**3GPP TSG-SA5 Meeting #156S5-243528**

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

**Source: Nokia**

**Title: Rel-19 pCR TR28.858 Update R19 continuation use cases**

**Document for: Approval**

**Agenda Item: 6.19.1**

# 1 Decision/action requested

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

# 2 References

[1] 3GPP TR 28.908-018 “Study on Artificial Intelligence/Machine Learning (AI/ ML) management”.

[2] 3GPP TR 28.858-010 “Study on Artificial Intelligence/Machine Learning (AI/ ML) Lifecycle Management (LCM) Phase 2”.

# 3 Rationale

This pCR is to update the R19 study use cases that are a continuation of R18.

# 4 Detailed proposal

|  |
| --- |
| **Start of modification** |

## 5 AI/ML management use cases and requirements

## 5.1 ML model training

### 5.1.1 ML-Knowledge-based Transfer Learning

Editor’s Note: The contentin this clause may need to be aligned with the latest version of TS28.105

#### 5.1.1.1 Description

It is known that existing ML capability can be leveraged in producing or improving new or other ML capability. Specifically, using transfer learning knowledge contained in one or more ML models may be transferred to another ML model. Transfer learning relies on task and domain similarity to deduce whether some parts of a deployed ML model can be reused in another domain / task with some modifications. aspects of transfer learning that are appropriate in multi-vendor environments need to be supported in network management systems. However, ML models are not to be multi-vendor objects, i.e. it will not be possible to transfer an ML model from one function to another**.** Instead, the knowledge contained in the model, called ML knowledge, should be transferred instead of transferring the ML model itself.

As an example, the knowledge contained in an ML model deployed to perform mobility optimization by day can be leveraged to produce a new ML model to perform mobility optimization by night. As such and as illustrated by figure 5.1.1.1-1, the network or its management system needs to have the required management services for ML Transfer Learning (MLKLT), where ML Transfer Learning refers to means to allow and support the usage and fulfilment of transfer learning between any two ML models.



Figure 5.1.1.1-1: ML Knowledge Transfer Learning (MLKLT) flow between the source MLKLT  
(which is the model with the pre-trained ML model), the peer MLKLT  
(which is the model that shall train a new ML model) and the MLKLT MnS consumer  
(which may be the operator or another management function that wishes to trigger or control MLKLT)

#### 5.1.1.2 Use cases

##### 5.1.1.2.1 Discovering sharable Knowledge

For the transfer learning, it is expected that the source ML Knowledge Transfer Learning MnS producer shares its knowledge with the target ML Training function, either simply as single knowledge transfer instance or through an interactive transfer learning process. The concept of knowledge here represents any experiences or information gathered by the MLModel in the ML Knowledge Transfer Learning MnS producer through training, inference, updates, or testing. This information or experiences can be in the form of - but not limited to - data statistics or other features of the underlying ML model. It may also be the output of an MLModel. The 3GPP management systems should provide means for an MnS consumer to discover this potentially shareable knowledge as well as means for the provider of MLKLT to share the knowledge with the MnS consumer.

##### 5.1.1.2.2 Knowledge sharing and transfer learning

The transfer learning may be triggered by a MnS consumer either to fulfil the learning for itself or for it to be accomplished through another ML Training function. The model containing the knowledge may be an independent managed entity (the ML model). Alternatively, the ML model may also be an entity that is not independently managed but is an attribute of a managed ML model or ML function in which case MLKLT does not involve sharing the ML model or parts thereof but may imply implementing the means and services to enable the sharing of knowledge contained within the ML model or ML-enabled function. The 3GPP management system should provide means and the related services needed to realize the ML transfer learning process.

Specifically, the 3GPP management system should provide means for an MnS consumer to request and receive sharable knowledge as well as means for the provider of MLKLT to share the knowledge with the MnS consumer or any stated target ML Training function. Similarly, the 3GPP management system should provide means for an MnS consumer to manage and control the MLKLT process and the related requests associated with transfer learning between two ML models or between the two ML models and a shared knowledge repository.

The two use cases should address the three scenarios represented by figures 5.1.1.2.2-1 to 5.1.1.2.2-4. Note that, the use case and requirements here focus on the required management capabilities. The implementation of the knowledge transfer learning processes are implementation details that are out of the scope of the present document.

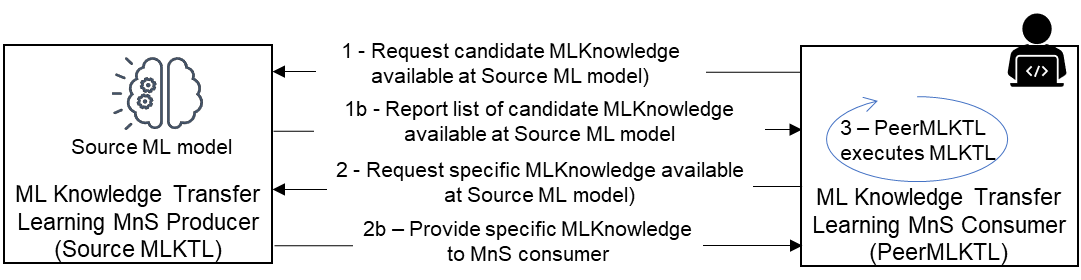


Figure 5.1.1.2.2-1: Scenario 1 - Interactions for ML-Knowledge Transfer Learning (MLKLT) to  
support training at the ML knowledge Transfer MnS consumer -  
the ML knowledge Transfer MnS consumer obtains the ML knowledge  
which it then uses for training the new ML model based on knowledge received  
from the MLKLT source MnS producer

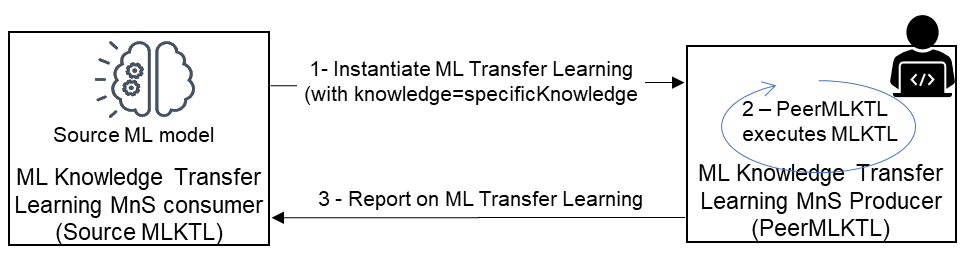


Figure 5.1.1.2.2-2: Scenario 2 - interactions for ML-Knowledge Transfer Learning (MLKTL) to  
support training at the ML knowledge transfer MnS consumer triggered by the MLKTL Source -  
the ML Transfer Learning MnS consumer acting as the MLKTLSource  
(the source of the ML knowledge) triggers the training at the ML knowledge Transfer MnS consumer by providing the ML knowledge to be used for the training,  
the ML Transfer Learning MnS consumer then undertakes the training

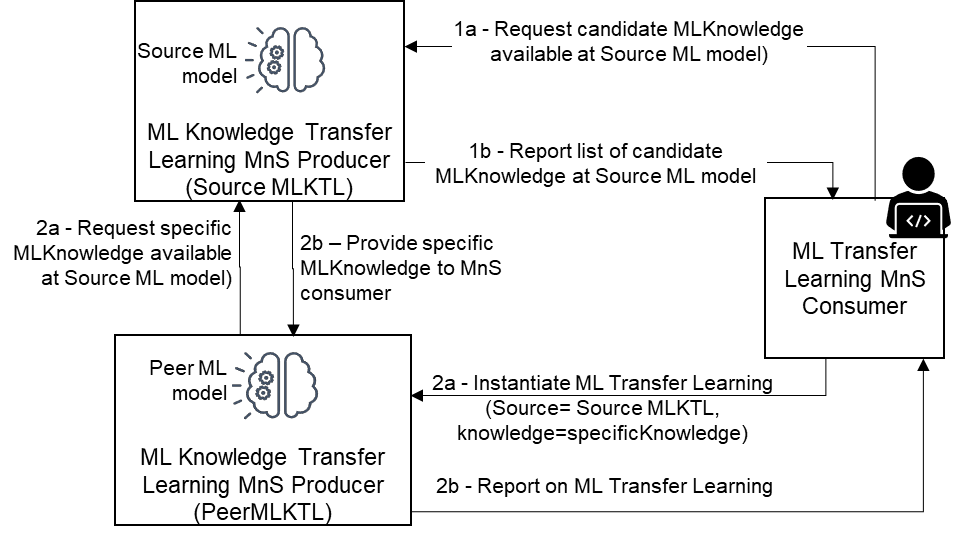


Figure 5.1.1.2.2-3: Scenario 3 - interactions for ML-Knowledge Transfer Learning (MLKLT) to  
support training at the Peer ML knowledge Transfer MnS producer who is different from the  
ML knowledge Transfer MnS consumer - the ML knowledge Transfer MnS consumer triggers  
training at the MLKLT peer MnS producer. The MLKLT MnS consumer then obtains the  
ML knowledge from the MLKLT source MnS producer and then uses the knowledge for training  
the new ML model based on knowledge received from the MLKLT source MnS producer

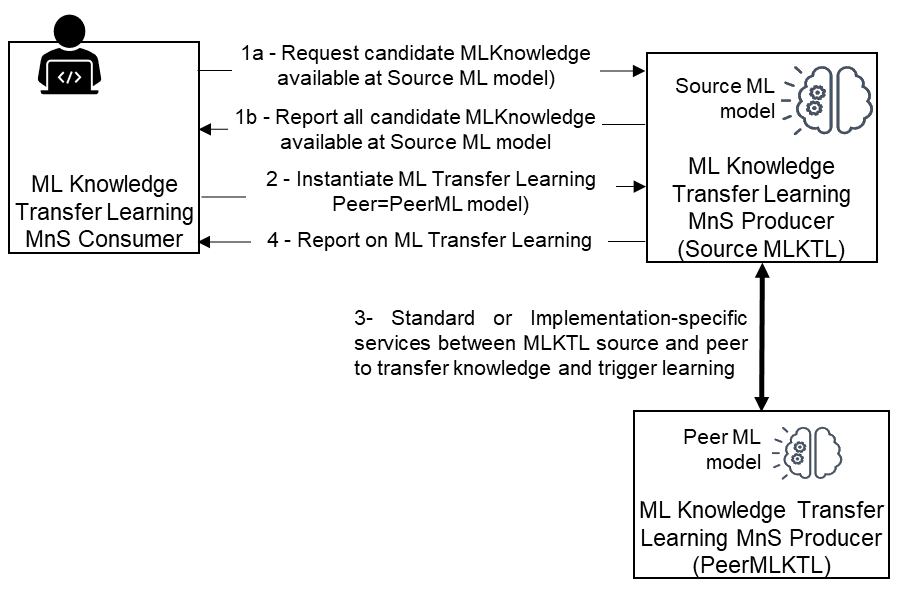


Figure 5.1.1.2.2-4: Scenario 4 - interactions for ML-Knowledge Transfer Learning (MLKLT) to  
support training at the Source ML knowledge Transfer MnS producer -  
the ML knowledge Transfer MnS consumer triggers training at the MLKLT source MnS producer.  
The MLKLT MnS consumer then obtains the ML knowledge from the MLKLT source MnS producer  
and then uses the knowledge for training the new ML model based on knowledge received from  
the MLKLT source MnS producer

#### 5.1.1.3 Potential requirements

**REQ-MLKLT-1:** The 3GPP management systemshould have a capability enabling an authorized MnS consumer to discover the available shared knowledge from a given MLKLT MnS producer according to a stated set of criteria.

**REQ-MLKLT-2:** The 3GPP management systemshould have a capability enabling an authorized MnS consumer to request a MLKLT MnS producer to provide some or all the knowledge available for sharing according to some stated criteria.

**REQ-MLKLT-3:** The 3GPP management systemshould have a capability for a MLKLT MnS producer to report to an authorized MnS consumer on the available shared knowledge according to a ReportingCriteria specified in a request for information on available Knowledge.

**REQ-MLKLT-4:** The 3GPP management systemshould have a capability enabling an authorized MnS consumer to request a MLKLT MnS producer to trigger and execute a transfer learning instance to a specified ML model or ML-enabled function.

**REQ-MLKLT-5:** The 3GPP management systemshould have a capability for an authorized MnS consumer (e.g. an operator or the function/entity that generated the request for available Knowledge or for information thereon) to manage the request for knowledge or its information and subsequent process, e.g. to suspend, re-activate or cancel the MLKnowledgeRequest; or to adjust the description of the desired knowledge.

**REQ-MLKLT-6:** The 3GPP management systemshould have a capability for an authorized MnS consumer (e.g. an operator or the function/entity that generated the request for MLKLT) to manage or control a specific MLKLTJob, e.g. to start, suspend or restart the MLKLTJob; or to adjust the transfer learning conditions or characteristics i.e. Modify MLKLTJob attributes.

Note: the MLKLTJob represents the process of knowledge-based trasfer learning.

**REQ-MLKLT-7:** The 3GPP management systemshould have a capability enabling an MLModel to register available knowledge to a shared knowledge repository, e.g. through a MLKnowledgeRegistration process.

**REQ-MLKLT-8:** The 3GPP management systemshould have a capability enabling KnowledgeRepo to act as the MLKLT MnS Producer to enable an authorized MnS consumer to request the shared knowledge repository to provide information on the available knowledge according to some given criteria.

**REQ-MLKLT-9:** The 3GPP management systemshould have a capability enabling KnowledgeRepo to act as the MLKLT MnS Producer to enable an authorized MnS consumer to request the KnowledgeRepo to provide some or all the knowledge available for sharing according to some given criteria.

**REQ-MLKLT-10:** The 3GPP management systemshould have a capability enabling KnowledgeRepo to act as the MLKLT MnS Producer to enable an authorized MnS consumer (e.g. an operator or the function/entity that generated the MLKnowledgeRequest) to manage the request, e.g. to suspend, re-activate or cancel the MLKnowledgeRequest ; or to adjust the description of the desired knowledge.

**REQ-MLKLT-11:** The 3GPP management systemshould have a capability enabling KnowledgeRepo to act as the MLKLT MnS Producer to enable an authorized MnS consumer (e.g. an operator) to manage or control a specific MLKLTJob, e.g. to start, suspend or restart the MLKLTJob; or to adjust the transfer learning conditions or characteristics.

#### 5.1.1.4 Possible solutions

Discovering sharable Knowledge

To discover sharable knowledge:

- The MnS consumer may send a request to the MLKLT MnS producer to provide information on the available sharable knowledge. In other words, the MLKLT MnS producer receives a request to report on the available sharable knowledge.

- The request may be generic or may state a set of criteria which the knowledge should fulfil.

- The request may be referred to as MLKnowledgeInfoRequest.

- The MLKnowledgeInfoRequest must have informational description (Metadata description) of the task and domain related to the required knowledge or given a network problem.

- An ML model or a function containing an ML model may register its available knowledge to a shared knowledge repository, e.g. through a MLKnowledgeRegistration process.

- The MLKnowledgeRegistration must contain informational description (Metadata description) of the task and domain related to the registered knowledge or suitable network problem.

Knowledge sharing and transfer learning

To share knowledge:

- Introduce an IOC for an ML Knowledge request. The MnS consumer may send a request to the MLKLT MnS producer to share a specific kind of knowledge. i.e. the MLKLT MnS producer receives a request to provide sharable knowledge, The request may be referred to as MLKnowledgeRequest.

- Introduce an IOC for an ML transfer learning process or job which is instantiated for any request for transfer learning or ML knowledge transfer. The MLKLT MnS producer instantiates a ML transfer learning process. The process may be referred to as MLKLTJob.

- The MLKLTJob is responsible for adapting the required knowledge into a shareable format with the MLKLT consumer.

- MLKLTJob may be a continuous process where knowledge is shared with the MLKLT consumer frequently to account for updates in the knowledge.

NOTE: It may also be the case that the consumer directly instantiates the MLKLTJob without a separate request.

#### 5.1.1.5 Evaluation

The solution described in clause 5.1.1.4 proposes simple information objects that can enable ML functions to exchange their knowledge to be used towards transfer learning but in a way that enables the vendor specific aspects of the ML models not to be exposed. Therefore, the solution described in clause 5.1.1.4 is a feasible solution to be developed further in the normative specifications.

|  |
| --- |
| **Next modification** |

## 5.5 AI/ML inference

### 5.5.1 Coordination between the ML capabilities

Editor’s Note: The contentin this clause may need to be aligned with the latest version of TS28.105

#### 5.5.1.1 Description

For ML in 5GC or RAN, the ML capabilities in 5GC or RAN may be needed to coordinate with 3GPP management analytics and possibly other aspects in order to improve the overall performance.

Typically, due to the type of the collected data used for model training/inference for 5GC or RAN, the performance of a model in 5GC or RAN may be biased in some respects. On the other hand, the 3GPP management system collects data of longer range and from a wide scope of RAN nodes/5GC, which consequently implies that the predictions calculated by the management system will be unbiased towards the overall RAN nodes/5GC. However, 3GPP management system predictions may lack insight of patterns related to specific node behaviour or finer granularity of time. Hence with coordination or alignment of the ML capability among 5GC/RAN and 3GPP management system, the overall performance may be improved.

To enable the coordination between the ML capabilities, the configuration (e.g. a triggering condition, i.e. when a result is needed from the RAN analytics to MDA) may be needed. On the other hand, the result of the coordination may be communicated towards the consumer(s) and hence there is a need to enhance the reporting in order to capture the deviation of predictions (and/or the related context) so the consumer can gain a better understanding regarding the coordination of ML capabilities.

#### 5.5.1.2 Use cases

##### 5.5.1.2.1 Alignment of the ML capability between 5GC/RAN and 3GPP management system

Generally, the typical data from 5GS data is measurements (PM, KPI), which may be modeled as "time series data" and the analytics of "time series data" is normally to learn the seasonality, trend, etc., patterns. Different types of seasonality patterns exist, e.g. daily, weekly, monthly, seasonally, annually, etc. A RAN node collects finer granularity data for short duration. The finer seasonality pattern will be well captured in a timely manner, while the 3GPP management system with longer range of data (which is more aggregated) will more accurately capture the higher level of seasonality patterns. Hence, combing the analytics results from RAN, 5G core and 3GPP management system or between RAN and 5G core may improve the accuracy for overall predictions.

On another matter, the data collected from one RAN or 5G core node would tend to be biased for that specific RAN node or NF, which implies that the prediction with the data learned and inferred will tend to be biased for that specific node. On the other hand, the 3GPP management system collects data of longer range and from a large amount of different RAN nodes or NFs, which consequently implies that the prediction will be unbiased towards the overall RAN nodes or 5G core area. Hence combining the results from both the RAN or 5G core and 3GPP management system, or between NWDAF and RAN may also improve the overall predictions accuracy (and mitigate the bias).

On the other hand, from a single ML capability perspective, one ML model may support one (or a set of) specific type of inference(s) capability. One ML inference function may employ two or more existing ML models to perform more sophisticated inference(s) instead of developing a new specific ML model. The involved ML models may be composed together in a coordinated manner, for which a coordination of ML capabilities would be needed.

#### 5.5.1.3 Potential requirements

**REQ-AIML\_COORD-01:** 3GPP management system should have the capability to allow an authorized consumer to request a coordination of different ML capabilities, e.g., between MDAF and RAN function, MDAF and NWDAF, or among MDAF(s) or among MLEntity(ies).

**REQ-AIML\_COORD-02:** 3GPP management system should have the capability to report the result of coordination of different ML capabilities.

#### 5.5.1.4 Possible solutions

##### 5.5.1.4.1 Possible solution #1

1) Introduce the information Elements (e.g. instance of IOC or a dataType) for interaction between ML MnS producer and consumer (e.g. the RAN analytics or NWDAF, or entity consuming the RAN analytics or NWDAF) to support coordination of the ML capability between 5GC/RAN and 3GPP management system: MLCapabilityCoordinationRequest and a response informant element MLCapabilityCoordinationResponse. This information element may represent the triggering configuration (or a triggering policy) for predictions coordination from two ML capabilities. This information Element may allow an MOI (or MOI using this information element) to be created on the ML (inference) MnS Producer and may contain the following attributes:

- The analytics deviation indicator, such as a threshold (determining that the prediction calculated by a data analytics function exceeds a configurable certain value, corresponding to the prediction available at a different data analytics function, by a "threshold").

- The requested analytics (analytics type name, list of analytics values or output, time intervals, confidence degree, etc.).

- The target objects, e.g. gNBs, and the related characteristics.

- Area of interest, geographical area or TA.

2) MLCapabilityCoordinationResponse- this information element may represent the response indicating the analytics or data obtained according to the MLCapabilityCoordinationRequest. This information Element may be created by the ML MnS (inference) producer towards the MnS consumer and includes output analytics which can be statistics or predictions. The MLCapabilityCoordinationResponse and MLCapabilityCoordinationRequest may also be data attribute in analytic request and response or analytics report.

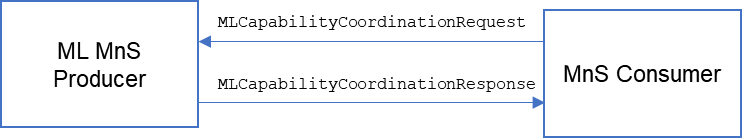


Figure 5.5.1.4.1-1: Interaction between ML MnS producer and consumer  
to support coordination of the ML capability

#### 5.5.1.5 Evaluation

The solution described in clause 5.5.1.4.1 proposes simple information elements and procedure that may enable MnS Producer to trigger configuration to be used for analytic coordination. This NRM based solution reuses the existing provisioning MnS Operations and notifications for control and reporting. Therefore, the solution described in clause 5.5.1.4.1 is a feasible solution to be developed further in the normative specifications.

### 5.1.1 ML inference emulation

Editor’s Note: The contentin this clause may need to be aligned with the latest version of TS28.105

#### 5.5.3.1 Description

A trained ML model can be used for inference within the stated scope e.g. on a managed function or in a management function. Accordingly, there may be an AI/ML inference MnS producer that is responsible for executing the inference.

#### 5.5.3.2 Use cases

##### 5.5.3.2.1 AI/ML inference emulation

After an ML model is trained, validation is done to ensure the training process is completed successfully. Typically, validation is done by preserving part of the training data set and using it after training to check whether the ML model has been trained correctly or not. However, even after the ML model is validated during development, inference emulation is necessary to check if the ML model containing the ML model is working correctly under certain runtime context or using certain inference emulation data set. In principle, the two operations are similar on a functional level, where both of them check the ML performance against given context or data to ensure the ML functionality is functioning correctly. But inference emulation involves interaction with third parties, e.g. the operators who use the ML model or third-party systems that may rely on the results computed by the ML model. For these reasons, it is necessary to support inference emulation, specifically to support means:

- For a given MnS consumer to request for a specific AI/ML capability to be executed in ML inference emulator environment.

- For a given MnS consumer to request a specific ML inference emulator to execute a given AI/ML capability.

- For a managed function to act as a ML inference emulator and execute AI/ML capabilities in a controlled way.

The network or its management system needs to have the capabilities and provide the services needed to enable the MnS consumer to request inference emulation and receive feedback on the inference emulation of a specific ML model or of an application or function that contains an ML model.

##### 5.5.3.2.2 Managing ML inference emulation

The 3GPP management system may have resources for multiple emulation environments to be used depending on need. These may include simulation environments, a digital twin of the network, a test network or the real network under curtain constrained conditions, e.g. for a selected set of UEs. The multiple emulation environments may represent different levels of trust that the operator or management system has in the ML model or AI/ML inference functions. Correspondingly, 3GPP management system needs to have means and method for Orchestrating the inference emulation i.e. say called the inference emulation orchestrator or inference emulation function. Accordingly:

- the emulation progression process involves choosing the right type and instance of an emulation environment to which an ML model, AI/ML inference function or the action thereof may be tested depending on the needs of the function to be tested and the available emulation environments and their resources;

- the emulation process may also involve executing the ML model, AI/ML inference function or its action on the real network but in a controlled fashion, e.g. only within certain hours or only on cells with a particular kind of load or only on cells in a particular area or in limited subscriber groups.

Relatedly, the actions taken by the inference emulation function may include:

- Controlling the allowed parameter space/ranges of the parameters optimized by the ML model or AI/ML inference function depending on the emulation environment to which the ML model, AI/ML inference function or the actions are being executed.

- Adjusting the parameter space in consideration of the observed behaviour of the ML model or AI/ML inference function.

- Deploying the actions of the ML model or AI/ML inference function on a selected emulation environment or on the real network.

- Blocking the ML model or AI/ML inference function from being used on the network.

#### 5.5.3.3 Potential requirements

**REQ-AI/ML\_EMUL-1:** The MnS producer for AI/ML inference emulation should have a capability to allow an authorized MnS consumer to query the available emulation environment(s).

**REQ-AI/ML\_EMUL-2:** The MnS producer for AI/ML inference emulation should have a capability to inform an authorized MnS consumer of the available emulation environment(s).

**REQ-AI/ML\_EMUL-3:** The MnS producer for AI/ML inference emulation should have a capability to allow an authorized MnS consumer to request an ML inference emulation for a specific ML model or models.

**REQ-AI/ML\_EMUL-4:** The MnS producer for AI/ML inference emulation should have a capability to allow an authorized MnS consumer to request for ML Inference Emulation for a specific ML model using specified data or data with specifically stated characteristics and inference emulation features.

**REQ-AI/ML\_EMUL-5:** The MnS producer for AI/ML inference emulation should have a capability to inform authorized MnS consumer about the status of the emulation of an ML model under emulation.

**REQ-AI/ML\_EMUL-6:** The MnS producer for AI/ML inference emulation should have a capability to allow an authorized MnS consumer (e.g. an operator) to manage or control a specific ML inference emulation process, e.g. to start, suspend or restart the inference emulation; or to adjust the inference emulation conditions or characteristics.

**REQ-AI/ML\_EMUL-7:** The MnS producer for AI/ML inference emulation should have a capability to allow an authorized MnS consumer to request reporting, and receive reports on the progress and outcome of an emulation process.

**REQ-AI/ML\_EMUL-8:** The MnS producer for AI/ML inference emulation should have a capability to allow an authorized MnS consumer to configure an ML model or AI/ML inference function supporting with the level of trust that expresses the degree to which the ML model or AI/ML inference function or the different action thereof have been confirmed as trusted.

**REQ-AI/ML\_EMUL-9:** The MnS producer for AI/ML inference emulation should have a capability to graduate an ML model, AI/ML inference function or the different action thereof through different levels of trust each expressing a different degree to which the ML model, AI/ML inference function or action has been confirmed as trusted.

#### 5.5.3.4 Possible solutions

1) Introduce an IOC with the properties of the ML inference emulation function. This may be termed as an MLInferenceEmulationFunction to be name-contained in either a Subnetwork, a ManagedFunction or a ManagementFunction. The MLInferenceEmulationFunction may be a separate function or may be name-contained in an inference function.

This IOC contains attributes including the following:

- list of the ML models which can be emulated and possibly in which emulation environments;

- indication of progression of the MLModel or AI/ML inference function that is under emulation to indicate the degree to which the inference has been emulated and trusted;

- different characteristics for which different emulations may be supported;

- a hierarchy for different emulation environments, e.g. an emulator that uses only a test network vs. an emulator that activates the actions in the real network at specified time such as the maintenance window.

2) Introduce an IOC representing an available emulation environment, e.g. a new IOC named as AvailableEmulationEnvironment, or EmulationSubNetwork, or the existing SubNetwork IOC with an attribute indicating it is a subnetwork used for emulation.

The instance of this IOC is created by the MnS producer to allow the consumer to query, or be informed of, the information of the available emulation environments. This IOC contains the following properties (e.g. the subordinated IOC or attributes):

- inference functions or ML models under emulation.

3) Introduce an IOC for the request for ML inference emulation which shall capture the consumer's requirements for inference emulation. This may be named as an MLInferenceEmulationRequest to be name-contained by the MLInferenceEmulationFunction.

The MLInferenceEmulationRequest shall be associated with at least 1 MLModel for which the inference emulation is being executed.

This IOC contains attributes including the following:

- identifier of the ML models requested for emulation;

- identifier of the selected emulation environment;

- time window for the emulation.

Each MLInferenceEmulationRequest may have a RequestStatus field that is used to track the status of the specific MLInferenceEmulationRequest or the associated MLInferenceEmulationProcess. The RequestStatus is an enumeration with the possible values as: "Pending" if no action has been taken for the request; "Triggered" when an MLInferenceEmulationProcess has been instantiated; "Suspended" when the request or its job has been suspended by MnS consumer or producer and "Served" when the job has run to completion.

4) Introduce an IOC for the process of ML inference emulation from the objects for inference emulation shall be instantiated. This may be named as an MLInferenceEmulationProcess to be name-contained by the MLInferenceEmulationfunction.

The MLInferenceEmulationProcess shall be associated with at least one MLModel for which the inference emulation is being executed.

This IOC contains the following attributes:

- progress indicator;

- identifier of the corresponding emulation request.

5) Introduce a report on ML inference emulation, which may provide reporting on ML emulation for one or more MLInferenceEmulationRequests or MLInferenceEmulationProcesses. This report may be equivalent to the inference report, and:

- the ML inference emulation report may indicate the emulation environments in which the ML model has been emulated and passed;

- the ML inference emulation report may also include the specific performance metrics for that emulation environments.

May also introduce the IOCs and datatypes for request, process and report on ML Inference and then or update these IOCs with the features of the ML Inference Emulation as introduced above.

6) The IOCs, attributes and performance measurements for the solutions of performance evaluation for AI/ML inference for inference phase, as described in clause 5.2.6.4, which can be reused for monitoring the inference performance of the ML models during the emulation.

7) The IOCs and attributes for configuration management of 5G system, as defined in 3GPP TS 28.541 [20], which can be reused for configuring the emulation environment.

#### 5.5.3.5 Evaluation

The solution described in clause 5.5.3.4 reuses the existing provisioning MnS operations and notifications in combination with extensions of the NRM. Requests for inference emulation for a given ML model may be instantiated using provisioning management service implemented via CRUD (Create, Read, Update, Delete) operations on the request objects. The solution provides the flexibility to allow any function that can execute n ML model to be the MnS producer for ML inference emulation, e.g. an inference function or a generic sandbox function.

Therefore, the solution described in clause 5.5.3.4 is a feasible solution to be developed further in the normative specifications.

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