

ETSI TS 128 105 V17.0.0 (2022-07)



**5G;
Management and orchestration;
Artificial Intelligence/ Machine Learning (AI/ML) management
(3GPP TS 28.105 version 17.0.0 Release 17)**



Reference

DTS/TSGS-0528105vh00

Keywords

5G

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Important notice

The present document can be downloaded from:

<http://www.etsi.org/standards-search>

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

Users of the present document should be aware that the document may be subject to revision or change of status.

Information on the current status of this and other ETSI documents is available at

<https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx>

If you find errors in the present document, please send your comment to one of the following services:

<https://portal.etsi.org/People/CommitteeSupportStaff.aspx>

If you find a security vulnerability in the present document, please report it through our
Coordinated Vulnerability Disclosure Program:

<https://www.etsi.org/standards/coordinated-vulnerability-disclosure>

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.

No recommendation as to products and services or vendors is made or should be implied.

No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.

In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

Copyright Notification

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.

The content of the PDF version shall not be modified without the written authorization of ETSI.

The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2022.
All rights reserved.

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: "*Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards*", which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<https://ipr.etsi.org/>).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM®** and the GSM logo are trademarks registered and owned by the GSM Association.

Legal Notice

This Technical Specification (TS) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities. These shall be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between 3GPP and ETSI identities can be found under <http://webapp.etsi.org/key/queryform.asp>.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the [ETSI Drafting Rules](#) (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Contents

Intellectual Property Rights	2
Legal Notice	2
Modal verbs terminology.....	2
Foreword.....	5
1 Scope	7
2 References	7
3 Definitions of terms, symbols and abbreviations	8
3.1 Terms.....	8
3.2 Symbols.....	8
3.3 Abbreviations	8
4 Concepts and overview	8
4.1 Overview	8
5 AI/ML management functionality and service framework	9
5.1 Functionality and service framework for AI/ML training	9
6 AI/ML management use cases and requirements	10
6.1 General	10
6.2 AI/ML training	10
6.2.1 Description.....	10
6.2.2 Use cases.....	11
6.2.2.1 AI/ML training requested by consumer	11
6.2.2.2 AI/ML training initiated by producer.....	11
6.2.2.3 Selecting AI/ML models and AI/ML-enabled Functions.....	12
6.2.2.4 Managing AI/ML Training Processes	12
6.2.2.5 Handling errors in data and ML decisions	12
6.2.3 Requirements for AI/ML training	13
7 Information model definitions for AI/ML management.....	15
7.1 Imported and associated information entities	15
7.1.1 Imported information entities and local labels.....	15
7.2 Class diagram	15
7.2.1 Relationships.....	15
7.2.2 Inheritance	15
7.3 Class definitions	16
7.3.1 AIMLTrainingFunction.....	16
7.3.1.1 Definition	16
7.3.1.2 Attributes.....	16
7.3.1.3 Attribute constraints	16
7.3.1.4 Notifications.....	16
7.3.2 AIMLTrainingRequest	16
7.3.2.1 Definition	16
7.3.2.2 Attributes.....	17
7.3.2.3 Attribute constraints	17
7.3.2.4 Notifications.....	17
7.3.3 AIMLTrainingReport	17
7.3.3.1 Definition	17
7.3.3.2 Attributes.....	18
7.3.3.3 Attribute constraints	18
7.3.3.4 Notifications.....	18
7.3.4 AIMLTrainingProcess	18
7.3.4.1 Definition	18
7.3.4.2 Attributes.....	19
7.3.4.3 Attribute constraints	20
7.3.4.4 Notifications.....	20

7.4	Data type definitions	20
7.4.1	ModelPerformance <<dataType>>	20
7.4.1.1	Definition	20
7.4.1.2	Attributes.....	20
7.4.1.3	Attribute constraints	20
7.4.1.4	Notifications.....	20
7.4.2	AIMLEntity <<dataType>>	20
7.4.2.1	Definition	20
7.4.2.2	Attributes.....	21
7.4.3.3	Attribute constraints	21
7.4.3.4	Notifications.....	21
7.4.3	AIMLContext <<dataType>>	21
7.4.3.1	Definition	21
7.4.3.2	Attributes.....	21
7.4.3.3	Attribute constraints	21
7.4.3.4	Notifications.....	21
7.5	Attribute definitions	22
7.5.1	Attribute properties	22
7.5.2	Constraints	25
7.6	Common notifications	26
7.6.1	Configuration notifications	26
8	Service components.....	26
8.1	Service components for AI/ML model training MnS	26
9	Solution Set (SS)	26
Annex A (informative): PlantUML source code for NRM class diagrams.....		27
A.1	General	27
A.2	PlantUML code for Figure 7.2.1-1: NRM fragment for AI/ML model training	27
A.3	PlantUML code for Figure 7.2.2-1: Inheritance Hierarchy for AI/ML model training related NRM	
	NRMs	28
Annex B (normative): OpenAPI definition of the AI/ML NRM.....		29
B.1	General	29
B.2	Solution Set (SS) definitions	29
B.2.1	OpenAPI document "AiMlNrm.yaml"	29
Annex C (informative): Change history		33
	History	34

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
 - 1 presented to TSG for information;
 - 2 presented to TSG for approval;
 - 3 or greater indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

In the present document, modal verbs have the following meanings:

- shall** indicates a mandatory requirement to do something
- shall not** indicates an interdiction (prohibition) to do something

The constructions "shall" and "shall not" are confined to the context of normative provisions, and do not appear in Technical Reports.

The constructions "must" and "must not" are not used as substitutes for "shall" and "shall not". Their use is avoided insofar as possible, and they are not used in a normative context except in a direct citation from an external, referenced, non-3GPP document, or so as to maintain continuity of style when extending or modifying the provisions of such a referenced document.

- should** indicates a recommendation to do something
- should not** indicates a recommendation not to do something
- may** indicates permission to do something
- need not** indicates permission not to do something

The construction "may not" is ambiguous and is not used in normative elements. The unambiguous constructions "might not" or "shall not" are used instead, depending upon the meaning intended.

- can** indicates that something is possible
- cannot** indicates that something is impossible

The constructions "can" and "cannot" are not substitutes for "may" and "need not".

- will** indicates that something is certain or expected to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document
- will not** indicates that something is certain or expected not to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document
- might** indicates a likelihood that something will happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

might not indicates a likelihood that something will not happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

In addition:

is (or any other verb in the indicative mood) indicates a statement of fact

is not (or any other negative verb in the indicative mood) indicates a statement of fact

The constructions "is" and "is not" do not indicate requirements.

1 Scope

The present document specifies the Artificial Intelligence / Machine Learning (AI/ML) management capabilities and services for 5GS where AI/ML is used, including management and orchestration (e.g. MDA, see 3GPP TS 28.104 [2]) and 5G networks (e.g. NWDAF, see 3GPP TS 23.288 [3]).

The present document also describes the functionality and service framework for AI/ML management.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

- [1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".
- [2] 3GPP TS 28.104: "Management and orchestration; Management Data Analytics".
- [3] 3GPP TS 23.288: "Architecture enhancements for 5G System (5GS) to support network data analytics services".
- [4] 3GPP TS 28.552: "Management and orchestration; 5G performance measurements".
- [5] 3GPP TS 32.425: "Telecommunication management; Performance Management (PM); Performance measurements Evolved Universal Terrestrial Radio Access Network (E-UTRAN)".
- [6] 3GPP TS 28.554: "Management and orchestration; 5G end to end Key Performance Indicators (KPI)".
- [7] 3GPP TS 32.422: "Telecommunication management; Subscriber and equipment trace; Trace control and configuration management".
- [8] 3GPP TS 32.423: "Telecommunication management; Subscriber and equipment trace; Trace data definition and management".
- [9] 3GPP TS 28.405: "Telecommunication management; Quality of Experience (QoE) measurement collection; Control and configuration".
- [10] 3GPP TS 28.406: "Telecommunication management; Quality of Experience (QoE) measurement collection; Information definition and transport".
- [11] 3GPP TS 28.532: "Management and orchestration; Generic management services".
- [12] 3GPP TS 28.622: "Telecommunication management; Generic Network Resource Model (NRM) Integration Reference Point (IRP); Information Service (IS)".
- [13] 3GPP TS 32.156: "Telecommunication management; Fixed Mobile Convergence (FMC) Model repertoire".
- [14] 3GPP TS 32.160: "Management and orchestration; Management service template".

3 Definitions of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in 3GPP TR 21.905 [1] 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].

AI/ML entity: any entity that is either an AI/ML model or contains an AI/ML model and that can be managed as a single composite entity

AI/ML model: mathematical algorithm that can be "trained" by data and human expert input as examples to replicate a decision an expert would make when provided that same information

AI/ML model training: capabilities of an AI/ML Training Function to take data, run it through an AI/ML model, derive the associated loss and adjust the parameterization of that AI/ML model based on the computed loss

AI/ML training: capabilities and associated end-to-end processes to enable an AI/ML Training Function to train its constituent AI/ML model, e.g. to interact with external parties to collect and format the data required for training the AI/ML model

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in 3GPP TR 21.905 [1].

SBMA	Service Based Management Architecture
------	---------------------------------------

4 Concepts and overview

4.1 Overview

The AI/ML techniques and relevant applications are being increasingly adopted by the wider industries and proved to be successful. These are now being applied to telecommunication industry including mobile networks.

Although AI/ML techniques in general are quite mature nowadays, some of the relevant aspects of the technology are still evolving while new complementary techniques are frequently emerging.

The AI/ML techniques can be generally characterized from different perspectives including the followings:

- **Learning methods**

The learning methods include supervised learning, unsupervised learning and reinforcement learning. Each learning method fits one or more specific category of inference (e.g. prediction), and requires specific type of training data. A brief comparison of these learning methods is provided in table 4.1-1.

Table 4.1-1: Comparison of Learning methods

	Supervised learning	Semi-supervised learning	Unsupervised learning	Reinforcement learning
Category of inference	Regression (numeric), classification	Regression (numeric), classification	Association, Clustering	Reward-based behaviour
Type of training data	Labelled data (Note)	Labelled data (Note), and unlabelled data	Unlabelled data	Not pre-defined
NOTE: The labelled data means the input and output parameters are explicitly labelled for each training data example.				

- **Learning complexity:**
 - As per the learning complexity, there are Machine Learning (i.e. basic learning) and Deep Learning.
- **Learning architecture**
 - Based on the topology and location where the learning tasks take place, the AI/ML can be categorized to centralized learning, distributed learning and federated learning.
- **Learning continuity**
 - From learning continuity perspective, the AI/ML can be offline learning or continual learning.

Artificial Intelligence/Machine Learning (AI/ML) capabilities are used in various domains in 5GS, including management and orchestration (e.g. MDA, see 3GPP TS 28.104 [2]) and 5G networks (e.g. NWDAF, see 3GPP TS 23.288 [3]).

The AI/ML-enabled function in the 5GS uses the AI/ML model for inference.

Each AI/ML technique, depending on the adopted specific characteristics as mentioned above, may be suitable for supporting certain type/category of use case(s) in 5GS.

To enable and facilitate the AI/ML capabilities with the suitable AI/ML techniques in 5GS, the AI/ML model and AI/ML-enabled function (i.e. inference function) need to be managed.

The present document specifies the AI/ML management related capabilities and services, which include the followings:

- AI/ML training.

5 AI/ML management functionality and service framework

5.1 Functionality and service framework for AI/ML training

An AI/ML training Function playing the role of AI/ML training MnS producer, may consume various data for AI/ML training purpose.

As illustrated in Figure 5.1-1 the AI/ML training capability is provided via AI/ML training MnS in the context of SBMA to the authorized consumer(s) by AI/ML training MnS producer.

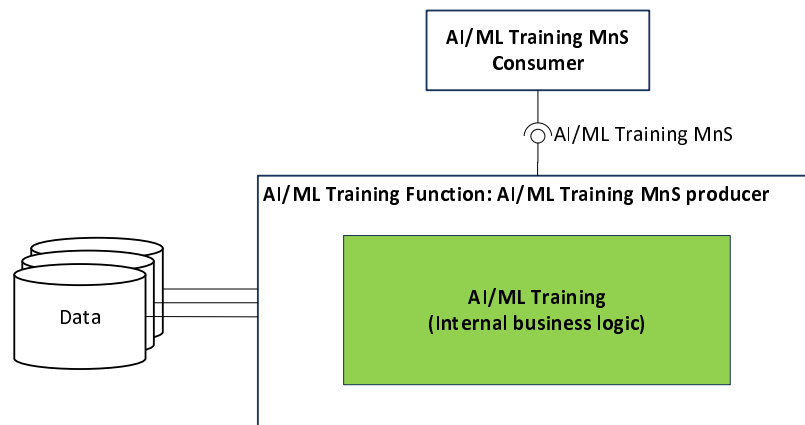


Figure 5.1-1: Functional overview and service framework for AI/ML model training

The internal business logic of AI/ML training leverages the current and historical relevant data, including those listed below to monitor the networks and/or services where relevant to the AI/ML model, prepare the data, trigger and conduct the training:

- Performance Measurements (PM) as per 3GPP TS 28.552 [4], 3GPP TS 32.425 [5] and Key Performance Indicators (KPIs) as per 3GPP TS 28.554 [6].
- Trace/MDT/RLF/RCEF data, as per 3GPP TS 32.422 [7] and 3GPP TS 32.423 [8].
- QoE and service experience data as per 3GPP TS 28.405 [9] and 3GPP TS 28.406 [10].
- Analytics data offered by NWDAF as per 3GPP TS 23.288 [3].
- Alarm information and notifications as per 3GPP TS 28.532 [11].
- CM information and notifications.
- MDA reports from MDA MnS producers as per 3GPP TS 28.104 [2].
- Management data from non-3GPP systems.
- Other data that can be used for training.

6 AI/ML management use cases and requirements

6.1 General

The use cases and requirements for AI/ML management are specified in the following clauses.

6.2 AI/ML training

6.2.1 Description

In operational environment before the AI/ML Entity (i.e. Inference Function) is deployed to conduct inference, it needs to be trained (e.g. by a separate or an external entity to the Inference function).

The AI/ML Entity is trained by the AI/ML training (AIMLT) MnS producer, and the training can be triggered by request(s) from one or more AIMLT MnS consumer(s), or initiated by the AIMLT MnS producer (e.g. as result of model evaluation).

6.2.2 Use cases

6.2.2.1 AI/ML training requested by consumer

The AI/ML training capabilities are provided by an AIMLT MnS producer to one or more consumer(s).

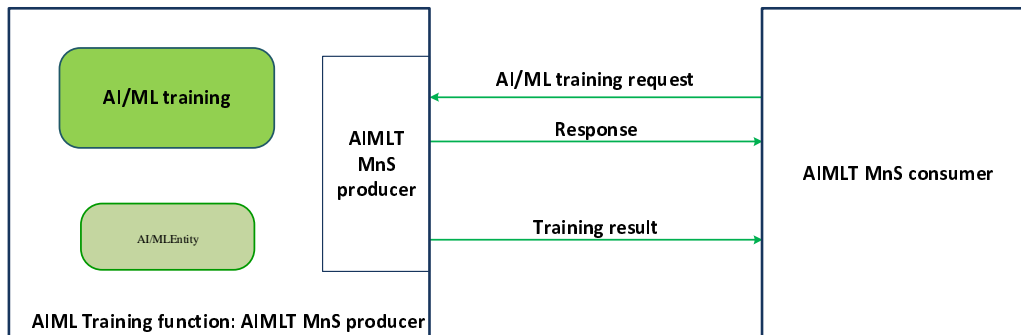


Figure 6.2.2.1-1: AI/ML training requested by AIMLT MnS consumer

The AI/ML training may be triggered by the request(s) from one or more AIMLT MnS consumer(s). To trigger an AI/ML training, the AIMLT MnS consumer requests the AIMLT MnS producer to train the AI/ML model or AI/ML enabled function. In the AI/ML training request, the consumer should specify the inference type which indicates the function or purpose of the AI/ML Entity, e.g. CoverageProblemAnalysis. The AIMLT MnS producer can perform the training according to the designated inference type. The consumer may provide the data source(s) that contain(s) the training data which are considered as inputs candidates for training. To obtain the valid training outcomes, consumers may also designate their requirements for model performance (e.g. accuracy, etc) in the training request.

The AIMLT MnS producer provides a response to the consumer indicating whether the request was accepted.

If the request is accepted, the AIMLT MnS producer decides when to start the AI/ML training with consideration of the request(s) from the consumer(s). Once the training is decided, the producer performs the followings:

- selects the training data, with consideration of the consumer provided candidate training data. Since the training data directly influences the algorithm and performance of the trained AI/ML Entity, the AIMLT MnS producer may examine the consumer's provided training data and decide to select none, some or all of them. In addition, the AIMLT MnS producer may select some other training data that are available;
- trains the AI/ML Entity using the selected training data; and
- provides the training results (including the location of the trained AI/ML Entity, etc.) to the AIMLT MnS consumer(s).

6.2.2.2 AI/ML training initiated by producer

The AI/ML training may be initiated by the AIMLT MnS producer, for instance as a result of performance evaluation of the AI/ML model, based on feedback or new training data received from the consumer, or when new training data which are not from the consumer describing the new network status/events become available.

When the AIMLT MnS producer decides to start the AI/ML training, the producer performs the followings:

- selects the training data;
- trains the AI/ML Entity using the selected training data; and
- provides the training results (including the location of the trained AI/ML Entity, etc.) to the AIMLT MnS consumer(s) who have subscribed to receive the AI/ML training results.

6.2.2.3 Selecting AI/ML models and AI/ML-enabled Functions

For a given machine learning-based use case, different entities that apply the respective ML model or AI/ML enabled function may have different inference requirements and capabilities. For example, one consumer with specific responsibility and wish to have an AI/ML enabled function trained for city central business district where mobile users move at speeds not exceeding 30 km/hr. On the other hand, another consumer for the same use case may support a rural environment and as such wish to have a model fitting that environment. The different consumers need to know the available versions of AI/ML enabled functions and to select the appropriate AI/ML enabled function for their respective conditions.

Besides there is no guarantee that the available AI/ML enabled functions have been trained according to the characteristics that the consumers expect. As such the consumers need to know the conditions for which the models or AI/ML enabled functions have been trained to then enable the consumers to select the models that are best fitted to their conditions.

The models that have been trained may differ in terms of complexity and performance. For example, a generic comprehensive and complex model may have been trained in a cloud-like environment but when such a model cannot be used in the gNB and instead, a less complex model, trained as a derivative of this generic model, could be a better candidate. Moreover, multiple less complex models could be trained with different level of complexity and performance which would then allow different relevant models to be delivered to different network functions depending on operating conditions and performance requirements. The network functions need to know the alternative models available and interactively request and replace them when needed and depending on the observed inference-related constraints and performance.

6.2.2.4 Managing AI/ML Training Processes

This machine learning capability relates to means for managing and controlling AI/ML training processes.

To achieve the desired outcomes of any machine learning relevant use-case, the AI/ML Model applied for such analytics and decision making, needs to be trained with the appropriate network data. The training may be undertaken in managed function or in a management function.

In either case, the network (or the OAM system thereof) not only needs to have the required training capabilities but needs to also have the means to manage the training of the AI/ML models and or AI/ML-enabled functions. The consumers need to be able to interact with the training process, e.g. to suspend or restart the process; and also need to manage and control the requests related to any such training process.

6.2.2.5 Handling errors in data and ML decisions

Traditionally, the machine-learning-enabled Functions (e.g. AIML Entity₁ and AIML Entity) are trained on good quality data, i.e. data that was collected when the network was working correctly, to represent the expected context in which the AIML Entity is meant to operate. Good quality data is void of errors, such as:

- Imprecise measurements, with added noise (such as RSRP, SINR, or QoE estimations).
- Missing values or entire records, e.g. because of communication link failures.
- Records which are communicated with a significant delay (in case of online measurements).

Without errors, an AIML Entity can depend on a few precise inputs, and don't need to exploit the redundancy present in the training data. However, during inference, the AIML Entity is very likely to come across these inconsistencies. When this happens, the AIML Entity shows high error in the inference outputs, even if redundant and uncorrupted data is available from other sources.

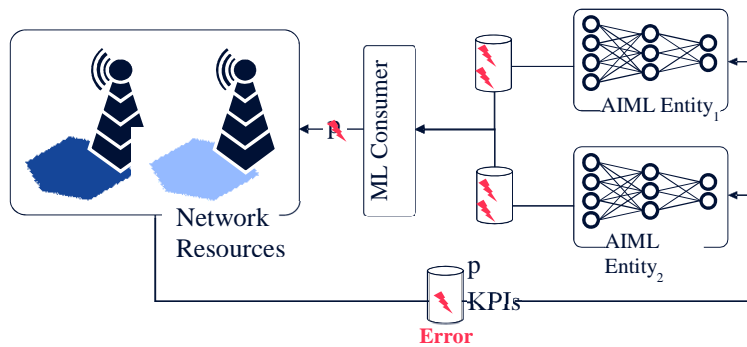


Figure 6.2.2.5-1: The propagation of erroneous information

As such the system needs to account for errors and inconsistencies in the input data and how the consumers of ML decisions should deal with decisions that are made based on such erroneous and inconsistent data. The system should:

- 1) enable functions to undertake the training in a way that prepares the AIML Entity s to deal with the errors, i.e. to identify the errors in the data during training; and
- 2) enable the ML consumers to account for the possibility of erroneous input data into the ML decision makers.

6.2.3 Requirements for AI/ML training

Table 6.2.3-1

Requirement label	Description	Related use case(s)
REQ-AIML_TRAIN-FUN-01	The AIMLT MnS producer shall have a capability allowing the consumer to request AI/ML training.	AI/ML training requested by consumer (clause 6.2.2.1)
REQ- AIML_TRAIN-FUN-02	The AIMLT MnS producer shall have a capability allowing the consumer to specify the data sources containing the candidate training data for AI/ML training.	AI/ML training requested by consumer (clause 6.2.2.1)
REQ- AIML_TRAIN-FUN-03	The AIMLT MnS producer shall have a capability allowing the consumer to specify the inference type of the AI/ML Entity to be trained.	AI/ML training requested by consumer (clause 6.2.2.1)
REQ- AIML_TRAIN-FUN-04	The AIMLT MnS producer shall have a capability to provide the training result (including the location of the trained AI/ML Entity) to the consumer.	AI/ML training requested by consumer (clause 6.2.2.1), and AI/ML training initiated by producer (clause 6.2.2.2)
REQ-AIML_SELECT-01	3GPP management system shall have the capability for authorized consumer to discover the characteristics of available models including the contexts under which each of the models was trained.	Selecting AI/ML models and AI/ML-enabled Functions (clause 6.2.2.3)
REQ-AIML_SELECT-02	3GPP management system shall have the capability to enable an authorized consumer to select an AI/ML model.	Selecting AI/ML models and AI/ML-enabled Functions (clause 6.2.2.3)
REQ-AIML_SELECT-03	3GPP management system shall have the capability to enable an authorized consumer to request for a model to be trained to satisfy the consumer's expectations.	Selecting AI/ML models and AI/ML-enabled Functions (clause 6.2.2.3)
REQ-AIML_SELECT-04	3GPP management system shall have the capability to enable an authorized consumer to request for information and be informed about the available alternative models of differing complexity and performance.	Selecting AI/ML models and AI/ML-enabled Functions (clause 6.2.2.3)

Requirement label	Description	Related use case(s)
REQ-AIML_SELECT--05	3GPP management system shall have the capability to enable an authorized consumer to request one of the known or available alternative models of differing complexity and performance to be used for inference.	Selecting AI/ML models and AI/ML-enabled Functions (clause 6.2.2.3)
REQ-AIML_SELECT-06	The 3GPP management system shall have a capability to provide a selected AI/ML enabled function to the consumer.	Selecting AI/ML models and AI/ML-enabled Functions (clause 6.2.2.3)
REQ-AIML_TRAIN-MGT_01	3GPP management system shall have the capability to enable an authorized consumer to manage and configure one or more requests for the training of specific AI/ML models or AI/ML enabled functions, e.g. to modify the characteristics of the request or to delete a request.	Managing AI/ML Training Processes (clause 6.2.2.4)
REQ-AIML_TRAIN-MGT_02	3GPP management system shall have the capability to enable an authorized consumer to manage and configure one or more training processes, e.g. to start, suspend or restart the training; or to adjust the training conditions and/or characteristics.	Managing AI/ML Training Processes (clause 6.2.2.4)
REQ-AIML_TRAIN-MGT_03	3GPP management system shall have the capability to enable an authorized consumer (e.g. the function/entity different from the function that generated a request for AI/ML enabled function training) to request for a report on the outcomes of a specific training instance.	Managing AI/ML Training Processes (clause 6.2.2.4)
REQ-AIML_TRAIN-MGT_04	3GPP management system shall have the capability to enable an authorized consumer to define the reporting characteristics related to a specific training request or training instance.	Managing AI/ML Training Processes (clause 6.2.2.4)
REQ-AIML_TRAIN-MGT_05	3GPP management system shall have the capability to enable the AI/ML Training function to report to any authorized consumer about specific ML Training process and/or report about the outcomes of any such ML Training process.	Managing AI/ML Training Processes (clause 6.2.2.4)
REQ-ML_ERROR_01	The 3GPP management system shall enable an authorized consumer of data services (e.g. an ML-enabled function) to request from a producer of data services a Value Quality Score of the data, which is the numerical value that represents the dependability/quality of a given observation and measurement type.	Handling errors in data and ML decisions (clause 6.2.2.5)
REQ-ML_ERROR_02	The 3GPP management system shall enable an authorized consumer of ML decisions (e.g. a controller) to request ML decision confidence score which is the numerical value that represents the dependability/quality of a given decision generated by the ML-based function.	Handling errors in data and ML decisions (clause 6.2.2.5)
REQ-ML_ERROR_03	The 3GPP management system shall enable a producer of data services (e.g. a gNB) to provide to an authorized consumer (e.g. an ML-enabled function) a Value Quality Score of the data, which is the numerical value that represents the dependability/quality of a given observation and measurement type.	Handling errors in data and ML decisions (clause 6.2.2.5)
REQ-ML_ERROR_04	The 3GPP management system shall enable a producer of ML decisions (e.g. an ML-enabled function) to provide to an authorized consumer of ML decisions (e.g. a controller) an ML decision confidence score which is the numerical value that represents the dependability/quality of a given decision generated by the ML-based function.	Handling errors in data and ML decisions (clause 6.2.2.5)

7 Information model definitions for AI/ML management

7.1 Imported and associated information entities

7.1.1 Imported information entities and local labels

Table 7.1.1-1

Label reference	Local label
3GPP TS 28.622 [12], IOC, Top	Top
3GPP TS 28.622 [12], IOC, SubNetwork	SubNetwork
3GPP TS 28.622 [12], IOC, ManagedElement	ManagedElement
3GPP TS 28.622 [12], IOC, ManagedFunction	ManagedFunction

7.2 Class diagram

7.2.1 Relationships

This clause depicts the set of classes (e.g. IOCs) that encapsulates the information relevant to AI/ML model training. For the UML semantics, see 3GPP TS 32.156 [13].

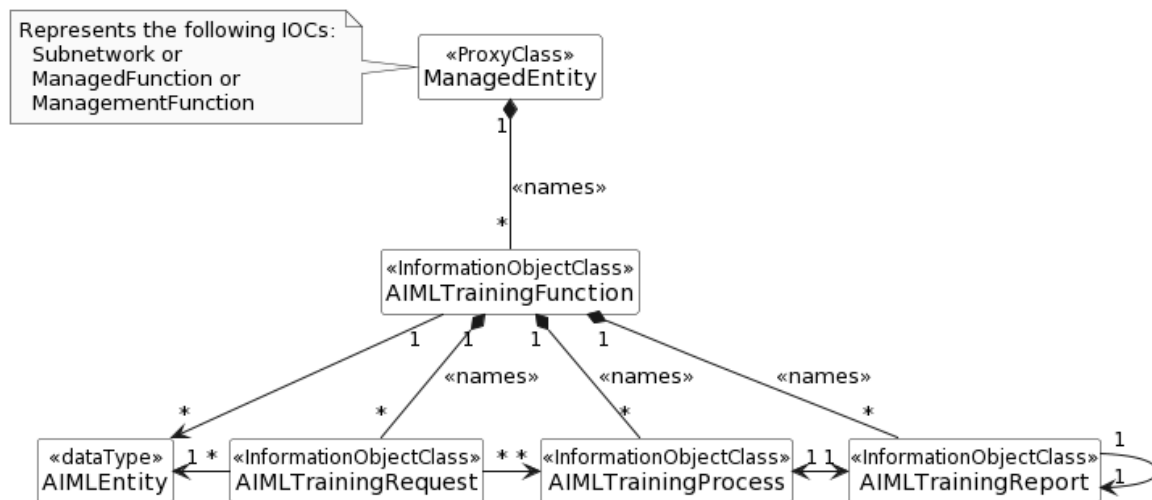


Figure 7.2.1-1: NRM fragment for AI/ML model training

7.2.2 Inheritance

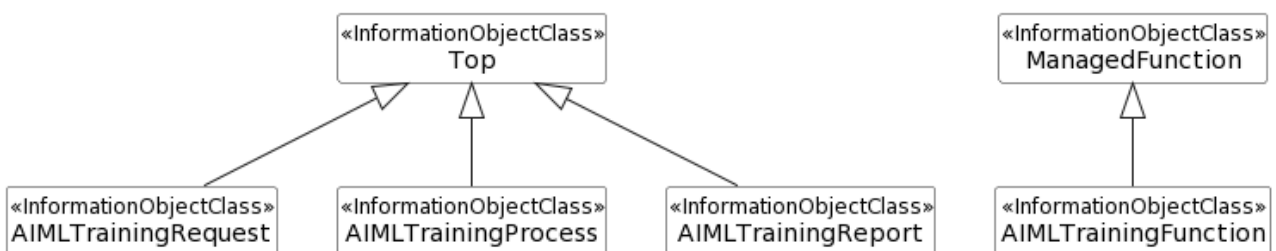


Figure 7.2.2-1: Inheritance Hierarchy for AI/ML model training related NRMs

7.3 Class definitions

7.3.1 AIMLTrainingFunction

7.3.1.1 Definition

The IOC `AIMLTrainingFunction` represents the entity that undertakes AI/ML training and is also the container of the `AIMLTrainingRequest` IOC(s).

The entity represented by `AIMLTrainingFunction` MOI supports training of one or more `AIMLEntity(s)`.

7.3.1.2 Attributes

Table 7.3.1.2-1

Attribute name	Support Qualifier	isReadable	isWritable	isInvariant	isNotifiable
<code>aIMLEntityList</code>	M	T	F	F	F

7.3.1.3 Attribute constraints

None.

7.3.1.4 Notifications

The common notifications defined in clause 7.6 are valid for this IOC, without exceptions or additions.

7.3.2 AIMLTrainingRequest

7.3.2.1 Definition

The IOC `AIMLTrainingRequest` represents the AI/ML model training request that is created by the AI/ML training MnS consumer.

The `AIMLTrainingRequest` MOI is contained under one `AIMLTrainingFunction` MOI. Each `AIMLTrainingRequest` is associated to at least one `AIMLEntity`.

The `AIMLTrainingRequest` may have a source to identify where it is coming from, and which may be used to prioritize the training resources for different sources. The sources may be for example the network functions, operator roles, or other functional differentiations.

Each `AIMLTrainingRequest` may indicate the `expectedRunTimeContext` that describes the specific conditions for which the `AIMLEntity` (either AI/ML Model or AI/ML-enabled function) should be trained for.

In case the request is accepted, the AI/ML training MnS producer decides when to start the AI/ML training. Once the MnS producer decides to start the training based on the request, the AI/ML training MnS producer instantiates one or more `AIMLTrainingProcess` MOI(s) that are responsible to perform the followings:

- collects (more) data for training, if the training data are not available or the data are available but not sufficient for the training;
- prepares and selects the training data, with consideration of the consumer provided candidate training data if any. The AI/ML training MnS producer may examine the consumer's provided candidate training data and select none, some or all of them for training. In addition, the AI/ML training MnS producer may select some other training data that are available;
- trains the `AIMLEntity` using the selected and prepared training data.

The `AIMLTrainingRequest` may have a `requestStatus` field to represent the status of the specific `AIMLTrainingRequest`:

- The attribute values are "NOT_STARTED", "TRAINING_IN_PROGRESS", "SUSPENDED", "FINISHED", and "CANCELLED".
- When value turns to "TrainingInProgress", the AI/ML training MnS producer instantiates one or more `AIMLTrainingProcess` MOI(s) representing the training process(es) being performed per the request and notifies the MnS consumer(s) who subscribed to the notification.

When all of the training process associated to this request are completed, the value turns to "FINISHED".

7.3.2.2 Attributes

Table 7.3.2.2-1

Attribute name	Support Qualifier	isReadable	isWritable	isInvariant	isNotifyable
<code>aIMLEntityId</code>	M	T	T	F	T
<code>candidateTraingDataSource</code>	O	T	T	F	T
<code>traingDataQualityScore</code>	O	T	T	F	T
<code>trainingRequestSource</code>	M	T	T	F	T
<code>requestStatus</code>	M	T	T	F	T
<code>expectedRuntimeContext</code>	O	T	T	F	T
<code>performanceRequirements</code>	M	T	T	F	T
<code>cancelRequest</code>	O	T	T	F	T
<code>suspendRequest</code>	O	T	T	F	T
Attribute related to role					

7.3.2.3 Attribute constraints

None.

7.3.2.4 Notifications

The common notifications defined in clause 7.6 are valid for this IOC, without exceptions or additions.

7.3.3 AIMLTrainingReport

7.3.3.1 Definition

The IOC `AIMLTrainingReport` represents the AI/ML model training report that is provided by the training MnS producer.

The `AIMLTrainingReport` MOI is contained under one `AIMLTrainingFunction` MOI.

7.3.3.2 Attributes

Table 7.3.3.2-1

Attribute name	Support Qualifier	isReadable	isWritable	isInvariant	isNotifiable
aIMLEntityId	M	T	F	F	T
areConsumerTrainingDataUsed	M	T	F	F	T
usedConsumerTrainingData	CM	T	F	F	T
confidenceIndication	O	T	F	F	T
modelPerformanceTraining	CM	T	F	F	T
areNewTrainingDataUsed	CO	T	F	F	T
Attribute related to role					
trainingRequestRef	CM	T	F	F	T
trainingProcessRef	M	T	F	F	T
lastTrainingRef	CM	T	F	F	T

7.3.3.3 Attribute constraints

Table 7.3.3.3-1

Name	Definition
usedConsumerTrainingData Support Qualifier	Condition: The value of areConsumerTrainingDataUsed attribute is ALL or PARTIALLY.
trainingRequestRef Support Qualifier	Condition: The AIMLTrainingReport MOI represents the report for the AI/ML model training that was requested by the MnS consumer (via AIMLTrainingRequest MOI).
lastTrainingRef Support Qualifier	Condition: The AIMLTrainingReport MOI represents the report for the AI/ML model training that was not initial training (i.e. the model has been trained before).
modelPerformanceTraining	The condition is use case " AI/ML training initiated by consumer" is supported.
areNewTrainingDataUsed	The condition is use case " AI/ML training initiated by producer" is supported.

7.3.3.4 Notifications

The common notifications defined in clause 7.6 are valid for this IOC, without exceptions or additions.

7.3.4 AIMLTrainingProcess

7.3.4.1 Definition

The IOC AIMLTrainingProcess represents the AI/ML training process.

One AIMLTrainingProcess MOI may be instantiated for each AIMLTrainingRequest MOI or a set of AIMLTrainingRequest MOIs.

For each AIMLEntity under training, a AIMLTrainingProcess is instantiated, i.e. an AIMLTrainingProcess is associated with exactly one AIMLEntity. The AIMLTrainingProcess may be associated with one or more AIMLTrainingRequest MOI.

The AIMLTrainingProcess does not have to correspond to a specific AIMLTrainingRequest, i.e. a AIMLTrainingRequest does not have to be associated to a specific AIMLTrainingProcess. The AIMLTrainingProcess may be managed separately from the AIMLTrainingRequest MOIs, e.g. the AIMLTrainingRequest MOI may come from consumers which are network functions while the operator may wish to manage the AIMLTrainingProcess that is instantiated following the requests. Thus, the AIMLTrainingProcess may be associated to either one or more AIMLTrainingRequest MOI.

Each `AIMLTrainingProcess` instance needs to be managed differently from the related `AIMLEntity`, although the `AIMLTrainingProcess` may be associated to only one `AIMLEntity`. For example, the `AIMLTrainingProcess` may be triggered to start with a specific version of the `AIMLEntity` and multiple `AIMLTrainingProcess` instances may be triggered for different versions of the `AIMLEntity`. In either case the `AIMLTrainingProcess` instances are still associated with the same `AIMLEntity` but are managed separately from the `AIMLEntity`.

Each `AIMLTrainingProcess` has a priority that may be used to prioritize the execution of different `AIMLTrainingProcess` instances. By default, the priority of the `AIMLTrainingProcess` may be related in a 1:1 manner with the priority of the `AIMLTrainingRequest` for which the `AIMLTrainingProcess` is instantiated.

Each `AIMLTrainingProcess` may have one or more termination conditions used to define the points at which the `AIMLTrainingProcess` may terminate.

The "ProgressStatus" attribute represents the status of the AI/ML model training and includes information the MnS consumer can use to monitor the progress and results. The data type of this attribute is "ProcessMonitor" (see 3GPP TS 28.622 [11]). The following specializations are provided for this data type for the AI/ML training process:

The "ProgressStatus" attribute represents the status of the AI/ML model training and includes information the AI/ML training MnS consumer can use to monitor the progress and results. The data type of this attribute is "ProcessMonitor" (see 3GPP TS 28.622 [12]). The following specializations are provided for this data type for the AI/ML training process:

- The "status" attribute values are "RUNNING", "CANCELLING", "SUSPENDED", "FINISHED", and "CANCELLED". The other values are not used.
- The "timer" attribute is not used.
- When the "status" is equal to "RUNNING" the "progressStateInfo" attribute shall indicate one of the following states: "COLLECTING_DATA", "PREPARING_TRAINING_DATA", "TRAINING".
- No specifications are provided for the "resultStateInfo" attribute. Vendor specific information may be provided though.

When the training is completed with "status" equal to "FINISHED", the MnS producer provides the training report, by creating an `AIMLTrainingReport` MOI, to the MnS consumer.

7.3.4.2 Attributes

Table 7.3.4.2-1

Attribute name	Support Qualifier	isReadable	isWritable	isInvariant	isNotifyable
<code>aIMLTrainingProcessId</code>	M	T	T	F	T
<code>priority</code>	M	T	T	F	T
<code>terminationConditions</code>	M	T	T	F	T
<code>progressStatus</code>	M	T	F	F	T
<code>cancelProcess</code>	O	T	T	F	T
<code>suspendProcess</code>	O	T	T	F	T
Attribute related to role					
<code>trainingRequestRef</code>	CM	T	F	F	T
<code>trainingReportRef</code>	M	T	F	F	T

7.3.4.3 Attribute constraints

Table 7.3.5.3-1

Name	Definition
trainingRequestRef Support Qualifier	Condition: The <code>AIMLTrainingReport</code> MOI represents the report for the AI/ML model training that was requested by the training MnS consumer (via <code>AIMLTrainingRequest</code> MOI).

7.3.4.4 Notifications

The common notifications defined in clause 7.6 are valid for this IOC, without exceptions or additions.

7.4 Data type definitions

7.4.1 ModelPerformance <<dataType>>

7.4.1.1 Definition

This data type specifies the performance of an AI/ML entity when performing inference. The performance score is provided for each inference output.

7.4.1.2 Attributes

Table 7.4.1.2-1

Attribute name	Support Qualifier	isReadable	isWritable	isInvariant	isNotifyable
<code>inferenceOutputName</code>	M	T	F	F	T
<code>performanceScore</code>	M	T	F	F	T
<code>performanceMetric</code>	M	T	F	F	T
<code>decisionConfidenceScore</code>	O	T	F	F	T

7.4.1.3 Attribute constraints

None.

7.4.1.4 Notifications

The notifications specified for the IOC using this <<dataType>> for its attribute(s), shall be applicable.

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. AIML training may be requested for either an AI/ML model or AI/ML-enabled function. The algorithm of AI/ML model or AI/ML-enabled 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

Table 7.4.2.2-1

Attribute name	Support Qualifier	isReadable	isWritable	isInvariant	isNotifiable
aIMLEntityId	M	T	F	F	T
inferenceType	M	T	F	F	T
aIMLEntityVersion	M	T	F	F	T
expectedRunTimeContext	O	T	T	F	T
trainingContext	CM	T	F	F	T
runTimeContext	O	T	F	F	T

7.4.3.3 Attribute constraints

Table 7.4.3.3-1

Name	Definition
trainingContext Support Qualifier	Condition: The trainingContext represents the status and conditions related to training and should be added when training is completed.

7.4.3.4 Notifications

The notifications specified for the IOC using this <<dataType>> for its attribute(s), shall be applicable.

7.4.3 AIMLContext <<dataType>>

7.4.3.1 Definition

The AIMLContext represents the status and conditions related to the AIMLEntity. Specially it may be one of three types of context - the ExpectedRunTimeContext, the TrainingContext and the RunTimeContext.

7.4.3.2 Attributes

Table 7.4.3.2-1

Attribute name	Support Qualifier	isReadable	isWritable	isInvariant	isNotifiable
managedEntityRef	M	T	F	F	F
dataProviderRef	M	T	F	F	F

7.4.3.3 Attribute constraints

None.

7.4.3.4 Notifications

The notifications specified for the IOC using this <<dataType>> for its attribute(s), shall be applicable.

7.5 Attribute definitions

7.5.1 Attribute properties

Table 7.5.1-1

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
candidateTrainingDataSource	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 3GPP TS 28.104 [2]), Analytics ID(s) of NWDAF (see 3GPP 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 [13]) multiplicity: * isOrdered: False isUnique: True defaultValue: None isNullable: True
trainingReportRef	It is the DN of the AIMLTrainingReport MOI that represents the reports of the AI/ML training. allowedValues: DN.	type: DN (see 3GPP TS 32.156 [12]) multiplicity: 1 isOrdered: N/A isUnique: N/A 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 3GPP TS 32.156 [13]) 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

Attribute Name	Documentation and Allowed Values	Properties
		multiplicity: * isOrdered: False isUnique: N/True defaultValue: None isNullable: False
trainingRequestSource	It describes the entity that requested to instantiate the AI/MLTrainingRequest 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. Lower value indicates a higher priority. allowedValues: { 0..65535 }.	type: Integer multiplicity: 1 isOrdered: N/A isUnique: N/A defaultValue: 0 isNullable: False
terminationConditions	It indicates the conditions to be considered by the AI/MLTraining 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 [12]) 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: ModelPerformance 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: ModelPerformance multiplicity: * isOrdered: N/A isUnique: N/A defaultValue: None isNullable: False

Attribute Name	Documentation and Allowed Values	Properties
aIMLTrainingProcess.progress Status.progressStateInfo	<p>It provides the following specialization for the "progressStateInfo" attribute of the "ProcessMonitor" data type for the "aIMLTrainingProcess".</p> <p>When the AI/ML training is in progress, and the "status" is equal to "RUNNING" it provides the more detailed progress information.</p> <p>allowedValues for "status" = "RUNNING":</p> <ul style="list-style-type: none"> - COLLECTING_DATA - PREPARING_TRAINING_DATA - TRAINING <p>The allowed values for "status" = "CANCELLED" are vendor specific.</p>	<p>Type: String multiplicity: 0..1 isOrdered: N/A isUnique: N/A defaultValue: None isNullable: False</p>
inferenceOutputName	<p>It indicates the name of an inference output of an AI/ML entity.</p> <p>allowedValues: the name of the MDA output IEs (see 3GPP 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.</p>	<p>Type: String multiplicity: 0..1 isOrdered: N/A isUnique: N/A defaultValue: None isNullable: False</p>
performanceMetric	<p>It indicates the performance metric used to evaluate the performance of an AI/ML entity, e.g. "accuracy", "precision", "F1 score", etc.</p> <p>allowedValues: N/A.</p>	<p>Type: String multiplicity: 1 isOrdered: N/A isUnique: True defaultValue: None isNullable: False</p>
performanceScore	<p>It indicates the performance score (in unit of percentage) of an AI/ML entity when performing inference on a specific data set (Note).</p> <p>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".</p> <p>allowedValues: { 0..100 }.</p>	<p>Type: Real multiplicity: 0..1 isOrdered: N/A isUnique: N/A defaultValue: None isNullable: False</p>
cancelRequest	<p>It indicates whether the AI/ML training 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".</p> <p>allowedValues: TRUE, FALSE.</p>	<p>Type: Boolean multiplicity: 0..1 isOrdered: N/A isUnique: N/A defaultValue: FALSE isNullable: False</p>
suspendRequest	<p>It indicates whether the AI/ML training 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".</p> <p>allowedValues: TRUE, FALSE.</p>	<p>Type: Boolean multiplicity: 0..1 isOrdered: N/A isUnique: N/A defaultValue: FALSE isNullable: False</p>

Attribute Name	Documentation and Allowed Values	Properties
cancelProcess	It indicates whether the AI/ML training 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: Boolean multiplicity: 0..1 isOrdered: N/A isUnique: N/A defaultValue: FALSE isNullable: False
suspendProcess	It indicates whether the AI/ML training 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: Boolean multiplicity: 0..1 isOrdered: N/A isUnique: N/A defaultValue: FALSE isNullable: False
managedEntityRef	It describes the entities that the AI/ML entity is responsible for managing or optimizing.	Type: DN (see 3GPP TS 32.156 [13]) multiplicity: * isOrdered: False isUnique: True defaultValue: None isNullable: True
dataProviderRef	It describes the entities that have provided or should provide data needed by the AI/ML entity say for training or inference	Type: DN (see 3GPP TS 32.156 [13]) multiplicity: * isOrdered: False isUnique: True defaultValue: None isNullable: True
areNewTrainingDataUsed	It indicates whether the other new training data have been used for the AI/ML model training. allowedValues: TRUE, FALSE.	type: Boolean multiplicity: 1 isOrdered: N/A isUnique: N/A defaultValue: None isNullable: False
traingDataQualityScore	It indicates numerical value that represents the dependability/quality of a given observation and measurement type. The lowest value indicates the lowest level of dependability of the data, i.e. that the data is not usable at all. allowedValues: { 0..100 }.	Type: Real multiplicity: 0..1 isOrdered: N/A isUnique: N/A defaultValue: None isNullable: False
decisionConfidenceScore	the numerical value that represents the dependability/quality of a given decision generated by the ML-based function. The lowest value indicates the lowest level of dependability of the decisions, i.e. that the data is not usable at all. allowedValues: { 0..100 }.	Type: Real multiplicity: 0..1 isOrdered: N/A isUnique: N/A defaultValue: None 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

None.

7.6 Common notifications

7.6.1 Configuration notifications

This clause presents a list of notifications, defined in 3GPP TS 28.532 [11], that an MnS consumer may receive. The notification header attribute `objectClass/objectInstance` shall capture the DN of an instance of a class defined in the present document.

Table 7.6.1-1

Name	Qualifier	Notes
<code>notifyMOICreation</code>	O	--
<code>notifyMOIDeletion</code>	O	--
<code>notifyMOIAttributeValueChanges</code>	O	--
<code>notifyEvent</code>	O	--

8 Service components

8.1 Service components for AI/ML model training MnS

The components for AI/ML model training MnS are listed in table 8.1-1.

Table 8.1-1: Components for AI/ML model training MnS

Management service component type A	Management service component type B	Management service component type C
The operations and notifications for generic provisioning management service (see clause 11.1.1 of 3GPP TS 28.532 [11]).	<code>AIMLTrainingFunction</code> IOC; <code>AIMLTrainingRequest</code> IOC; <code>AIMLTrainingReport</code> IOC; <code>AIMLTrainingProcess</code> IOC.	N/A

9 Solution Set (SS)

The present document defines the following NRM Solution Set definitions for AI/ML management:

- YAML based Solution Set (Annex B).

Annex A (informative): PlantUML source code for NRM class diagrams

A.1 General

This annex contains the PlantUML source code for the NRM diagrams defined in clause 7.2 of the present document.

A.2 PlantUML code for Figure 7.2.1-1: NRM fragment for AI/ML model training

```
@startuml
skinparam ClassStereotypeFontStyle normal
skinparam ClassBackgroundColor White
skinparam shadowing false
skinparam monochrome true
hide members
hide circle
'skinparam maxMessageSize 250

class ManagedEntity <<ProxyClass>>
class AIMLEntity <<dataType>>
class AIMLTrainingFunction <<InformationObjectClass>>
class AIMLTrainingRequest <<InformationObjectClass>>
class AIMLTrainingReport <<InformationObjectClass>>
class AIMLTrainingProcess <<InformationObjectClass>>

ManagedEntity "1" *-- "*" AIMLTrainingFunction: <<names>>
AIMLTrainingFunction "1" -d-> "*" AIMLEntity
AIMLTrainingFunction "1" *-- "*" AIMLTrainingProcess: <<names>>
AIMLTrainingFunction "1" *-- "*" AIMLTrainingRequest: <<names>>
AIMLTrainingFunction "1" *-- "*" AIMLTrainingReport: <<names>>

AIMLTrainingProcess "1" <-r-> "1" AIMLTrainingReport
AIMLTrainingReport "1" --> "1" AIMLTrainingReport
AIMLTrainingRequest "*" -l-> "1" AIMLEntity
AIMLTrainingRequest "*" -r-> "*" AIMLTrainingProcess

note left of ManagedEntity
  Represents the following IOCs:
  Subnetwork or
  ManagedFunction or
  ManagementFunction
end note

@enduml
```

A.3 PlantUML code for Figure 7.2.2-1: Inheritance Hierarchy for AI/ML model training related NRMs

```
@startuml
skinparam ClassStereotypeFontStyle normal
skinparam ClassBackgroundColor White
skinparam shadowing false
skinparam monochrome true
hide members
hide circle
'skinparam maxMessageSize 250

class Top <<InformationObjectClass>>
class ManagedFunction <<InformationObjectClass>>
class AIMLTrainingFunction <<InformationObjectClass>>
class AIMLTrainingRequest <<InformationObjectClass>>
class AIMLTrainingProcess <<InformationObjectClass>>
class AIMLTrainingReport <<InformationObjectClass>>

ManagedFunction <|-- AIMLTrainingFunction
Top <|-- AIMLTrainingRequest
Top <|-- AIMLTrainingProcess
Top <|-- AIMLTrainingReport

@enduml
```

Annex B (normative): OpenAPI definition of the AI/ML NRM

B.1 General

This annex contains the OpenAPI definition of the AI/ML NRM in YAML format.

The information models of the AI/ML NRM are defined in clause 7.

Mapping rules to produce the OpenAPI definition based on the information model are defined in 3GPP TS 32.160 [14].

B.2 Solution Set (SS) definitions

B.2.1 OpenAPI document "AiMLNrm.yaml"

```

openapi: 3.0.1
info:
  title: AI/ML NRM
  version: 17.0.0
  description: >-
    OAS 3.0.1 specification of the AI/ML NRM
    © 2020, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).
    All rights reserved.
externalDocs:
  description: TS 28.105; AI/ML Management
  url: http://www.3gpp.org/ftp/Specs/archive/28_series/28.105/
paths: {}
components:
  schemas:

#----- Definition of types-----

  AIMLEntityList:
    type: array
    items:
      $ref: '#/components/schemas/AIMLEntity'

  AIMLEntity:
    type: object
    properties:
      aIMLEntityId:
        type: string
      inferenceType:
        type: string
      aIMLEntityVersion:
        type: string
      expectedRunTimeContext:
        type: string
      trainingContext:
        type: string
      runTimeContext:
        type: string

  RequestStatus:
    type: string
    enum:
      - NOT_STARTED
      - TRAINING_IN_PROGRESS
      - SUSPENDED
      - FINISHED
      - CANCELLED

  PerformanceRequirements:
    type: array
    items:
      $ref: '#/components/schemas/ModelPerformance'

```

```

ModelPerformance:
  type: object
  properties:
    inferenceOutputName:
      type: string
    performanceMetric:
      type: string
    performanceScore:
      type: number
      format: float
    decisionConfidenceScore:
      type: number
      format: float

TrainingProcessMonitor:
  description: >-
    This data type is the "ProcessMonitor" data type defined in "genericNrm.yaml" with
  specialisations for usage in the "AIMLTrainingProcess".
  type: object
  properties:
    aIMLTrainingProcessId:
      type: string
    status:
      type: string
      enum:
        - RUNNING
        - CANCELLING
        - CANCELLED
        - SUSPENDED
        - FINISHED
    progressPercentage:
      type: integer
      minimum: 0
      maximum: 100
    progressStateInfo:
      type: string
      enum:
        - COLLECTING_DATA
        - PREPARING_TRAINING_DATA
        - TRAINING
    resultStateInfo:
      type: string

#----- Definition of abstract IOCs -----

#----- Definition of concrete IOCs -----

SubNetwork-Single:
  allOf:
    - $ref: 'genericNrm.yaml#/components/schemas/Top'
    - type: object
      properties:
        attributes:
          $ref: 'genericNrm.yaml#/components/schemas/SubNetwork-Attr'
    - $ref: 'genericNrm.yaml#/components/schemas/SubNetwork-ncO'
    - type: object
      properties:
        SubNetwork:
          $ref: '#/components/schemas/SubNetwork-Multiple'
        ManagedElement:
          $ref: '#/components/schemas/ManagedElement-Multiple'
        AIMLTrainingFunction:
          $ref: '#/components/schemas/AIMLTrainingFunction-Multiple'

ManagedElement-Single:
  allOf:
    - $ref: 'genericNrm.yaml#/components/schemas/Top'
    - type: object
      properties:
        attributes:
          $ref: 'genericNrm.yaml#/components/schemas/ManagedElement-Attr'
    - $ref: 'genericNrm.yaml#/components/schemas/ManagedElement-ncO'
    - type: object
      properties:
        AIMLTrainingFunction:
          $ref: '#/components/schemas/AIMLTrainingFunction-Multiple'

```

```

AIMLTrainingFunction-Single:
  allOf:
    - $ref: 'genericNrm.yaml#/components/schemas/Top'
    - type: object
      properties:
        attributes:
          allOf:
            - $ref: 'genericNrm.yaml#/components/schemas/ManagedFunction-Attr'
            - type: object
              properties:
                aIMLEntityList:
                  $ref: '#/components/schemas/AIMLEntityList'
    - $ref: 'genericNrm.yaml#/components/schemas/ManagedFunction-ncO'
    - type: object
      properties:
        AIMLTrainingRequest:
          $ref: '#/components/schemas/AIMLTrainingRequest-Multiple'
        AIMLTrainingProcess:
          $ref: '#/components/schemas/AIMLTrainingProcess-Multiple'
        AIMLTrainingReport:
          $ref: '#/components/schemas/AIMLTrainingReport-Multiple'

AIMLTrainingRequest-Single:
  allOf:
    - $ref: 'genericNrm.yaml#/components/schemas/Top'
    - type: object
      properties:
        attributes:
          allOf:
            - type: object
              properties:
                aIMLEntityId:
                  type: string
                candidateTraingDataSource:
                  type: array
                  items:
                    type: string
                traingDataQualityScore:
                  type: number
                  format: float
                trainingRequestSource:
                  type: string
                requestStatus:
                  $ref: '#/components/schemas/RequestStatus'
                expectedRuntimeContext:
                  $ref: 'comDefs.yaml#/components/schemas/DateTime'
                performanceRequirements:
                  $ref: '#/components/schemas/PerformanceRequirements'
                cancelRequest:
                  type: boolean
                suspendRequest:
                  type: boolean

AIMLTrainingProcess-Single:
  allOf:
    - $ref: 'genericNrm.yaml#/components/schemas/Top'
    - type: object
      properties:
        attributes:
          allOf:
            - type: object
              properties:
                aIMLTrainingProcessId:
                  type: string
                priority:
                  type: integer
                terminationConditions:
                  type: string
                progressStatus:
                  $ref: '#/components/schemas/TrainingProcessMonitor'
                cancelProcess:
                  type: boolean
                suspendProcess:
                  type: boolean
                trainingRequestRef:
                  $ref: 'comDefs.yaml#/components/schemas/DnList'
                trainingReportRef:

```



```
$ref: 'comDefs.yaml#/components/schemas/Dn'
```

```
AIMLTrainingReport-Single:
  allOf:
  - $ref: 'genericNrm.yaml#/components/schemas/Top'
  - type: object
    properties:
      attributes:
        allOf:
        - type: object
          properties:
            aIMLEntityId:
              type: string
            areConsumerTrainingDataUsed:
              type: boolean
            usedConsumerTrainingData:
              type: array
              items:
                type: string
            confidenceIndication:
              type: integer
            modelPerformanceTraining:
              type: array
              items:
                $ref: '#/components/schemas/ModelPerformance'
            areNewTrainingDataUsed:
              type: boolean
```

```
#----- Definition of JSON arrays for name-contained IOCs -----
```

```
SubNetwork-Multiple:
  type: array
  items:
    $ref: '#/components/schemas/SubNetwork-Single'
ManagedElement-Multiple:
  type: array
  items:
    $ref: '#/components/schemas/ManagedElement-Single'
AIMLTrainingFunction-Multiple:
  type: array
  items:
    $ref: '#/components/schemas/AIMLTrainingFunction-Single'
AIMLTrainingRequest-Multiple:
  type: array
  items:
    $ref: '#/components/schemas/AIMLTrainingRequest-Single'
AIMLTrainingProcess-Multiple:
  type: array
  items:
    $ref: '#/components/schemas/AIMLTrainingProcess-Single'
AIMLTrainingReport-Multiple:
  type: array
  items:
    $ref: '#/components/schemas/AIMLTrainingReport-Single'
```

```
#----- Definitions in 3GPP TS 28.104 for 3GPP TS 28.532 -----
```

```
resources-AiMlNrm:
  oneOf:
  - $ref: '#/components/schemas/SubNetwork-Single'
  - $ref: '#/components/schemas/ManagedElement-Single'
  - $ref: '#/components/schemas/AIMLTrainingFunction-Single'
  - $ref: '#/components/schemas/AIMLTrainingRequest-Single'
  - $ref: '#/components/schemas/AIMLTrainingProcess-Single'
  - $ref: '#/components/schemas/AIMLTrainingReport-Single'
```

Annex C (informative): Change history

Change history							
Date	Meeting	TDoc	CR	Rev	Cat	Subject/Comment	New version
2022-01	SA5#141e	n/a	-	-	-	Initial skeleton	0.0.0
2022-01	SA5#141e	S5-221616	-	-	-	Add scope	0.1.0
2022-01	SA5#141e	S5-221617	-	-	-	Add overview	0.1.0
2022-01	SA5#141e	S5-221618	-	-	-	Add service framework for AI-ML model training	0.1.0
2022-01	SA5#141e	S5-221335	-	-	-	Move in ML model training part from TS 28.104	0.1.0
2022-01	SA5#141e	S5-221614	-	-	-	Add the requirements for ML model training for MDA	0.1.0
2022-02	SA5#141e	S5-221620	-	-	-	Add NRMs for AI-ML model training	0.1.0
2022-03	SA#95e	SP-220128				Presented for information	1.0.0
2022-04	SA5#142e	S5-222688	-	-	-	Add requirements for AI/ML selection	1.1.0
2022-04	SA5#142e	S5-222689	-	-	-	Add requirements for control of AI/ML Training	1.1.0
2022-04	SA5#142e	S5-222690	-	-	-	Add NRM fragments for AI/ML Training	1.1.0
2022-04	SA5#142e	S5-222691	-	-	-	Support for Training of ML-based functions	1.1.0
2022-04	SA5#142e	S5-222696	-	-	-	Add AI-ML background	1.1.0
2022-05	SA5#143e	S5-223646	-	-	-	Add AIML training attribute in case of AIML training initiated by producer	1.2.0
2022-05	SA5#143e	S5-223383	-	-	-	Correct the attribute definitions of NRMs for AI-ML model training	1.2.0
2022-05	SA5#143e	S5-223474	-	-	-	Update AI-ML NRM	1.2.0
2022-05	SA5#143e	S5-223487	-	-	-	Updates to AI/ML NRM	1.2.0
2022-05	SA5#143e	S5-223650	-	-	-	Rapporteur clean-up	1.2.0
2022-05	SA5#143e	S5-223652	-	-	-	Add requirements for handling erroneous data & decisions	1.2.0
2022-05	SA5#143e	S5-223745	-	-	-	Update the definition of performanceScore	1.2.0
2022-05	SA5#143e	S5-223647	-	-	-	Add stage 3 solution sets for AI-ML NRM	1.2.0
2022-06	SA#96	SP-220492				Presented for approval	2.0.0
2022-06	SA#96					Upgrade to change control version	17.0.0

History

Document history		
V17.0.0	July 2022	Publication