ETSI TS 128 105 V17.6.0 (2024-01)

TECHNICAL SPECIFICATION

5G;
Management and orchestration;
Artificial Intelligence/ Machine Learning (AI/ML) management
(3GPP TS 28.105 version 17.6.0 Release 17)

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:
https://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/CommiteeSupportStaff.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.

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 https://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

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".

[15]   3GPP TS 28.533: "Management and orchestration; Architecture framework".

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].

ML entity: an entity that is either an ML model or contains an ML model and ML model related metadata, it can be managed as a single composite entity.

　　NOTE 1: Metadata may include e.g. the applicable runtime context for the ML model.

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.

　　NOTE 2: The ML models are proprietary and not in scope for standardization.

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

ML training: capabilities and associated end-to-end processes to enable an ML training function to perform ML model training (as defined above).

　　NOTE 3: ML training capabilities may include interaction with other parties to collect and format the data required for ML model training.

ML training function: a function with ML training capabilities; it is also referred to as MLT function.

AI/ML inference function: a function that employs an ML model to conduct inference.

3.2　Symbols

Void.

3.3　Abbreviations

For the purposes of the present document, the abbreviations given in TR 21.905 [1] and TS 28.533 [15]. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in TR 21.905 [1] and TS 28.533 [15].

　　AI　　　　　Artificial Intelligence
　　ML　　　　　Machine Learning

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, semi-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-inference function in the 5GS uses the 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 ML model and AI/ML inference function need to be managed.

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

- ML training.

4A   AI/ML management functionality and service framework

4A.1   Functionality and service framework for ML training

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

As illustrated in Figure 4A.1-1 the ML training capability is provided via ML training MnS in the context of SBMA to the authorized consumer(s) by ML training MnS producer.
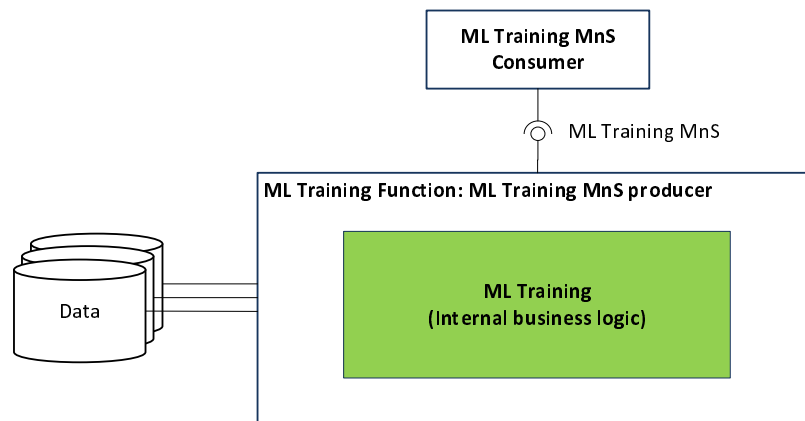
Figure 4A.1-1: Functional overview and service framework for ML training

The internal business logic of ML training leverages the current and historical relevant data, including those listed below to monitor the networks and/or services where relevant to the 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.

5 Void

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 ML training

6.2.1 Description

In operational environment before the ML entity is deployed to conduct inference, the ML model associated with the ML entity needs to be trained (e.g. by ML training function which may be a separate or an external entity to the AI/ML inference function).

NOTE: In the present document, ML entity training refers to ML model training associated with an ML entity.

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

6.2.2   Use cases

6.2.2.1   ML training requested by consumer

The ML training capabilities are provided by an MLT MnS producer to one or more consumer(s).
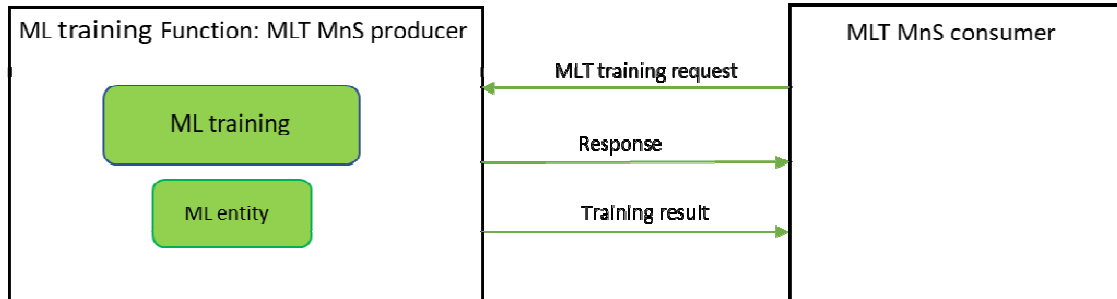
Figure 6.2.2.1-1: ML training requested by MLT MnS consumer

The ML training may be triggered by the request(s) from one or more MLT MnS consumer(s). The consumer may be for example a network function, a management function, an operator, or another functional differentiation To trigger an ML training, the MLT MnS consumer requests the MLT MnS producer to train the ML model. In the ML training request, the consumer should specify the inference type which indicates the function or purpose of the ML entity, e.g. CoverageProblemAnalysis. The MLT 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 MLT MnS producer provides a response to the consumer indicating whether the request was accepted.

If the request is accepted, the MLT MnS producer decides when to start the 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 ML Entity, the MLT MnS producer may examine the consumer's provided training data and decide to select none, some or all of them. In addition, the MLT MnS producer may select some other training data that are available;

- trains the ML entity using the selected training data;

- provides the training results to the MLT MnS consumer(s).

6.2.2.2   ML training initiated by producer

The ML training may be initiated by the MLT MnS producer, for instance as a result of performance evaluation of the 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 MLT MnS producer decides to start the ML training, the producer performs the followings:

- selects the training data;

- trains the ML entity using the selected training data;

- provides the training results to the MLT MnS consumer(s) who have subscribed to receive the ML training results.

6.2.2.3   ML model and ML entity selection

For a given machine learning-based use case, different entities that apply the respective ML model or AI/ML inference function may have different inference requirements and capabilities. For example, one consumer with specific responsibility and wish to have an AI/ML inference function supported by an ML model or entity 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 wishes to have an ML model and AI/ML inference function fitting that type of environment. The different consumers need to know the available versions of ML entities, with the variants of trained ML models or entities and to select the appropriate one for their respective conditions.

Besides, there is no guarantee that the available ML models/entities have been trained according to the characteristics that the consumers expect. As such the consumers need to know the conditions for which the ML models or ML entities have been trained to then enable them to select the models that are best fit to their conditions and needs.

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 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 levels of complexity and performance which would then allow different relevant models to be delivered to different consumer depending on operating conditions and performance requirements. The consumer 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 requirements.

6.2.2.4   Managing ML training processes

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

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

In either case, the network management system not only needs to have the required training capabilities but needs to also have the means to manage the training of the ML models/entities. 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 ML models/entities (e.g. , ML entity1 and ML entity2 in figure 6.2.2.5-1) are trained on good quality data, i.e. , data that were collected correctly and reflected the real network status to represent the expected context in which the ML 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 ML entity can depend on a few precise inputs, and does not need to exploit the redundancy present in the training data. However, during inference, the ML entity is very likely to come across these inconsistencies. When this happens, the ML entity shows high error in the inference outputs, even if redundant and uncorrupted data are available from other sources.
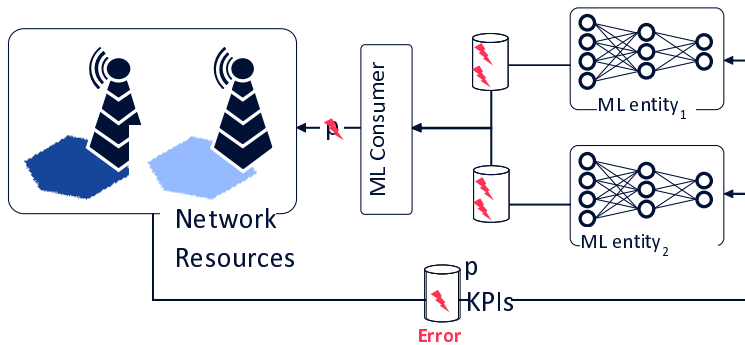
Figure 6.2.2.5-1: The propagation of erroneous information

As such the system needs to account for errors and inconsistencies in the input data and the consumers 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 ML entities to deal with the errors in the training data, i.e. , to identify the errors in the data during training;

2) enable the MLT MnS consumers to be aware of the possibility of erroneous input data that are used by the ML entity.

6.2.3   Requirements for ML training

Table 6.2.3-1

Requirement label	Description	Related use case(s)
REQ-ML_TRAIN-FUN-01	The MLT MnS producer shall have a capability allowing the consumer to request ML training.	ML training requested by consumer (clause 6.2.2.1)
REQ- ML_TRAIN-FUN-02	The MLT MnS producer shall have a capability allowing the consumer to specify the data sources containing the candidate training data for ML training.	ML training requested by consumer (clause 6.2.2.1)
REQ- ML_TRAIN-FUN-03	The MLT MnS producer shall have a capability allowing the consumer to specify the inference type of the ML model entity to be trained.	ML training requested by consumer (clause 6.2.2.1)
REQ- ML_TRAIN-FUN-04	The MLT MnS producer shall have a capability to provide the training result to the consumer.	ML training requested by consumer (clause 6.2.2.1), /ML training initiated by producer (clause 6.2.2.2)
REQ-ML_SELECT-01	3GPP management system shall have the capability to enable an authorized consumer to discover the characteristics of available models including the contexts under which each of the models was trained.	ML model and ML entity selection (clause 6.2.2.3)
REQ-ML_SELECT-02	3GPP management system shall have the capability to enable an authorized consumer to select an ML model.	ML models and ML entity selection (clause 6.2.2.3)
REQ-ML_SELECT-03	The MLT MnS producer shall have the capability to enable an authorized consumer to request for a model to be trained to satisfy the consumer's expectations.	ML training requested by consumer (clause 6.2.2.1), ML model and ML entity selection (clause 6.2.2.3)
REQ-ML_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.	ML model and ML entity selection (clause 6.2.2.3)
REQ-ML_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.	ML model and ML entity selection (clause 6.2.2.3)

Requirement label	Description	Related use case(s)
REQ-ML_SELECT-06	The 3GPP management system shall have a capability to provide a selected ML model/entity to the consumer.	ML model and ML entity selection (clause 6.2.2.3)
REQ-ML_TRAIN- MGT-01	The MLT MnS producer shall have a capability allowing an authorized consumer to manage and configure one or more requests for the training of specific ML models or ML entities, e.g. to modify the characteristics of the request or to delete a request.	ML training requested by consumer (clause 6.2.2.1),Managing ML Training Processes (clause 6.2.2.4)
REQ-ML_TRAIN- MGT-02	The MLT MnS producer shall have a capability allowing 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.	ML training requested by consumer (clause 6.2.2.1), Managing ML training processes (clause 6.2.2.4)
REQ-ML_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 ML model/entity training) to request for a report on the outcomes of a specific training instance.	Managing ML training processes (clause 6.2.2.4)
REQ-ML_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 ML training processes (clause 6.2.2.4)
REQ-ML_TRAIN- MGT-05	3GPP management system shall have the capability to enable the MLT 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 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 MLT 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 AI/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 an AI/ML-inference 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 MLT 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 AI/ML inference function) to provide to an authorized consumer of AI/ML decisions (e.g. a controller) an AI/ML decision confidence score which is the numerical value that represents the dependability/quality of a given decision generated by the inference 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 ML model training. For the UML semantics, see TS 32.156 [13].
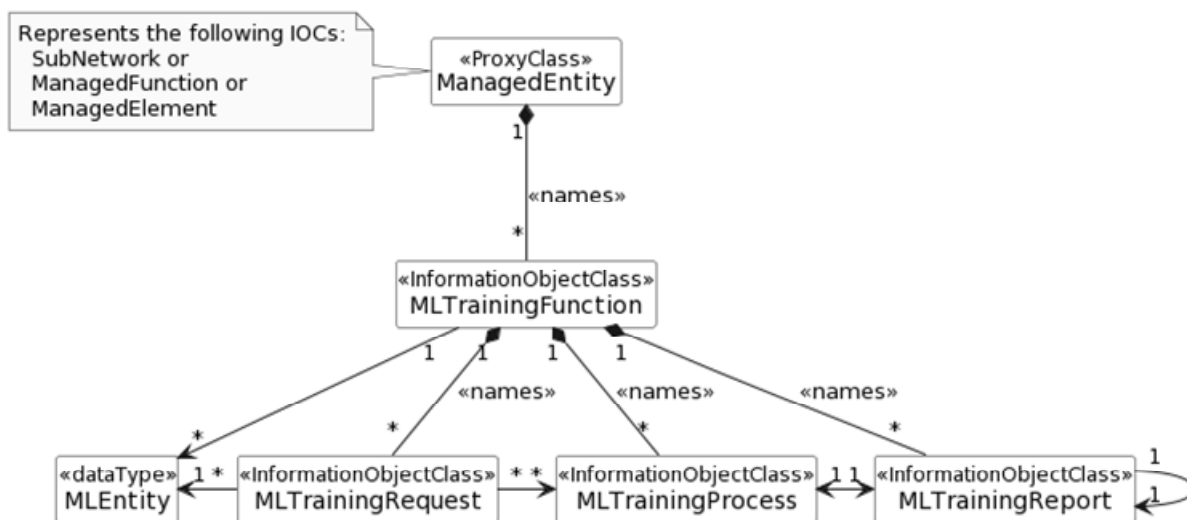
Figure 7.2.1-1: NRM fragment for ML training

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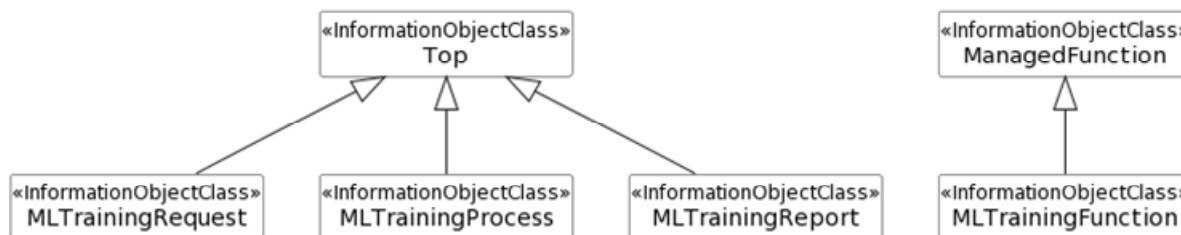
Figure 7.2.2-1: Inheritance Hierarchy for ML training related NRMs

7.3 Class definitions

7.3.1 `MLTrainingFunction`

7.3.1.1 Definition

The IOC `MLTrainingFunction` represents the entity that undertakes ML training and is also the container of the `MLTrainingRequest` IOC(s).

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

7.3.1.2 Attributes

Table 7.3.1.2-1

Attribute name	Support Qualifier	isReadable	isWritable	isInvariant	isNotifyable
mLEntityList	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 `MLTrainingRequest`

7.3.2.1 Definition

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

The `MLTrainingRequest` MOI is contained under one `MLTrainingFunction` MOI. Each `MLTrainingRequest` is associated to at least one `MLEntity`.

The `MLTrainingRequest` 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 `MLTrainingRequest` may indicate the expectedRunTimeContext that describes the specific conditions for which the `MLEntity` should be trained.

In case the request is accepted, the ML training MnS producer decides when to start the ML training. Once the MnS producer decides to start the training based on the request, the ML training MnS producer instantiates one or more MLTrainingProcess 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 required training data, with consideration of the consumer's request provided candidate training data if any. The 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 ML training MnS producer may select some other training data that are available in order to meet the consumer's requirements for the MLentity training;

- trains the `MLEntity` using the selected and prepared training data.

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

- The attribute values are "NOT_STARTED", "TRAINING_IN_PROGRESS", "SUSPENDED", "FINISHED", and "CANCELLED".

- When value turns to "TRAINING_IN_PROGRESS", the ML training MnS producer instantiates one or more `MLTrainingProcess` MOI(s) representing the training process(es) being performed per the request and notifies the MLT 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
mLEntityId	M	T	T	F	T
candidateTrainingDataSource	O	T	T	F	T
trainingDataQualityScore	O	T	T	F	T
trainingRequestSource	M	T	T	F	T
requestStatus	M	T	F	F	T
expectedRuntimeContext	O	T	T	F	T
performanceRequirements	M	T	T	F	T
cancelRequest	O	T	T	F	T
suspendRequest	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   MLTrainingReport

7.3.3.1   Definition

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

The `MLTrainingReport` MOI is contained under one `MLTrainingFunction` MOI.

7.3.3.2 Attributes

Table 7.3.3.2-1

Attribute name	Support Qualifier	isReadable	isWritable	isInvariant	isNotifyable
mLEntityId	M	T	F	F	T
areConsumerTrainingDataUsed	M	T	F	F	T
usedConsumerTrainingData	CM	T	F	F	T
modelConfidenceIndication	O	T	F	F	T
modelPerformanceTraining	M	T	F	F	T
areNewTrainingDataUsed	M	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 MLTrainingReport MOI represents the report for the ML model training that was requested by the MnS consumer (via MLTrainingRequest MOI).
lastTrainingRef Support Qualifier	Condition: The MLTrainingReport MOI represents the report for the ML model training that was not initial training (i.e. the model has been trained before).

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 MLTrainingProcess

7.3.4.1 Definition

The IOC MLTrainingProcess represents the ML training process.

One MLTrainingProcess MOI may be instantiated for each MLTrainingRequest MOI or a set of MLTrainingRequest MOIs.

For each MLEntity under training, a MLTrainingProcess is instantiated, i.e. an MLTrainingProcess is associated with exactly one MLEntity. The MLTrainingProcess may be associated with one or more MLTrainingRequest MOI.

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

Each MLTrainingProcess instance needs to be managed differently from the related MLEntity, although the MLTrainingProcess may be associated to only one MLEntity. For example, the MLTrainingProcess may be triggered to start with a specific version of the MLEntity and multiple MLTrainingProcess instances may be

triggered for different versions of the MLEntity. In either case the MLTrainingProcess instances are still associated with the same MLEntity but are managed separately from the MLEntity.

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

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

The "progressStatus" attribute represents the status of the ML model training and includes information the 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 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 MLT MnS producer provides the training report, by creating an MLTrainingReport MOI, to the MLT MnS consumer.

7.3.4.2   Attributes

Table 7.3.4.2-1

Attribute name	Support Qualifier	isReadable	isWritable	isInvariant	isNotifyable
mLTrainingProcessId	M	T	T	F	T
priority	M	T	T	F	T
terminationConditions	M	T	T	F	T
progressStatus	M	T	F	F	T
cancelProcess	O	T	T	F	T
suspendProcess	O	T	T	F	T
Attribute related to role					
trainingRequestRef	CM	T	F	F	T
trainingReportRef	M	T	F	F	T

7.3.4.3   Attribute constraints

Table 7.3.5.3-1

Name	Definition
trainingRequestRef Support Qualifier	Condition: The MLTrainingReport MOI represents the report for the ML model training that was requested by the training MnS consumer (via MLTrainingRequest 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 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
inferenceOutputName	M	T	T/F (NOTE)	F	T
performanceScore	M	T	T/F (NOTE)	F	T
performanceMetric	M	T	T/F (NOTE)	F	T
decisionConfidenceScore	O	T	F	F	T
NOTE: The isWritable qualifier is "T" if the attribute is used in MLTrainingRequest; The isWritable qualifier is "F" otherwise.					

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   `MLEntity <<dataType>>`

7.4.2.1   Definition

This data type represents the properties of an ML entity. ML training may be requested for either an ML model or ML entity. ML model is not to be standardized.

For each `MLEntity` under training, one or more `MLTrainingProcess` are instantiated.

The `MLEntity` may contain 3 types of contexts - TrainingContext which is the context under which the `MLEntity` has been trained, the ExpectedRunTimeContext which is the context where an `MLEntity` is expected to be applied or/and the RunTimeContext which is the context where the MLmodel or entity is being applied.

7.4.2.2   Attributes

Table 7.4.2.2-1

Attribute name	Support Qualifier	isReadable	isWritable	isInvariant	isNotifyable
mLEntityId	M	T	F	F	T
inferenceType	M	T	F	F	T
mLEntityVersion	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 MLContext <<dataType>>

7.4.3.1 Definition

The `MLContext` represents the status and conditions related to the `MLEntity`. 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	isNotifyable
`inferenceEntityRef`	CM	T	F	F	F
`dataProviderRef`	M	T	F	F	F

7.4.3.3 Attribute constraints

Table 7.4.3.3-1

Name	Definition
`inferenceEntityRef` Support Qualifier	Condition: The `MLContext` is used for `ExpectedRunTimeContext` or `RunTimeContext`.

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
`mLEntityId`	It identifies the 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: False

Attribute Name	Documentation and Allowed Values	Properties
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: False
inferenceType	It indicates the type of inference that the 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: False
areConsumerTrainingDataUsed	It indicates whether the consumer provided training data have been used for the ML model training. allowedValues: ALL, PARTIALLY, NONE.	type: Enum multiplicity: 1 isOrdered: N/A isUnique: N/A defaultValue: None isNullable: False
usedConsumerTrainingData	It provides the address(es) where lists of the consumer-provided training data are located, which have been used for the ML model training. allowedValues: N/A.	type: String multiplicity: * isOrdered: False isUnique: True defaultValue: None isNullable: False
trainingRequestRef	It is the DN(s) of the related MLTrainingRequest MOI(s). allowedValues: DN.	type: DN (see TS 32.156 [13]) multiplicity: * isOrdered: False isUnique: True defaultValue: None isNullable: False
trainingProcessRef	It is the DN(s) of the related MLTrainingProcess MOI(s) that produced the MLTrainingReport. allowedValues: DN.	type: DN (see TS 32.156 [13]) multiplicity: 0..1 isOrdered: N/A isUnique: N/A defaultValue: None isNullable: False
trainingReportRef	It is the DN of the MLTrainingReport MOI that represents the reports of the ML training. allowedValues: DN.	Type: DN (see TS 32.156 [13]) multiplicity: 0..1 isOrdered: N/A isUnique: N/A defaultValue: None isNullable: False
lastTrainingRef	It is the DN of the MLTrainingReport MOI that represents the reports for the last training of the ML model. allowedValues: DN.	type: DN (see 3GPP TS 32.156 [13]) multiplicity: 0..1 isOrdered: N/A isUnique: N/A defaultValue: None isNullable: False
modelConfidenceIndication	It indicates the average confidence value (in unit of percentage) that the ML model would perform for inference on the data with the same distribution as training data. Essentially, this is a measure of degree of the convergence of the trained ML model. allowedValues: { 0..100 }.	type: integer multiplicity: 1 isOrdered: N/A isUnique: N/A defaultValue: None isNullable: False
mLEntityList	It describes the list of MLEntity.	type: MLEntity multiplicity: * isOrdered: False isUnique: True defaultValue: None isNullable: False

Attribute Name	Documentation and Allowed Values	Properties
`trainingRequestSource`	It describes the entity that requested to instantiate the `MLTrainingRequest` MOI.	type: String multiplicity: 1 isOrdered: N/A isUnique: N/A defaultValue: None isNullable: False
`requestStatus`	It describes the status of a particular ML training request. 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
`mLTrainingProcessId`	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: False
`priority`	It indicates the priority of the training process. The priority may be used by the 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 MLtraining MnS producer to terminate a specific training process. allowedValues: MODEL UPDATED_IN_INFERENCE_FUNCTION, INFERENCE FUNCTION_TERMINATED, INFERENCE FUNCTION_UPGRADED, INFERENCE_CONTEXT_CHANGED	type: String multiplicity: 1 isOrdered: N/A isUnique: N/A defaultValue: None isNullable: False
`progressStatus`	It indicates the status of the 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
`mLEntityVersion`	It indicates the version number of the 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 ML entity when performing on the training data. allowedValues: N/A.	type: ModelPerformance multiplicity: * isOrdered: False isUnique: True defaultValue: None isNullable: False
`modelperformanceTraining`	It indicates the performance score of the ML entity when performing on the training data. allowedValues: N/A.	type: ModelPerformance multiplicity: * isOrdered: False isUnique: True defaultValue: None isNullable: False

Attribute Name	Documentation and Allowed Values	Properties
`mLTrainingProcess.progressStatus.progressStateInfo`	It provides the following specialization for the "progressStateInfo" attribute of the "ProcessMonitor" data type for the `"MLTrainingProcess"`. When the ML training is in progress, and the " mLTrainingProcess.progressStatus.status " is equal to "RUNNING", it provides the more detailed progress information. allowedValues for " mLTrainingProcess.progressStatus.status " = "RUNNING": - COLLECTING_DATA - PREPARING_TRAINING_DATA - TRAINING The allowed values for " mLTrainingProcess.progressStatus.status " = "CANCELLED" are vendor specific.	Type: String multiplicity: 0..1 isOrdered: N/A isUnique: N/A defaultValue: None isNullable: False
`inferenceOutputName`	It indicates the name of an inference output of an ML entity. 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.	Type: String multiplicity: 1 isOrdered: N/A isUnique: N/A defaultValue: None isNullable: False
`performanceMetric`	It indicates the performance metric used to evaluate the performance of an ML entity, e.g. "accuracy", "precision", "F1 score", etc. allowedValues: N/A.	Type: String multiplicity: 1 isOrdered: N/A isUnique: N/A defaultValue: None isNullable: False
`performanceScore`	It indicates the performance score (in unit of percentage) of an ML entity when performing inference on a specific data set (Note). The performance metrics may be different for different kinds of 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". allowedValues: { 0..100 }.	Type: Real multiplicity: 1 isOrdered: N/A isUnique: N/A defaultValue: None isNullable: False
`cancelRequest`	It indicates whether the ML training MnS consumer cancels the ML training request. Setting this attribute to "TRUE" cancels the 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". allowedValues: TRUE, FALSE.	Type: Boolean multiplicity: 0..1 isOrdered: N/A isUnique: N/A defaultValue: FALSE isNullable: False
`suspendRequest`	It indicates whether the ML training MnS consumer suspends the /ML training request. Setting this attribute to "TRUE" suspends the ML training request. Suspension is possible when the `requestStatus` is not the "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

Attribute Name	Documentation and Allowed Values	Properties
cancelProcess	It indicates whether the ML training MnS consumer cancels the ML training process. Setting this attribute to "TRUE" cancels the ML training request. Cancellation is possible when the " mLTrainingProcess.progressStatus.status" is not the "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 ML training MnS consumer suspends the ML training process. Setting this attribute to "TRUE" suspends the ML training request. Suspension is possible when the " mLTrainingProcess.progressStatus.status" is not the "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
inferenceEntityRef	It describes the target entities that will use the ML entity for inference.	Type: DN (see 3GPP TS 32.156 [13]) multiplicity: * isOrdered: False isUnique: True defaultValue: None isNullable: False
dataProviderRef	It describes the entities that have provided or should provide data needed by the ML entity e.g. for training or inference	Type: DN (see 3GPP TS 32.156 [13]) multiplicity: * isOrdered: False isUnique: True defaultValue: None isNullable: False
areNewTrainingDataUsed	It indicates whether the other new training data have been used for the ML model training. allowedValues: TRUE, FALSE.	type: Boolean multiplicity: 1 isOrdered: N/A isUnique: N/A defaultValue: None isNullable: False
trainingDataQualityScore	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	It is the numerical value that represents the dependability/quality of a given decision generated by the AI/ML inference 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
expectedRuntimeContext	This describes the context where an MLEntity is expected to be applied or/and the RunTimeContext which is the context where the MLmodel or entity is being applied. allowedValues: NA	Type: MLContext multiplicity: 0..1 isOrdered: N/A isUnique: N/A defaultValue: None isNullable: False
trainingContext	This specify the context under which the MLEntity has been trained. allowedValues: NA	Type: MLContext multiplicity: 1 isOrdered: N/A isUnique: N/A defaultValue: None isNullable: False

Attribute Name	Documentation and Allowed Values	Properties
runTimeContext	This specifies the context where the MLmodel or entity is being applied. allowedValues: NA	Type: MLContext multiplicity: 1 isOrdered: N/A isUnique: N/A defaultValue: None isNullable: False
NOTE: When the performanceScore is to indicate the performance score for 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
notifyMOICreation	O	--
notifyMOIDeletion	O	--
notifyMOIAttributeValueChanges	O	--
notifyEvent	O	--

8 Service components

8.1 Service components for ML model training MnS

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

Table 8.1-1: Components for 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]).	MLTrainingFunction IOC; MLTrainingRequest IOC; MLTrainingReport IOC; MLTrainingProcess IOC.	N/A

9 Solution Set (SS)

The present document defines the following NRM Solution Set definitions for 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 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 MLEntity <<dataType>>
class MLTrainingFunction <<InformationObjectClass>>
class MLTrainingRequest <<InformationObjectClass>>
class MLTrainingReport <<InformationObjectClass>>
class MLTrainingProcess <<InformationObjectClass>>


ManagedEntity "1" *-- "*" MLTrainingFunction: <<names>>
MLTrainingFunction "1" -d-> "*" MLEntity
MLTrainingFunction "1" *-- "*" MLTrainingProcess: <<names>>
MLTrainingFunction "1" *-- "*" MLTrainingRequest: <<names>>
MLTrainingFunction "1" *-- "*" MLTrainingReport: <<names>>

MLTrainingProcess "1" <-r-> "1" MLTrainingReport
MLTrainingReport "1" --> "1" MLTrainingReport
MLTrainingRequest "*" -l-> "1" MLEntity
MLTrainingRequest "*" -r-> "*" MLTrainingProcess

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

@enduml
```

A.3 PlantUML code for Figure 7.2.2-1: Inheritance Hierarchy for 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 MLTrainingFunction <<InformationObjectClass>>
class MLTrainingRequest <<InformationObjectClass>>
class MLTrainingProcess <<InformationObjectClass>>
class MLTrainingReport <<InformationObjectClass>>

ManagedFunction <|-- MLTrainingFunction
Top <|-- MLTrainingRequest
Top <|-- MLTrainingProcess
Top <|-- MLTrainingReport

@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 "TS28105_AiMlNrm.yaml"

```
openapi: 3.0.1
info:
  title: AI/ML NRM
  version: 17.6.0
  description: >-
    OAS 3.0.1 specification of the AI/ML NRM
    © 2023, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).
    All rights reserved.
externalDocs:
  description: 3GPP TS 28.105; AI/ML Management
  url: http://www.3gpp.org/ftp/Specs/archive/28_series/28.105/
paths: {}
components:
  schemas:

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

    MLEntityList:
      type: array
      items:
        $ref: '#/components/schemas/MLEntity'

    MLEntity:
      type: object
      properties:
        mLEntityId:
          type: string
        inferenceType:
          type: string
        mLEntityVersion:
          type: string
        expectedRunTimeContext:
          $ref: '#/components/schemas/MLContext'
        trainingContext:
          $ref: '#/components/schemas/MLContext'
        runTimeContext:
          $ref: '#/components/schemas/MLContext'

    MLContext:
      type: object
      properties:
        inferenceEntityRef:
          $ref: 'TS28623_ComDefs.yaml#/components/schemas/DnList'
        dataProviderRef:
          $ref: 'TS28623_ComDefs.yaml#/components/schemas/DnList'

    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 "MLTrainingProcess".
      type: object
      properties:
        mLTrainingProcessId:
          type: string
        status:
          type: string
          enum:
            - RUNNING
            - CANCELLING
            - CANCELLED
            - SUSPENDED
            - FINSHED
        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: 'TS28623_GenericNrm.yaml#/components/schemas/Top'
        - type: object
          properties:
            attributes:
              $ref: 'TS28623_GenericNrm.yaml#/components/schemas/SubNetwork-Attr'
        - $ref: 'TS28623_GenericNrm.yaml#/components/schemas/SubNetwork-ncO'
        - type: object
          properties:
            SubNetwork:
              $ref: '#/components/schemas/SubNetwork-Multiple'
            ManagedElement:
              $ref: '#/components/schemas/ManagedElement-Multiple'
            MLTrainingFunction:
              $ref: '#/components/schemas/MLTrainingFunction-Multiple'

    ManagedElement-Single:
      allOf:
        - $ref: 'TS28623_GenericNrm.yaml#/components/schemas/Top'
        - type: object
```

```
            properties:
              attributes:
                $ref: 'TS28623_GenericNrm.yaml#/components/schemas/ManagedElement-Attr'
          - $ref: 'TS28623_GenericNrm.yaml#/components/schemas/ManagedElement-ncO'
          - type: object
            properties:
              MLTrainingFunction:
                $ref: '#/components/schemas/MLTrainingFunction-Multiple'

    MLTrainingFunction-Single:
      allOf:
        - $ref: 'TS28623_GenericNrm.yaml#/components/schemas/Top'
        - type: object
          properties:
            attributes:
              allOf:
                - $ref: 'TS28623_GenericNrm.yaml#/components/schemas/ManagedFunction-Attr'
                - type: object
                  properties:
                    mLEntityList:
                      $ref: '#/components/schemas/MLEntityList'
        - $ref: 'TS28623_GenericNrm.yaml#/components/schemas/ManagedFunction-ncO'
        - type: object
          properties:
            MLTrainingRequest:
              $ref: '#/components/schemas/MLTrainingRequest-Multiple'
            MLTrainingProcess:
              $ref: '#/components/schemas/MLTrainingProcess-Multiple'
            MLTrainingReport:
              $ref: '#/components/schemas/MLTrainingReport-Multiple'

    MLTrainingRequest-Single:
      allOf:
        - $ref: 'TS28623_GenericNrm.yaml#/components/schemas/Top'
        - type: object
          properties:
            attributes:
              allOf:
                - type: object
                  properties:
                    mLEntityId:
                      type: string
                    candidateTrainingDataSource:
                      type: array
                      items:
                        type: string
                    trainingDataQualityScore:
                      type: number
                      format: float
                    trainingRequestSource:
                      $ref: 'TS28623_ComDefs.yaml#/components/schemas/Dn'
                    requestStatus:
                      $ref: '#/components/schemas/RequestStatus'
                    expectedRuntimeContext:
                      $ref: '#/components/schemas/MLContext'
                    performanceRequirements:
                      $ref: '#/components/schemas/PerformanceRequirements'
                    cancelRequest:
                      type: boolean
                    suspendRequest:
                      type: boolean

    MLTrainingProcess-Single:
      allOf:
        - $ref: 'TS28623_GenericNrm.yaml#/components/schemas/Top'
        - type: object
          properties:
            attributes:
              allOf:
                - type: object
                  properties:
                    mLTrainingProcessId:
                      type: string
                    priority:
                      type: integer
                    terminationConditions:
                      type: string
                    progressStatus:
```

```
                       $ref: '#/components/schemas/TrainingProcessMonitor'
                     cancelProcess:
                       type: boolean
                     suspendProcess:
                       type: boolean
                     trainingRequestRef:
                       $ref: 'TS28623_ComDefs.yaml#/components/schemas/DnList'
                     trainingReportRef:
                       $ref: 'TS28623_ComDefs.yaml#/components/schemas/Dn'


      MLTrainingReport-Single:
        allOf:
          - $ref: 'TS28623_GenericNrm.yaml#/components/schemas/Top'
          - type: object
            properties:
              attributes:
                allOf:
                  - type: object
                    properties:
                      mLEntityId:
                        type: string
                      areConsumerTrainingDataUsed:
                        type: boolean
                      usedConsumerTrainingData:
                        type: array
                        items:
                          type: string
                      modelConfidenceIndication:
                        type: integer
                      modelPerformanceTraining:
                        type: array
                        items:
                          $ref: '#/components/schemas/ModelPerformance'
                      areNewTrainingDataUsed:
                        type: boolean
                      trainingRequestRef:
                        $ref: 'TS28623_ComDefs.yaml#/components/schemas/DnList'
                      trainingProcessRef:
                        $ref: 'TS28623_ComDefs.yaml#/components/schemas/Dn'
                      trainingReportRef:
                        $ref: 'TS28623_ComDefs.yaml#/components/schemas/Dn'
                      lastTrainingRef:
                        $ref: 'TS28623_ComDefs.yaml#/components/schemas/Dn'

#-------- 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'
      MLTrainingFunction-Multiple:
        type: array
        items:
          $ref: '#/components/schemas/MLTrainingFunction-Single'
      MLTrainingRequest-Multiple:
        type: array
        items:
          $ref: '#/components/schemas/MLTrainingRequest-Single'
      MLTrainingProcess-Multiple:
        type: array
        items:
          $ref: '#/components/schemas/MLTrainingProcess-Single'
      MLTrainingReport-Multiple:
        type: array
        items:
          $ref: '#/components/schemas/MLTrainingReport-Single'


#-------- Definitions in TS 28.104 for TS 28.532 --------------------------------

      resources-AiMlNrm:
        oneOf:
          - $ref: '#/components/schemas/SubNetwork-Single'
```

```
            - $ref: '#/components/schemas/ManagedElement-Single'

            - $ref: '#/components/schemas/MLTrainingFunction-Single'
            - $ref: '#/components/schemas/MLTrainingRequest-Single'
            - $ref: '#/components/schemas/MLTrainingProcess-Single'
            - $ref: '#/components/schemas/MLTrainingReport-Single'
```

Annex C (informative):
Change history

Change history							
Date	Meeting	TDoc	CR	Rev	Cat	Subject/Comment	New version
2022-06	SA#96					Upgrade to change control version	17.0.0
2022-09	SA#97e	SP-220851	0003	-	F	Corrections to the terms and definition description and corresponding updates	17.1.0
2022-09	SA#97e	SP-220850	0004	1	F	fix incorrect yaml file name in TS28.105	17.1.0
2022-09	SA#97e	SP-220851	0005	1	F	Clarifications and corrections of Use cases	17.1.0
2022-09	SA#97e	SP-220851	0006	1	F	Clarifications and corrections into the Class definitions and Attribute properties	17.1.0
2022-09	SA#97e	SP-220851	0007	1	F	Correction and clarifications of the Requirements	17.1.0
2022-09	SA#97e					Alignment with content with FORGE	17.1.1
2022-12	SA#98e	SP-221166	0008	2	F	Adding missing attributes	17.2.0
2022-12	SA#98e	SP-221166	0009	-	F	Correction of stage 3 openAPI	17.2.0
2023-03	SA#99	SP-230193	0011	-	F	Adding the missing definition of attributes Stage 2 and Stage 3	17.3.0
2023-03	SA#99	SP-230193	0013	1	F	Correcting the attribute properties	17.3.0
2023-03	SA#99	SP-230193	0014	1	F	Correction of the Handling errors in data and ML decisions	17.3.0
2023-03	SA#99	SP-230193	0015	1	F	Correction of terminologies	17.3.0
2023-03	SA#99	SP-230193	0017	1	F	Correct AI/ML related terms	17.3.0
2023-03	SA#99	SP-230193	0018	1	F	Correct formatting and spelling errors	17.3.0
2023-03	SA#99	SP-230193	0019	1	F	Correct attribute definitions	17.3.0
2023-06	SA#100	SP-230655	0022	1	F	Correcting the attribute properties	17.4.0
2023-06	SA#100	SP-230649	0024	1	F	Grammatical Corrections	17.4.0
2023-06	SA#100	SP-230655	0030	-	F	Removal of SW loading from training phase	17.4.0
2023-06	SA#100	SP-230655	0031	1	F	Correction of the figure for ML training function	17.4.0
2023-09	SA#101	SP-230948	0034	-	F	Clarify ML models as proprietary	17.5.0
2023-09	SA#101	SP-230948	0038	1	F	Restore the wrongly voided clause "5 Service and functional framework"	17.5.0
2023-12	SA#102	SP-231467	0042	1	F	Correction on ModelPerformance	17.6.0
2023-12	SA#102	SP-231490	0044	-	F	Rel-17 CR TS 28.105 Corrections of ML training related use cases description	17.6.0
2023-12	SA#102	SP-231490	0046	-	F	Rel 17 CR TS 28.105 Remove unused decision entity term	17.6.0
2023-12	SA#102	SP-231490	0048	-	F	Rel 17 CR TS 28.105 Clarify the description of confidenceIndication attribute	17.6.0
2023-12	SA#102	SP-231467	0060	1	F	Rel-17 TS 28.105 Restore definition of attribute mLEntityList - Not implemented due to wrong baseline used	17.6.0
2023-12	SA#102	SP-231467	0062	1	F	CR TS 28.105 Rel-17 Correction of IOC name	17.6.0
2023-12	SA#102	SP-231467	0064	1	F	TS 28.105 Rel-17 Correction of attribute properties	17.6.0
2023-12	SA#102	SP-231490	0070	-	F	Rel 17 CR TS 28.105 Fix incorrect figure label	17.6.0
2022-09	SA#102					Alignment with content with FORGE	17.6.0
2022-09	SA#102	SP-231467	0067	1	F	Rel 17 CR TS 28.105 Resolve issues related to the usage of confidenceIndication attribute – Partially implemented (no revision mark for 1st Change)	17.6.0

History

Document history		
V17.0.0	July 2022	Publication
V17.1.1	October 2022	Publication
V17.2.0	January 2023	Publication
V17.3.0	April 2023	Publication
V17.4.0	July 2023	Publication
V17.5.0	September 2023	Publication
V17.6.0	January 2024	Publication