. The first threshold is related to the ML model to be loaded, and the second threshold is related to the existing ML model being used for inference. | type: AIMLManagementPolicymultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| thresholdList | It provides the list of threshold.  | type: ThresholdInfomultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| MLModelLoadingProcess.progressStatus.progressStateInfo | It provides the following specialization for the "progressStateInfo" attribute of the "ProcessMonitor" data type for the "MLModelLoadingProcess.progressStatus".When the ML model loading is in progress, and the " MLModelLoadingProcess.progressStatus.status " is equal to "RUNNING", it provides the more detailed progress information.allowedValues for " MLModelLoadingProcess.progressStatus.status " = "RUNNING":The allowed values for " MLModelLoadingProcess.progressStatus.status " = "CANCELLING" are vendor specific.The allowed values for " MLModelLoadingProcess.progressStatus.status " = "NOT\_STARTED" are vendor specific. | type: Stringmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| MLModelLoadingProcess.cancelProcess | It allows the MnS consumer to cancel the ML model loading process.Setting this attribute to "TRUE" cancels the process. Cancellation is possible when the "MLModelLoadingProcess.progressStatus.status" is not the "FINISHED" state. Setting the attribute to "FALSE" has no observable result. allowedValues: TRUE, FALSE. | type: Booleanmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: False |
| MLModelLoadingProcess.suspendProcess | It allows the MnS consumer to suspend the ML model loading process.Setting this attribute to "TRUE" suspends the process. The process can be resumed by setting this attribute to "FALSE" when it is suspended. Suspension is possible when the "MLModelLoadingProcess.progressStatus.status" is not the "FINISHED", "CANCELLING" or "CANCELLED" state. Setting the attribute to "FALSE" has no observable result. allowedValues: TRUE, FALSE. | type: Booleanmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: FALSEisNullable: False |
| mLModelLoadingRequestRef | It identifies the DN of the associated MLModelLoadingRequest. | type: DNmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| mLModelLoadingPolicyRef | It identifies the DN of the associated MLModelLoadingPolicyRef. | type: DNmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| loadedMLModelRef | It identifies the DN of the MLModel that has been loaded to the inference function.  | type: DNmultiplicity: 0..1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| activationStatus | It describes the activation status.allowedValues: ACTIVATED, DEACTIVATED. | type: Enummultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| AIMLManagementPolicy.managedActivationScope | It provides a list of sub scopes for which ML inference is activated as triggered by a policy on the MnS producer. For example, the sub scopes may be a list of cells or of geographical areas. The list is an ordered list indicating the inference is activated for the first sub scope and gradually extended to the next sub scope if the policy evaluates to true.allowedValues: N/A | type: ManagedActivationScopemultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| AIMLInferenceFunction.managedActivationScope | It provides a list of sub scopes for which ML inference is activated as triggered by a policy on the MnS producer. For example, the sub scopes may be a list of cells or of geographical areas. The list is an ordered list indicating the inference is activated for the first sub scope and gradually extended to the next sub scope if the policy evaluates to true.allowedValues: N/A | type: AIMLManagementPolicymultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| ManagedActivationScope.dNList | It indicates the list of DN, the list is an ordered list indicating the inference is activated for the first sub scope and gradually extended to the next sub scope.allowedValues: N/A | type: DNmultiplicity: \*isOrdered: TrueisUnique: TruedefaultValue: None isNullable: False |
| ManagedActivationScope.timeWindow | It indicates the list of time window; the list is an ordered list indicating the inference is activated for the first sub scope and gradually extended to the next sub scope.allowedValues: N/A | type: TimeWindowmultiplicity: \*isOrdered: TrueisUnique: TruedefaultValue: None isNullable: False |
| ManagedActivationScope.geoPolygon | It indicates the list of GeoArea, the list is an ordered list indicating the inference is activated for the first sub scope and gradually extended to the next sub scope.allowedValues: N/A | type: GeoAreamultiplicity: \*isOrdered: TrueisUnique: TruedefaultValue: None isNullable: False |
| usedByFunctionRefList | It provides the DNs of the functions supported by the AIMLInferenceFunction.allowedValues: N/A | type: DNmultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| inferenceOutputId  | It identifies an inference output within an AIMLinferenceReport. | type: Stringmultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| inferenceOutputs | It indicates the Outputs that have been derived by the AIMLInferenceFunction instance from a specific ML model.Each ML model, inferenceOutputs may be a set of values.allowedValues: N/A. | type: InferenceOutputmultiplicity: 1..\*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| inferencePerformance | It indicates the performance score of the ML model during Inference.allowedValues: N/A. | type: ModelPerformancemultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| inferenceOutputTime | It indicates the time at which the inference output is generated.allowedValues: N/A | type: DateTimemultiplicity: \*isOrdered: TrueisUnique: TruedefaultValue: None isNullable: False |
| outputResult | It indicates the result of an inference. | type: AttributeValuePairmultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: NullisNullable: False |
| mLCapabilitiesInfoList | It indicates information about what an ML model can generate inference for. allowedValues: N/A. | type: MLCapabilityInfomultiplicity: 1..\*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| capabilityName | It indicates the name of a capability for which an ML model can generate inference. The capability is defined by Mns producer which can be traffic analysis capability, coverage analysis capability, mobility analysis capability or vendor specific extensions.allowedValues: N/A. | type: Stringmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| mLCapabilityParameters | It indicates a set of optional parameters that apply for an aIMLInferenceName capabilityName. allowedValues: N/A | type: AttributeValuePair multiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| aIMLInferenceReportRefList | It indicates a list of DN of the AIMLInferenceReport MOI that represents an AIML inference report. | type: DNmultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| mLModelRefList | It identifies the list of MLModel DN. | type: DNmultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: None isNullable: False |
| EnvironmentType | It identifies the type of ML inference emulation environment Allowed values: NDT, TESTNETWORK, LIVENETWORK | type: ENUMmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: None isNullable: False |
| simScope | It provides a list of sub scopes for which the ML inference emulation environment can undertake emulation. The total scope is the union of the list of subscopes. allowedValues: N/A | type: ManagedActivationScopemultiplicity: 1..\*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***