**3GPP TSG-SA5 Meeting #158 *S5-247065***

Orlando, USA, 18 - 22 November 2024

**Source: Ericsson India Private Limited (TSDSI), NEC, ZTE**

**Title: PCR TR 28.858 Correct Fine-Tunning term definition**

**Document for:Approval**

**Agenda Item:** **FS\_AIML\_MGT\_Ph2**

# 1 Decision/action requested

***approval***

# 2 References

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

[2] 3GPP TS 28.105: Artificial Intelligence / Machine Learning (AI/ML) management".

# 3 Rationale

Update Description Use case , Evaluation and Conclusions for ML Fine tuning as well as adding Possible solution #2 to clause 5.1.4.X, which ensures compatibility with existing standards as well introduces a generic attribute TrainingType. Add evaluation description in clause 5.1.4.5.

Add conclusion for Pre-Training in clause 6.

# 4 Detailed proposal

|  |
| --- |
| **1st Change** |

## 3.1 Terms

For the purposes of the present document, the terms given in 3GPP TR 21.905 [1], TS 28.105 [2] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in 3GPP TR 21.905 [1] TS 28.105 [2].

### 5.1.3 ML Fine-tuning

#### 5.1.3.1 Description

ML fine-tuning refers to process of training an already trained ML model (e.g. a pre-trained ML model) with a changed scope.

In contrasts, re-training of an ML Model is not based on a specific type of inference, while fine-tuning of a pre-trained model means that the ML model will be trained with focus on some specific types of inference.

The ML Model training process defined in TS 28.105 includes provision for fine-tuning pre-trained models to ~~specialize the model for specific task~~ narrow down the scope of inference type of an model for a specific task.

#### 5.1.3.2 Use cases

##### 5.1.3.2.1 ML fine-tuning for a pre-trained ML model

A pre-trained ML model may be initially trained on a dataset that supports various types of inference, i.e.. However, fine-tuning allows the model to narrow down the scope of a inference type of a model.~~for a specific type/types of inference.~~

NOTE: It is for FFS whether fine-tuning process could be applied for the same producer who pre-trained the model or for another producer.

#### 5.1.3.3 Potential requirements

**REQ-ML\_ TRAIN-MLFT-1:** The ML training MnS producer shall have a capability to enable an authorized consumer to request the fine-tuning of a pre-trained ML model.

Note: It is for further discussion if the same requirements in clause 6.2b.3 "Requirements for ML model training" in TS 28.105 [2] related to training function also applies for fine-tuning.

#### 5.1.3.4 Possible solutions#1

The exsting MLModel IOC and MLTrainingRequest IOC can be enhanced to support fine-tuning.

**Enhancement Aspects on MLTrainingRequest IOC**: Since the exsting MLTrainingRequest IOC cannot support consumer requested ML fine-tuning, both following enhancements are possible.

* FineTuningIndication, indicates the consumer requested ML fine-tuning. The support qualifier should be CM, which means this attribute should be present when the MLTrainingRequest MOI represents the request for ML Model Fine-tuning. Besides, change the support qualifier of “aIMLInferenceName” to CM, which means that this attribute should be present when the MLTrainingRequest MOI does not represent the request for ML Model Fine-tuning.
* PotentialInferenceScope, indicates a set of types of inference, which may include a list of supported aIMLInferenceName.

**Enhancement Aspects on MLModel IOC**: Since fine-tuning is aiming at a specific inference type for a pre-trained model, the AIML capabilities would be narrowed which has the best performance. It is proposed to introduce following attributes:

* FinetuningModelIndication, indicates that the ML Model is a Finetuning Model. Besides, change the support qualifier of “aIMLInferenceName” to CM, where the condition is that the ML Model requested to train is not a pre-training ML Model.
* PotentialInferenceScope, indicates a set of types of inference, which may include a list of supported aIMLInferenceName.

#### 5.1.3.4.X.2 Possible solutions#2

The existing ML training procedures in TS 28.105 [2] already provide all the necessary mechanisms for initial training and re-training of an ML model. The existing MLTrainingRequest IOC may be used to support fine-tuning, similarly to how it is used for the re-training procedure.

The key distinction between retraining and fine-tuning is that retraining results in a new version of the same ML model, while fine-tuning produces results in a new ML model, which has a different mlModelId, MLModelRef, and aIMLInferenceName than the ML model that was requested to be fine-tuned.

To enable fine-tuning within the current framework the following enhancements are need:

Enhancement onMLTrainingRequest IOC:

* mLTrainingType, indicates the type of training, e.g. pre-training, fine-tunning etc (see clause [1] 5.1.2.4).

#### 5.1.3.5 Evaluation

Possible solution #1 described in clause 5.1.3.4 proposes to add “FineTuningIndication” and “PotentialInferenceScope” to the MLTrainingRequest IOC, and add “FinetuningModelIndication” and “PotentialInferenceScope” to the MLModel IOC. Only simple enhancement is introduced to support fine-tuning, which is feasible to be developed further in the normative specifications.

Compared to possible solution 1#, possible solution 2# offers a simpler and flexible approach to fine-tuning an ML model by reusing existing attributes, IOCs and mechanisms in TS 28.105 [2]. It also aligns with the use case for pre-training in clause [2] 5.1.2.4.2 by introducing the generic attribute mLTrainingType to indicate the type of training requested by the consumer.

In Solution #1, the addition of PotentialInferenceScope for the ML model is unnecessary, base of the fact that when the model is fine-tuned, it results in a new ML model with a different mlModelId, MLModelRef, and aIMLInferenceName. Further discussions are needed in normative work in regards of attributes aIMLInferenceName properties values.

Solution #2 minimizes complexity and is the preferred solution over Solution #1.



|  |
| --- |
| **End of Change** |