**3GPP TSG-SA5 Meeting #143eS5-223395**

**09 - 17 May 2022, E-meeting**

**Source: Nokia**

**Title: pCR 28.105 Add requirements for pre-processed event data for ML training**

**Document for: Approval**

**Agenda Item: 6.6.5**

# 1 Decision/action requested

***The group is asked to discuss and agree on the proposal.***

# 2 References

[1] 3GPP TS 28.105-110 “Management and orchestration; AI/ML management”.

# 3 Rationale

The AI/ML training needs to support the capabilities for handling errors in input data or in the learned decisions. This pCR presents the corresponding requirements.

# 4 Detailed proposal

|  |
| --- |
| **Start of modifications** |

## 6.N Pre-processed event data for ML training

### 6.N.1 Description

In analytics solutions, Performance Measurements (PMs) and Fault Reports (FRs) from various network function are collected and analytics applied on the PMs and FRs to come up with statistical insights and predictions of events from the raw data. For most algorithms, the prediction accuracy depends upon the amount of relevant historical data, motivating the need to store ever more data, which correspondingly increases the storage and processing resource requirements. However, not all recorded data is useful as the derived events, e.g. captured through analytics processes, may have loss of information OR mis-information e.g., with respect to time of the event.

Relatedly for AI/ML algorithms, a large amount of data points does not necessarily add value, e.g., if most of it includes biased data which ends up getting discarded during the pre-processing stages. Instead, the AI/ML algorithms need to have information-rich events data that is condensed but with most of it useful for the required training. For example, one could train an interference optimization solution that learns the best way to combat interference by looking at counters of handover failures correlated with signal quality. However, for most of the time in the radio network, there will be no interference events but this cannot be determined if the events are not captured form the data. As such all the data must be kept and used for training. However, the data could also be mined for the interference event or signatures thereof. Then an equivalent interference management solution could be trained using the interference signatures or the signatures combined with only a small amount of raw data.

It is as such necessary to provide means to isolate and store the information rich events in the network, to ensure that minimizing storage and processing costs by discarding the unnecessary raw data does not compromise the ability to and still avails adequate historical information to adequately train AI/ML applications. In other words it is necessary for network functions to their management system to generate data on about the observed network events, e.g., based on the criteria set by the Operator, which events can then be stored to be used later to train AI/ML applications.

Network Function

AI/ML enabled Function as Network events Data Consumer

Network Function

Events processing

Events data

Management Function

Events processing

Network/system data

Network/system events data

Fig 3. Exposing and storing network events data

### 6.N.2 Requirements

|  |  |  |
| --- | --- | --- |
| **Requirement label** | **Description** | **Related use case(s)** |
| **ML\_Error\_Req\_1** | The 3GPP management system shall enable an authorized consumer to request from the network data producer for network events corresponding to the data produced by that network data producer. | Pre-processed event data for ML training |
| **ML\_Error\_Req\_2** | The 3GPP management system shall enable A network data producer to generate network events in place of or alongside the network data that they produce | Pre-processed event data for ML training |
| **ML\_Error\_Req\_3** | The 3GPP management system shall enable a network events aggregator to take the events from different network entities and re-expose them in an aggregated way that eliminates duplications | Pre-processed event data for ML training |

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

### 7.4.2 AIMLEntity <<dataType>>

#### 7.4.2.1 Definition

This data type represents the properties of an AI/ML entity which could be either an AI/ML Model or AI/ML-enabled function containing the AI/ML model. AIMLTraining may be requested for either an AI/ML Model or AI/ML-enabled function. The algorithm of AI/ML Model or AI/ML-enbaled function is not to be standardized.

For each AIMLEntity under training, one or more AIMLTrainingProcess are instantiated.

The AIMLEntity may contain 3 types of contexts - TrainingContext which is the context under which the AIMLEntity has been trained, the ExpectedRunTimeContext which is the context where an AIMLEntity is expected to be applied or and the RunTimeContext which is the context where the model is being applied.

#### 7.4.2.2 Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
| aIMLEntityId | M | T | F | F | T |
| inferenceType | M | T | F | F | T |
| aIMLEntityVersion | M | T | F | F | T |
| requiredNetworkEventsData | CO | T | F | F | T |
| ExpectedRunTimeContext | O | T | T | F | T |
| TrainingContext | CM | T | F | F | T |
| RunTimeContext | O | T | F | F | T |
| **Attribute related to role** |  |  |  |  |  |
|  |  |  |  |  |  |

#### 7.4.3.3 Attribute constraints

None.

|  |  |
| --- | --- |
| Name | Definition |
| TrainingContext Support Qualifier | Condition: The TrainingContext represents the status and conditions related to training and should be added when training is completed |
| requiredNetworkEventsData | Condition: The requiredNetworkEventsData are only required if the training of the AI/ML Entity requires network events data |

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

## 7.5 Attribute definitions

### 7.5.1 Attribute properties

| Attribute Name | Documentation and Allowed Values | Properties |
| --- | --- | --- |
| aIMLEntityId | It identifies the AI/ML entity.  It is unique in each MnS producer.  allowedValues: N/A. | type: String  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: True |
| candidateTraingDataSource | It provides the address(es) of the candidate training data source provided by MnS consumer. The detailed training data format is vendor specific.  allowedValues: N/A. | type: String  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: True |
| inferenceType | It indicates the type of inference that the AI/ML model supports.  allowedValues: the values of the MDA type (see TS 28.104 [2]), Analytics ID(s) of NWDAF (see TS 23.288 [3]), types of inference for RAN-intelligence, and vendor’s specific extensions. | type: String  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: True |
| areConsumerTrainingDataUsed | It indicates whether the consumer provided training data have been used for the AI/ML model training.  allowedValues: ALL, PARTIALLY, NONE. | type: Enum  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: True |
| usedConsumerTrainingData | It provides the address(es) where lists of the consumer-provided training data are located, which have been used for the AI/ML model training.  allowedValues: N/A. | type: String  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: True |
| trainingRequestRef | It is the DN(s) of the related AIMLTrainingRequest MOI(s).  allowedValues: DN. | type: DN (see TS 32.156 [12])  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: True |
| lastTrainingRef | It is the DN of the AIMLTrainingReport MOI that represents the reports for the last training of the AI/ML model.  allowedValues: DN. | type: DN (see TS 32.156 [12])  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: True |
| confidenceIndication | It indicates the confidence (in unit of percentage) that the AI/ML model would perform for inference on the data with the same distribution as training data.  allowedValues: { 0..100 }. | type: integer  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| aIMLEntityList | It describes the list of aIMLEntity. | type: AIMLEntity  multiplicity: \*  isOrdered: False  isUnique: N/True  defaultValue: None  isNullable: False |
| trainingRequestSource | It descriobes the entity that requested to instantiatethe AIMLTrainingRequest MOI. | type: integer  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| requestStatus | It describes the status of a particular AI/ML training request. T.  allowedValues: NOT\_STARTED, TRAINING\_IN\_PROGRESS, CANCELLING, SUSPENDED, FINISHED, and CANCELLED. | type: Enum  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| aIMLTrainingProcessId | It identifies the training process.  It is unique in each instantiated process in the MnS producer.  allowedValues: N/A. | type: String  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: True |
| priority | It indicates the priority of the training process.  The priority may be used by the AI/ML training to schedule the training processes.  allowedValues: { 0..100 }. | type: ENUM  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| terminationConditions | It indicates the conditions to be considered by the AIMLTraining to terminate a specific training process.  Editor's Note: The specific nature of the termination conditions is FFS  allowedValues: FFS. | type: FFS  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| progressStatus | It indicates the status of the AI/ML training process.  allowedValues: N/A. | type: ProcessMonitor (see TS 28.622 [11])  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| aIMLEntityVersion | It indicates the version number of the AI/ML entity.  allowedValues: N/A. | type: String  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| performanceRequirements | It indicates the expected performance for a trained AI/ML entity when performing on the training data.  allowedValues: N/A. | type: ModelPeformance  multiplicity: \*  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: True |
| performanceTraining | It indicates the performance score of the AI/ML entity when performing on the training data.  allowedValues: N/A. | type: ModelPeformance  multiplicity: \*  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| AIMLTrainingProcess.progressStatus.progressStateInfo | It provides the following specialisation for the "progressStateInfo" attribute of the "ProcessMonitor" data type for the "AIMLTrainingProcess".  When the AI/ML training is in progress, and the "status" is equal to " RUNNING" it provides the more detailed progress information.  allowedValues for "status" = "RUNNING":  - COLLECTING\_DATA  - PREPARING\_TRAINING\_DATA  - TRAINING  The allowed values for "status" = "CANCELLED" are vendor specific. | Type: String  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| inferenceOutputName | It indicates the name of an inference output of an AI/ML entity.  allowedValues: the name of the MDA output IEs (see TS 28.104 [2]), name of analytics output IEs of NWDAF (see TS 23.288 [3]), RAN-intelligence inference output IE name(s), and vendor’s specific extensions. | Type: String  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| performanceScore | It indicates the performance score (in unit of percentage) of an AI/ML entity when performing inference on a specific data set (Note).  The performance metrics may be different for different kinds of AI/ML models depending on the nature of the model. For instance, for numeric prediction, the metric may be accuracy; for classification, the metric may be a combination of precision and recall, like the “F1 score”.  allowedValues: { 0..100 }. | Type: Real  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| cancelRequest | It indicates whether the MnS consumer cancels the AI/ML training request.  Setting this attribute to "TRUE" cancels the AI/ML training request. Cancellation is possible when the requestStatus is the "NOT\_STARTED", " TRAINING\_IN\_PROGRESS", and "SUSPENDED" state. Setting the attribute to "FALSE" has no observable result.  Default value is set to "FALSE".  allowedValues: TRUE, FALSE. | Type: ENUM  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: FALSE  isNullable: False |
| suspendRequest | It indicates whether the MnS consumer suspends the AI/ML training request.  Setting this attribute to "TRUE" suspends the AI/ML training request. Suspension is possible when the requestStatus is the not “FINISHED" state. Setting the attribute to "FALSE" has no observable result.  Default value is set to "FALSE".  allowedValues: TRUE, FALSE. | Type: ENUM  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: FALSE  isNullable: False |
| cancelProcess | It indicates whether the MnS consumer cancels the AI/ML training process.  Setting this attribute to "TRUE" cancels the AI/ML training request. Cancellation is possible when the progressStateInfo is the not “FINISHED" state. Setting the attribute to "FALSE" has no observable result.  Default value is set to "FALSE".  allowedValues: TRUE, FALSE. | Type: ENUM  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: FALSE  isNullable: False |
| suspendProcess | It indicates whether the MnS consumer suspends the AI/ML training process.  Setting this attribute to "TRUE" suspends the AI/ML training request. Suspension is possible when the progressStateInfo is the not “FINISHED", “CANCELLING” or “CANCELLED” state. Setting the attribute to "FALSE" has no observable result.  Default value is set to "FALSE".  allowedValues: TRUE, FALSE. | Type: ENUM  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: FALSE  isNullable: False |
| managedEntityRef | It describes the entities that the MLApp is responsible for managing or optmimizing | Type: DN (see TS 32.156 [12])  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: True |
| dataProviderRef | It describes the entities that have provided or should provide data needed by the MLApp, say for training or inference | Type: DN (see TS 32.156 [12])  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: True |
| requiredNetworkEventsData | It describes the network events that are required as input to the training and inference of the MLEntity. The nature of the network events is FFS | Type: ENUM  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: FALSE  isNullable: False |
| NOTE: when the performanceScore is to indicate the performance score for AI/ML training, the data set is the training data set. | | |

### 7.5.2 Constraints

|  |  |  |
| --- | --- | --- |
| Name | Affected attribute(s) | Definition |
|  |  |  |

|  |
| --- |
| **End of modifications** |