|  |
| --- |
|  |

|  |
| --- |
| Input Contribution |
| Meeting ID\* | RDM#56 |
| Title:\* | Key issue on AI/ML model management |
| Source:\* | JaeSeung Song, Sejong University, jssong@sejong.ac.kr  |
| Date:\* | 2022-09-28 |
| Input related to\* | TR-0068 (Rel-5) |
| Intended purpose ofdocument:\* | [x]  Decision[ ]  Discussion[ ]  Information[ ]  Other <specify> |
| Impacted other TS/TR(s) |  |
| Decision requested or recommendation:\* | Agree for inclusion in TR-0068 |
| Template Version: January 2017 (Do not modify) |

**oneM2M Notice**

The document to which this cover statement is attached is submitted to oneM2M. Participation in, or attendance at, any activity of oneM2M, constitutes acceptance of and agreement to be bound by terms of the Working Procedures and the Partnership Agreement, including the Intellectual Property Rights (IPR) Principles Governing oneM2M Work found in Annex 1 of the Partnership Agreement.

# Introduction

This contribution introduces a key issue on AI/ML model management.

### -----------------------Start of change 1-------------------------------------------

## 8.x Key Issue & Possible Concept x – AI/ML model management

### 8.x.1 Key Issue

The artificial intelligence service basically requires data for learning, an algorithm to make an accurate model, and parameter information for performing the algorithm. In the existing IoT platform, many datasets that can be used for AI services are stored and managed, but the functions necessary to build and use AI models are not provided. Therefore, most AI services were developed and provided on a separate, independent platform. In order to provide common service functions required for AI services through the IoT platform, various information to build and manage AI/ML models should be stored in the IoT platform.

However, the oneM2M platform does not provide any features supporting AI/ML services. If the oneM2M platform supports AI/ML-related functions, then AI/ML developers can use the AI-enabled oneM2M platform to build their required services.

### 8.x.2 Possible Solution

Possible solution should support the following requirements (see Clause 7.3):

1. The oneM2M System shall be able to manage structured and unstructured data for training, for example, preprocessing data, describing data and inferring meaning.
2. The oneM2M System shall be able to update trained AI/ML model according to continuous measuring data e.g., location, time series and historical data.
3. The oneM2M System shall be able to provide a classification function (e.g., split data into two parts, training and validating) in supervised Machine Learning.

If oneM2M systems support such functions, an AI application can create a resource(s) to build a model for its prediction as follows:

* Assumption 1: an oneM2M platform holds all the data for training and has data for prediction
* Assumption 2: an oneM2M platform knows a list of ML algorithms to perform
* An AI/ML application creates a resource to build an AI/ML model to solve or support a specific domain application.
* The AI/ML application triggers to build the model.
* The AI/ML application downloads the trained model and performs prediction.



Figure 8.x-1: An example structure of [*mlManagement*] resource

This can be done by introducing a set of attributes to a new resource called <*mlManagement*>) to hold the information required to manage AI/ML functions in the oneM2M platform. The required attributes are as follows:

* *datasetTrain*: list of resources storing train data
* *datasetValidation*: list of resources storing validation data
* *datasetTest*: list of resources for testing a model
* *datasetRatio*: ratio of ML dataset (for example three tuple of percentage)
* *selectedModel*: an ML algorithm that represents the model to perform
* *modelParameters*: parameters used by the selected algorithm
* *trainedModel*: the result model (e.g., executable software) after training and validation
* *trigerBuildModel*: a triggering value to start building a model (assumption is to have proper values on
*datasetTrain*, *datasetValication*, *datasetTest*, *dataSetRatio*, and *selectedModel*)

Figure 8.x-2 shows a high-level procedure that enables AI/ML features using the <mlManagement> resource. This allows an AI application to create a resource that manages AI/ML model and perform AI/ML function as a common service via the oneM2M platform.

First, an AI/ML application (oneM2M application) creates a resource to perform and manage AI/ML function. The AI/ML application configures the <*mlManagement*> resource with proper information to build and manage an AI/ML model. The application also subscribes to the <*mlManagement*> resource to be notified when an AI/ML model is built or updated. The AI/ML application requests to perform training to build a model via setting the *triggerBuildModel* attribute. The oneM2M platform performs AI-enabled CSF to build a model based on the given attributes in the <mlManagement> resource. For example, the CSF uses training dataset specified in the *datasetTrain* attribute to train a model using an AI/ML algorithm specified in the *selectedModel* attribute. The trained model is then validated and tested via dataset stored resources in the *datasetValication* and *datasetTest* attributes, respectively. The AI-enabled CSF then builds a model that can be used by the AI/ML application. This update is notified to the Application for the AI/ML service.



Figure 8.x-2: An example procedure to enable AI/ML feature to oneM2M using the [*mlManagement*] resource

### -----------------------End of change 1-------------------------------------------