**3GPP TSG-SA5 Meeting #158 *S5-246730***

Orlando, USA, 18 - 22 November 2024

**Source: ZTE Corporation**

**Title: R19 pCR TR 28.858 Update the Reinforcement Learning Use Case**

**Document for: Approval**

**Agenda Item: 6.19.1**

# 1 Decision/action requested

***In this box give a very clear / short /concise statement of what is wanted.***

# 2 References

[1] 3GPP TR 28.858: " Study on Artificial Intelligence / Machine Learning (AI/ML) management Phase 2"

# 3 Rationale

Based on the discussion in last meeting, the RL diagram should be updated with the following aspects:

1. Reward can be set by the third party (e.g., an ML model to generating reward), i.e., reward may not be obtained from the environment defectively. This contribution proposes to add a block for “Reward” to avoid the implication that the Reward can only be set by the environment.
2. The RL agent should be an entity capable of interacting with the environment. It is therefore misleading to equate the RL agent with an ML model. The RL agent should be understood as the entities where the ML model for RL is deployed to generate inferences and subsequently execute actions, i.e., the AIMLInferenceFunction.
3. RL Agent and RL Environment cannot be viewed and discussed in isolation from each other, as the RL Agent can also be considered a component of the RL environment. This interdependence is particularly evident in scenarios where ML models are loaded into a gNB serving as the RL agent within a specific geographical area. In such cases, not only the entities (e.g., UE, adjacent gNBs) impacted by actions of the RL Agent, but the gNB itself acting as the RL Agent should be considered as part of the RL environment. This holistic approach ensures a comprehensive understanding of the dynamic interactions and feedback loops that shape the RL process. Therefore, clarification is needed to enhance the description of RL environment.

# 4 Detailed proposal

### 5.1.7 Management of Reinforcement Learning

#### 5.1.7.1 Description

Reinforcement Learning (RL) is a type of machine learning where an agent learns to make decisions by taking actions in an environment to achieve a goal. The agent learns from the consequences of its actions, rather than from being explicitly taught and it selects its actions based on its past experiences (exploitation) and also by new choices (exploration), which is essentially a trial-and-error approach. In RL process, an agent interacts with the environment in discrete time steps. At each time step, the RL agent receives the current state of the environment and selects actions to be taken. The environment provides the new state in turn and the RL agent will receive the reward based on the state of the environment. For example, RL is applied for coverage problem analysis use case, the RL agent actions are used to optimize the coverage problem and the RL environment can be the simulation environment. The actions can be changes to the values of network adjustable parameters (e.g. change the transmission power of the NR sector carrier, see TS 28.104 [3]). The states can be the network PMs/KPIs like RSRP distribution etc. The reward can be the score of an RL performance metric to evaluate the PMs/KPIs. The goal of the agent is to learn a policy, which tells it what action to take under what circumstances, that maximizes the sum of rewards. The main advantage of this approach is in the ability to automatically adapt to the characteristics of the environment, making it suitable approach for handling dynamic environments such as mobile networks.



#### **Figure 5.1.7.1-1: Reinforcement Learning in Domain Management Function**5.1.7.2 Use cases

##### 5.1.7.2.1 Exploration in Reinforcement Learning

Reinforcement Learning (RL) has the ability to learn and adapt itself to dynamic environments and thus finds the near optimal solution to the problem. This makes the RL-based approaches very interesting for applications in the mobile networks. However, the potential negative impact to the mobile network caused by RL is still the main drawback. In particular, during the exploration step performing trials and learning from errors may have an impact on the operational network and may result in unsafe operations causing network performance degradations. Therefore, the exploration step in RL needs to be under a controlled environment or a stable training configuration (higher exploration rate, i.e., more exploration rounds, always results in a higher training efficiency with a more severe network performance degradation) that is not supposed to violate system performance requirements. If the RL agent behaves in an unexpected manner, there needs to be a set of fall-back actions in place, e.g., to switch from RL-based solution to non-RL-based solution, to fall back to last discrete time step, and to terminate the RL process.

For RL management, the MnS consumer can query the ML Training MnS producer to discover if RL is supported. When RL is supported, a consumer may want to provide a scope (e.g., geographical area, time window) that can aid the producer to select/create the environment when performing RL. The environment scope may include information of the entities where the ML model are deployed acting as RL agent. In the event RL is supported, the consumer may also want to state their preference for environment type for RL during training i.e. simulated environment or real network. When the real network is preferred by the MnS consumer, the consumer can provide network performance requirements (e.g. lower bound threshold, acceptable range, maximum performance deterioration Rate, etc.) of performing ML training of RL, so as to make the MnS producer adapt the training configurations to meet the network performance requirements.

NOTE: Support for both environment types can be considered optional in the RL training.

##### 5.1.7.2.2 Training Conflict in Reinforcement Learning

The training process of RL is realized by the actions with their impacts to the RL environment. In the online training, if there are multiple RL training processes (of multiple ML models for different AI/ML inference functions) sharing a same RL environment, simultaneously, they will interfere with each other, which may cause the training conflict. To be specific, if multiple ML training processes of RL have conflicts, their agents may make actions at the same time, then the state of the RL environment will be affected by these actions. This kind of training error will result in the performance loss of the trained ML models, even cause the training process difficult to converge.

For example, for the training processes whose related inference functions are load balancing optimization (LBO) and mobility robustness optimization (MRO), their RL environment states both include the cell load while the actions from the two RL agents both include the cell individual offset (CIO) adjustment to optimize the handover decisions. If these two RL training processes are processed at the same time with a same environment for an MnS producer, the loads of the adjacent cells will be affected by both training processes, thus causing the training conflict.

For control the conflict of reinforcement learning, the MnS consumer should know whether there are conflicts during the RL training. The producer should determine the conflict and provide the training conflict indication in RL training to authorized MnS consumer. The MnS consumer may specify the conflict resolution requirements to producer. For example, the consumer can cancel/suspend some training processes or configure conflict resolution requirements in advance.

#### 5.1.7.3 Potential Requirements

**REQ-RL\_MGMT-01:** The ML training MnS producer should have a capability allowing an authorized MnS consumer to query if RL training is supported.

**REQ-RL\_MGMT-02:** The ML training MnS producer should have a capability allowing an authorized MnS consumer to specify the preferred RL environment type.

**REQ-RL\_MGMT-03:** The ML training MnS producer should have a capability to allow an authorized MnS consumer to specify the preferred RL environment scope.

**REQ-RL\_MGMT-04:** The ML training MnS producer should have a capability to allow an authorized MnS consumer to specify RL configuration scope for exploration in reinforcement learning.

**REQ-RL\_MGMT-05:** The ML training MnS producer should have a capability allowing an authorized MnS consumer to provide network performance requirements of performing RL training.

**REQ-RL\_MGMT-06:** The ML training MnS producer should have a capability to provide the training conflict indication during RL online training to authorized MnS consumer.

**REQ-RL\_MGMT-07:** The ML training MnS producer should have a capability allowing an authorized MnS consumer to specify the conflict resolution requirements during RL online training.

#### 5.1.7.4 Possible solutions

##### 5.1.7.4.1 Possible solution #1

This solution proposes to enhance the existing MLTrainingRequest IOC and MLTrainingReport IOC to allow the RL consumer to specify RL requirement including environment selection and performance requirements.

**Enhancement on MLTrainingFunction IOC**:

Extend the existing IOC MLTrainingFunction with an attribute whose datatype is the supported learning technology including RL training, indicating the supported RL environment type (e.g. RL on simulation environments, RL on the real network) of the MnS producer.

**Enhancement on MLTrainingRequest IOC**: Introduce a new <<datatype>> “RLRequirement” to the MLTrainingRequest IOC, which may include following attributes:

* RLEnviromentType, represents required RL environment type. The allowed values may be “Online”, “Offline” representing real-network and simulation network respectively.
* RLEnviromentScope, represents RL environment scope, which may be a RL geographical area and/or network nodes, time window and required network load. The scope dose not only include the entities directly involved in RL process, but also includes performance of other entities impacted by RL agent actions. RLConfigurationScope, e.g., the range of actions that the agent is allowed to take.
* RLPerformanceRequirements, represents the attribute of the network performance requirements for online ML training, which indicates the tolerable network performance degradation (e.g. minimum or maximum performance threshold, acceptable range or maximum performance deterioration ratio). When the network performance is within the range, the RL training process can be continued. Otherwise, fall back actions can be determined by the producer. This attribute can also use to reflect the reward setting preference from the MnS Consumer.

##### 5.1.7.4.2 Possible solution #2

The existing MLTrainingProcess IOC or MLTrainingReport should be enhanced to support the management of reinforcement training conflict.

When training process detect the conflicts, it suspend the process and generate a conflict information.

* Option1: extend MLTrainingProcess IOC with this conflict information. The producer write the conflict information and notify to the consumer
* Option2: extend MLTrainingReport IOC with this conflict information. The producer write the conflict information into the training report and notify the report to the consumer

The MnS consumer can get this attribute from multiple MLTrainingProcess MOIs or MLTrainingReport MOI then the cosumer can determine the conflict training processes. The MnS consumer can modify the existing cancelprocess or suspendprocess attribute to manage the training processes, so as to avoid training conflicts.

#### 5.1.7.5 Evaluation

The solution described in clause 5.1.6.4 proposes the addition of a new attribute to the MLTrainingFunction IOC to indicate the supported RL training and new attributes to the MLTrainingRequest IOC to enable the MnS consumer to indicate the preferred RL environment type and RL environment scope. Therefore, the solution described in clause 5.1.6.4 is a feasible solution to be developed further in the normative specifications.