ITUPublications

International Telecommunication Union
Standardization Sector

ITU-T Technical Report

(03/2026)

QSTR.MEML

Methods and metrics for evaluating machine learning models in marketplace for future networks including IMT-2020

Technical Report ITU-T QSTR.MEML

Methods and metrics for evaluating machine learning models in marketplace for future networks including IMT-2020

Summary

Machine learning (ML) marketplace is a component which provides capabilities facilitating the exchange and delivery of ML models among multiple parties. Recommendation ITU-T Y.3176 specifies an architecture for ML marketplace integration in networks. This Technical Report studies methods and metrics to evaluate ML models in ML marketplace, which include an overview of ML marketplace in future networks including International Mobile Telecommunications-2020 (IMT-2020), and methods and metrics for evaluating ML models in ML marketplace.

Keywords

Machine learning marketplace, method and metric, model evaluation.

Note

This is an informative ITU-T publication. Mandatory provisions, such as those found in ITU-T Recommendations, are outside the scope of this publication. This publication should only be referenced bibliographically in ITU-T Recommendations.

Table of Contents

Technical Report ITU-T QSTR.MEML

Methods and metrics for evaluating machine learning models in marketplace for future networks including IMT-2020

1 Scope

This Technical Report studies methods and metrics to evaluate machine learning (ML) models in marketplace for future networks, including International Mobile Telecommunications-2020 (IMT-2020) and beyond, and covers the following aspects:

– Overview of ML marketplace;

– Methods for evaluating ML models in marketplace;

– Metrics for evaluating ML models in marketplace.

2 References

[ITU-T Y.3176] Recommendation ITU-T Y.3176 (2020), *Machine learning marketplace integration in future networks including IMT-2020.*

3 Definitions

3.1 Terms defined elsewhere

This Technical Report uses the following terms defined elsewhere:

3.1.1 machine learning (ML) [b-ITU-T Y.3172]: Processes that enable computational systems to understand data and gain knowledge from it without necessarily being explicitly programmed.

NOTE 1 – Definition adapted from [b-ETSI GR ENI 004].

NOTE 2 – Supervised machine learning and unsupervised machine learning are two examples of machine learning types.

3.1.2 machine learning function orchestrator (MLFO) [b-ITU-T Y.3172]: A logical node with functionalities that manage and orchestrate the nodes in a machine learning pipeline.

3.1.3 machine learning marketplace [ITU-T Y.3176]: A component which provides capabilities facilitating the exchange and delivery of machine learning models among multiple parties.

NOTE 1 – Examples of parties include suppliers and users of ML models. Capabilities provided to users of ML models include functionalities to find, learn about, deploy (or download) and use ML models. Capabilities provided to suppliers of ML models (e.g., data scientist) include functionalities to share (on-board, upload), describe (learn about) and market their ML models.

NOTE 2 – A network operator may use a machine learning marketplace deployed internally and/or externally to the network operator's administrative domains. Internal and external marketplaces differ only in the deployment perspective. A marketplace which is internal to a network operator may act as an external marketplace to another network operator and vice versa.

3.1.4 machine learning model [b-ITU-T Y.3172]: Model created by applying machine learning techniques to data to learn from.

NOTE 1 – A machine learning model is used to generate predictions (e.g., regression, classification, clustering) on new (untrained) data.

NOTE 2 – A machine learning model may be encapsulated in a deployable fashion in the form of a software (e.g., virtual machine, container) or hardware component (e.g., IoT device).

NOTE 3 – Machine learning techniques include learning algorithms (e.g., learning the function that maps input data attributes to output data).

3.1.5 machine learning pipeline [b-ITU-T Y.3172]: A set of logical nodes, each with specific functionalities, that can be combined to form a machine learning application in a telecommunication network.

NOTE – The nodes of a machine learning pipeline are entities that are managed in a standard manner and can be hosted in a variety of network functions [b-ITU-T Y.3100].

3.1.6 machine learning sandbox [b-ITU-T Y.3172]: An environment in which machine learning models can be trained, tested and their effects on the network evaluated.

NOTE – A machine learning sandbox is designed to prevent a machine learning application from affecting the network, or to restrict the usage of certain machine learning functionalities.

3.2 Terms defined in this Technical Report

None.

4 Abbreviations and acronyms

This Technical Report uses the following abbreviations and acronyms:

AI	Artificial Intelligence
CMC	Cumulative Matching Characteristics
CPU	Central Processing Unit
FLOP	Floating-Point Operation
GPU	Graphics Processing Unit
IMT-2020	International Mobile Telecommunications-2020
IoT	Internet of Things
KPI	Key Performance Indicator
MAP	Mean Average Precision
MEC	Multi-access Edge Computing
ML	Machine Learning
MLDB	Machine Learning Database
MLFO	Machine Learning Function Orchestrator
NPU	Neural Processing Unit
QoS	Quality of Service
RAG	Retrieval-Augmented Generation

5 Conventions

In this Technical Report, potential requirements which are derived from a given use case, are classified as follows:

The keywords "it is expected" indicate a consideration which would be important but not absolutely necessary to be fulfilled (e.g., by an implementation). Thus, such consideration would not need to be enabled to provide complete benefits of the use case.

6 Overview of ML marketplace in future networks

In the context of future networks, including IM-2020, an ML marketplace refers to a platform that enables the exchange, discovery and deployment of ML models among multiple stakeholders such as

network operators, service providers and model developers. [ITU-T Y.3176] specifies the architecture for ML marketplace integration in future networks, as shown in Figure 1. It includes an ML marketplace hosted internally to the network operator's administrative domains which optionally interface with another ML marketplace hosted externally to the network operator's administrative domains.

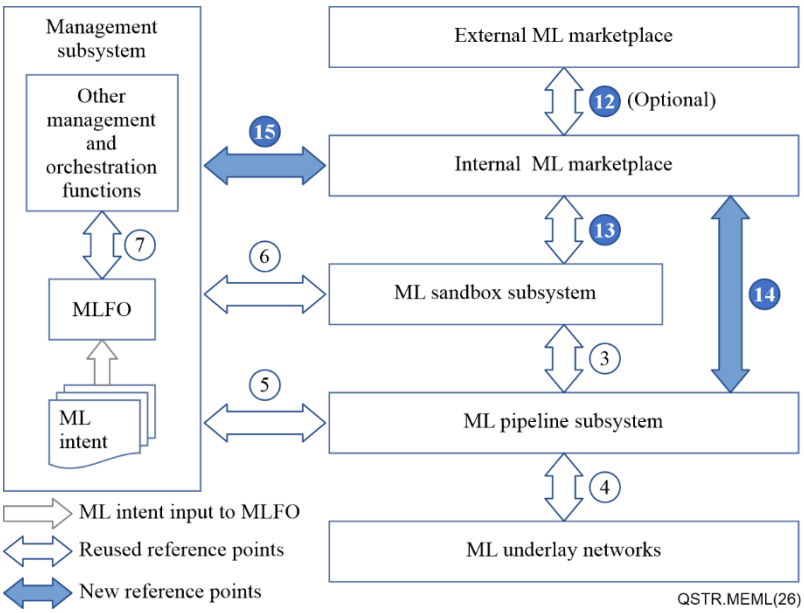

Figure 1 – Architecture for ML marketplace integration

The main functions of the internal marketplace include query, select, push, discover and deploy ML models by machine learning function orchestrator (MLFO). Model query refers to the MLFO sending a query to the ML marketplace for a particular set of ML models. Model selection refers to the MLFO selecting the ML model from the ML marketplace for deployment in the ML pipeline subsystem. Model discovery refers to the MLFO searching the updated versions of ML models that are already deployed in the ML pipeline subsystem from the ML marketplace. Model deployment refers to the MLFO requesting deploying the ML model in the ML pipeline subsystem.

ML models deployed in the ML pipeline subsystem forms ML application in the network. Therefore, monitoring and evaluating ML models in ML marketplace is prior to applying ML models to the network. The evaluation of ML models within ML marketplaces is a critical enabler for their trustworthy and effective integration into future networks, including IMT-2020. Its significance can be summarized as follows:

– Standardized evaluation ensures that operators can rely on objective and verifiable information about model performance, maturity and compliance;
– Evaluation provides a common basis for comparing models from different vendors or marketplaces;
– By assessing latency, accuracy, scalability and robustness, evaluation guarantees that models meet the stringent key performance indicators (KPIs) required in IMT-2020 and beyond environments.

7 Standardization activities

7.1 ITU-T

Significant progress has been achieved in the development of ITU-T Recommendations addressing ML and artificial intelligence (AI) in future networks including IM-2020. These Recommendations

collectively establish a coherent framework that spans architectural foundations, data handling, marketplace integration, model evaluation and monitoring.

[b-ITU-T Y.3172] laid the architectural foundation for ML in future networks including IMT-2020. It introduced the concept of an ML pipeline composed of logical nodes, MLFOs and the ML sandbox. It defined the architectural framework, reference points and functional components required to enable ML integration into telecommunication networks.

[b-ITU-T Y.3174] addressed the framework for data handling to enable ML in future networks. Recognizing that data is the critical enabler for ML performance, it specified requirements for data collection, storage, sharing and protection. It introduced the concept of the machine learning database (MLDB) and defined mechanisms for ensuring data quality, interoperability and compliance with privacy and security requirements.

[ITU-T Y.3176] extended the architectural framework by specifying high-level requirements and architecture for integrating ML marketplaces into future networks. It defined standardized metadata for ML models, requirements for marketplace functionalities and new reference points for interaction between internal and external marketplaces, ML sandboxes and ML pipelines. It emphasized the importance of interoperability, federation of marketplaces and lifecycle management of ML models. It enabled network operators to access, evaluate and deploy ML models from diverse sources while ensuring alignment with network KPIs and operational constraints.

[b-ITU-T Y.3179] further advanced the standardization of ML in networks by focusing on the architectural framework for ML model serving. It specified high-level requirements and architecture for preparing, optimizing and deploying ML models in heterogeneous environments to enable inference in future networks including IMT-2020.

[b-ITU-T Q.4081] addresses methods and metrics for monitoring ML/AI in future networks including IMT-2020. It emphasizes continuous monitoring during operation. It defines methods for monitoring input data, model performance, model drift, resource utilization and feedback loops. It also introduces a comprehensive set of functional and operational monitoring metrics, including statistical distance measures, prediction drift indicators and resource consumption metrics.

In addition, ITU-T has developed Recommendations under the F-series that focus on the evaluation and measurement of specific ML/AI technologies, such as deep learning software frameworks, image-based re-identification algorithms, foundation models, AI agents and retrieval-augmented generation (RAG) systems.

[b-ITU-T F.748.33] defines a benchmark framework, metrics, dataset requirements and evaluation processes for image-based re-identification algorithms. It introduces both on-site and off-site benchmarking approaches and specifies reliability and temporality indicators, such as cumulative matching characteristics (CMC) and mean average precision (MAP), to ensure fair and comprehensive evaluation in practical scenarios.

[b-ITU-T F.748.44] focuses on the evaluation requirements and methods for foundation models. It covers testing capabilities, datasets, methods and tools, addressing aspects such as understanding, generation, reasoning, knowledge, reliability and robustness. It provides guidelines for automated and manual evaluation to ensure consistency, transparency and comparability across different foundation models.

[b-ITU-T F.748.46] establishes a framework of technical requirements and evaluation methods for AI agents built on large-scale pre-trained models. It covers four key capabilities: perception and cognition, planning, memory and execution. Evaluation methods include both performance superiority and functional completeness assessments across various tasks and environments.

[b-ITU-T F.748.52] provides a framework for evaluating RAG systems based on large-scale pre-trained models. It addresses dataset construction, technical capabilities (retrieval, generation,

optimization) and application capabilities (maturity, stability). It also defines grading requirements for answer relevance, faithfulness, truthfulness and contextual coherence.

7.2 ISO/IEC

ISO/IEC has made significant progress in developing standards that address the evaluation and measurement of ML and AI systems. These standards complement ITU-T's work by focusing on classification performance, system impact assessment and robustness verification of neural networks.

[b-ISO/IEC TS 4213] defines methods for assessing the performance of ML classification systems. It establishes a standardized framework for measuring accuracy, precision, recall, F1-score and other key indicators. By providing consistent evaluation processes, the specification ensures comparability across different ML models and supports both industrial and academic benchmarking activities.

[b-ISO/IEC 42005] introduces a framework for AI system impact assessment. It emphasizes the need to evaluate AI systems not only from a technical perspective but also in terms of social, ethical and organizational impacts. The standard provides structured methodologies for identifying risks, benefits and broader consequences of AI deployment, supporting governance, compliance and responsible innovation.

[b-ISO/IEC 24029-2] focuses on assessing the robustness of neural networks using formal methods. It specifies methodologies for applying formal verification techniques to evaluate the stability and reliability of neural networks under adversarial conditions or uncertain inputs.

7.3 IEEE

IEEE has developed standards that address the evaluation of fairness in ML systems and the technical requirements for knowledge graph evaluation. These standards complement ITU-T and ISO/IEC efforts by focusing on ethical dimensions of ML performance and the structural assessment of AI knowledge representation.

[b-IEEE 3198] establishes methods for evaluating the fairness of ML systems. It defines categories of potential unfairness, including bias in data, algorithms and outcomes, and provides standardized approaches for measuring fairness across different contexts. The standard also outlines mitigation strategies, ensuring that ML systems can be assessed and improved to meet ethical and societal expectations. It is particularly relevant for applications where fairness and non-discrimination are critical, such as healthcare, finance and public services.

[b-IEEE 2807.1] specifies technical requirements and evaluation methods for knowledge graphs. It provides a framework for assessing the quality, consistency and usability of knowledge graphs in AI systems. The evaluation criteria include structural integrity, semantic accuracy, scalability and interoperability. By defining standardized metrics and processes, the standard supports the reliable deployment of knowledge graphs in applications such as search engines, recommendation systems and intelligent agents.

8 Methods of evaluating ML models in marketplace for future networks

This clause specifies the methodology and process for evaluating ML models in the context of ML marketplaces for future networks including IMT-2020. The evaluation methodology ensures that parameters described in this Technical Report are consistently applied at each stage, enabling transparent, repeatable and interoperable assessment of ML models.

Figure 2 illustrates the systematic, multi-stage process for evaluating ML models within ML marketplaces for future networks including IMT-2020. The process is designed to ensure that ML models are thoroughly assessed before deployment, enhancing their reliability, performance and alignment with network requirements.

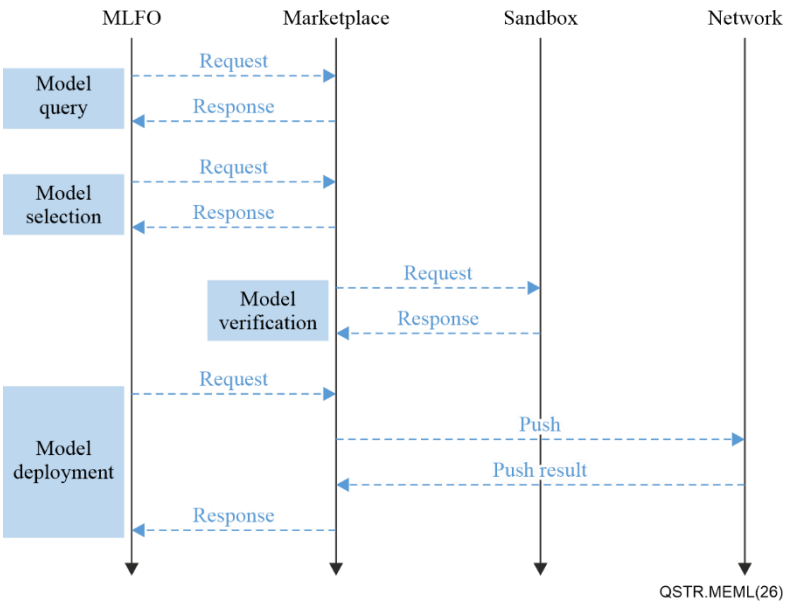

Figure 2 – Process of evaluating ML models in ML marketplace for future networks

The process begins with the model query phase, where the MLFO submits queries to the marketplace to discover available models based on metadata such as function type, input/output specifications and application domain. This phase ensures that only relevant and appropriately described models proceed to the next stage. Following discovery, the model selection phase involves comparing candidate models against KPIs, such as accuracy, latency, computational load and resource requirement, to identify the most suitable models for the intended use case. Selected models then undergo model verification, where they are tested in a controlled ML sandbox environment that simulates real network conditions. This step validates model behaviour, robustness and compliance with network KPIs without impacting live operations. Finally, in the model deployment phase, the MLFO orchestrates the integration of verified models into the ML pipeline subsystem.

8.1 Model query

The model query function allows network operators or MLFOs to discover ML models available in the marketplace by submitting queries based on metadata. The purpose of this operation is to enable the efficient discovery of candidate models that match an operator's requirements. In this step, MLFO sends a model query to marketplace and gets corresponding response, which is performed with interface 15 in [ITU-T Y.3176].

For evaluation, it is expected that:

– the completeness and correctness of model metadata is verified, including attributes such as model name, version, author, input/output specifications and runtime environment;

– the accuracy and relevance of the query response is evaluated to ensure that the returned models correspond to the specified search criteria;

– the query mechanism is assessed for compliance with the required metadata formats and interfaces defined in [ITU-T Y.3176].

8.2 Model selection

Model selection refers to the process of choosing one or more models from the query results for further validation and/or deployment. The purpose of this operation is to enable operators to identify the most suitable models for their intended use cases. MLFO conducts model selection in order to select proper models from the query response.

For evaluation, it is expected that:

– the performance indicators of candidate models, such as accuracy, latency and throughput, are assessed against operator requirements;
– the resource requirements of the models, including central processing unit (CPU)/graphics processing unit (GPU) utilization, memory footprint and bandwidth demand, are evaluated to ensure compatibility with the target deployment environment;
– the compatibility of the models with chaining mechanisms and pipeline integration is verified;
– the alignment of the selected models with operator intent and network KPIs is confirmed.

8.3 Model verification

[ITU-T Y.3176] specifies that before deployment, models may be tested in an ML sandbox environment. The purpose of this operation is to validate the model's behaviour in a controlled setting without impacting the live network. Sandbox provides a virtual network environment, runs the selected models and monitors the status of the models.

For evaluation, it is expected that:

– the maturity of the model is assessed, including compliance with standard test cases and benchmarks;
– the robustness of the model is evaluated against potential issues such as data drift, adversarial inputs and stress conditions;
– the model's outputs are validated against network KPIs, such as latency, reliability and scalability;
– the observability of the model during runtime is confirmed, ensuring that sufficient logs and monitoring data are available for analysis.

8.4 Model deployment

Model deployment is coordinated by the MLFO, which manages the integration of validated models into the ML pipeline subsystem. The purpose of this operation is to ensure that models are deployed in a manner consistent with operator policies, network conditions and lifecycle management requirements. The MLFO decides which model is the final result after aggregating data from the above steps.

For evaluation, it is expected that:

– the adaptability of the model to different deployment environments (e.g., virtual machines, containers, multi-access edge computing (MEC) platforms or edge devices) is verified;
– the lifecycle management of the model, including version control, rollback and update mechanisms, is assessed;
– the synchronization of deployment operations with network events and operator policies is confirmed, to ensure seamless integration.

9 Metrics of evaluating ML models in ML marketplace for future networks

This clause defines concrete and measurable parameters for evaluating ML models in ML marketplace for future networks including IMT-2020 and beyond. Parameters are organized by operation: model query, model selection, model validation in the sandbox and model deployment with the MLFO.

9.1 Metrics for model query

In the model query stage, the focus is on identifying the functional attributes of ML models available in the marketplace. The parameters describe the model's intended function, input and output data types, and application domain.

The detailed definitions of the metrics are given below.

a) Model function type
   i) Unit: category (string/enum)
   ii) Description: identifies the primary functionality of the model, such as image recognition, anomaly detection or traffic prediction.
   iii) Measurement method: declared in model metadata and validated against a standardized taxonomy.

b) Input data type
   i) Unit: category
   ii) Description: specifies the type of input data the model accepts, such as images, text or network KPIs.
   iii) Measurement method: declared in metadata and validated against a predefined metadata schema.

c) Output type
   i) Unit: category
   ii) Description: specifies the type of output generated by the model, such as classification labels or regression values.
   iii) Measurement method: declared in metadata and verified by test inference.

d) Application domain
   i) Unit: category
   ii) Description: indicates the intended application area, such as configuration management, network monitoring, or quality of service (QoS) optimization.
   iii) Measurement method: declared in metadata and mapped to standardized use case categories.

9.2 Metrics for model selection

In the model selection stage, the operator compares candidate models of the same functional type and selects the most suitable one. The parameters focus on performance and resource requirements.

Their detailed definitions are given below.

a) Accuracy
   i) Unit: %
   ii) Description: the proportion of correct predictions relative to the total predictions.
   iii) Measurement method: calculated as the number of correct predictions divided by the total number of predictions, expressed as a percentage.

b) Latency
   i) Unit: ms
   ii) Description: the average time taken to produce an inference result for a single request.
   iii) Measurement method: measured during benchmark tests under defined conditions.

c) Batch size

i) Unit: number

ii) Description: the number of samples processed simultaneously in one inference batch, which affects throughput and latency.

iii) Measurement method: declared in model configuration or benchmarked during testing.

d) Model size

i) Unit: byte

ii) Description: the storage size of the serialized model file.

iii) Measurement method: obtained directly from the model file metadata.

e) Parameter count

i) Unit: number

ii) Description: the total number of trainable parameters in the model, reflecting its complexity.

iii) Measurement method: extracted from the model architecture definition.

f) Computational load

i) Unit: floating-point operation (FLOP)

ii) Description: the amount of computation required to execute the model.

iii) Calculation: derived from the number of FLOPs per inference.

g) Memory footprint

i) Unit: byte

ii) Description: the memory required to load the model and maintain its runtime state.

iii) Measurement method: measured during sandbox execution.

h) Bandwidth requirement

i) Unit: bps

ii) Description: the network bandwidth required for input and output data transfer.

iii) Measurement method: measured by monitoring data transfer per inference.

9.3 Metrics for model verification

In the model verification stage, the selected model is tested in a sandbox environment to validate its compatibility with the target network and computing environment. The detailed definitions are given below.

a) CPU model

i) Unit: string

ii) Description: the specific CPU architecture required for stable operation.

iii) Measurement method: declared in deployment metadata or verified during sandbox profiling.

b) CPU frequency

i) Unit: GHz

ii) Description: the minimum CPU clock speed required to achieve target performance.

iii) Measurement method: measured during sandbox profiling.

c) CPU cores

i) Unit: number of cores

ii) Description: the number of CPU cores required for stable operation.

iii) Measurement method: measured during sandbox profiling.

d) AI accelerator type

i) Unit: string

ii) Description: the type of AI accelerator card required, such as GPU or neural processing unit (NPU).

iii) Measurement method: declared in deployment metadata.

e) AI accelerator count

i) Unit: number

ii) Description: the number of accelerator cards required for stable operation.

iii) Measurement method: measured during sandbox profiling.

f) AI accelerator memory

i) Unit: byte

ii) Description: the on-board memory capacity of the accelerator required for inference.

iii) Measurement method: measured during sandbox profiling.

g) System memory

i) Unit: byte

ii) Description: the host system memory required to load and run the model.

iii) Measurement method: measured during sandbox profiling.

h) Network bandwidth

i) Unit: bps

ii) Description: the network bandwidth required for stable operation in sandbox.

iii) Measurement method: measured during sandbox profiling.

9.4 Metrics for model deployment

In the model deployment stage, the focus is on evaluating how the ML model behaves when integrated into the network environment under the coordination of the MLFO. The metrics emphasize operational sustainability and the reliability of the deployed model. Their detailed definitions are given below.

a) Energy consumption

i) Unit: joules per inference or watts

ii) Description: represents the energy consumed by the model during operation, either per inference or as average power draw. This reflects the sustainability of the deployed model.

iii) Measurement method: measured through hardware monitoring during sandbox execution.

b) Long-term stability

i) Unit: hours

ii) Description: indicates the average time between failures of the deployed model, reflecting its operational reliability over extended periods.

iii) Measurement method: measured through long-duration testing in the sandbox environment.

c) Service availability

i) Unit: %

ii) Description: the percentage of time the model service remains available and responsive to requests, reflecting its reliability in production.

iii) Measurement method: calculated based on uptime during long-duration sandbox testing.

d) Fault recovery time

i) Unit: seconds

ii) Description: the time required for the model service to recover and resume normal operation after a failure or crash.

iii) Measurement method: measured through controlled fault injection testing in the sandbox environment.

e) Adversarial robustness

i) Unit: score

ii) Description: evaluates the ability of the deployed model to resist adversarial inputs or malicious perturbations that attempt to degrade performance or compromise security.

iii) Measurement method: evaluated through adversarial testing scenarios in the sandbox environment.

f) Verified concurrent users

i) Unit: number

ii) Description: the maximum number of concurrent requests supported without performance degradation.

iii) Measurement method: determined by stress testing in sandbox.

g) Verified accuracy

i) Unit: %

ii) Description: the accuracy of the model measured in the sandbox environment.

iii) Measurement method: evaluated with representative datasets.

h) Verified latency

i) Unit: ms

ii) Description: the latency of the model measured in the sandbox environment.

iii) Measurement method: benchmark under sandbox load.

10 Conclusion and future work

This Technical Report presents methods and parameters for evaluating ML models in ML marketplaces for future networks including IMT-2020. By organizing evaluation metrics across the four key operational stages – model query, model selection, model verification in sandbox and model deployment with MLFO – it provides a framework to ensure that ML models can be discovered, compared, validated and deployed in an interoperable and trustworthy manner.

This Technical Report establishes a foundation for evaluating ML models in marketplaces but further refinement and extension of the framework are needed. Future work can expand the scope of metrics to include parameters that reflect user experience, fairness, explainability and the sustainability of ML models. In addition, cross-domain interoperability must be addressed to support federated ML marketplaces, ensuring that evaluation results are portable and comparable across different administrative domains. Finally, stronger integration with monitoring frameworks is encouraged, linking pre-deployment evaluation with post-deployment monitoring, such as that defined in [b-ITU-T Q.4081] to create a closed-loop lifecycle management system.

Bibliography

[b-ITU-T F.748.33] Recommendation ITU-T F.748.33 (2024), *Metrics and evaluation methods for image-based re-identification algorithm.*

[b-ITU-T F.748.44] Recommendation ITU-T F.748.44 (2025), *Assessment criteria for foundation models – Benchmark.*

[b-ITU-T F.748.46] Recommendation ITU-T F.748.46 (2025), *Requirements and evaluation methods of artificial intelligence agents based on large scale pre-trained models.*

[b-ITU-T F.748.52] Recommendation ITU-T F.748.52 (2025), *Requirements and evaluation methods for retrieval augmented generation of large scale pre-trained models.*

[b-ITU-T Q.4081] Recommendation ITU-T Q.4081 (2026), *Methods and metrics for monitoring machine learning/artificial intelligence in future networks including IMT-2020.*

[b-ITU-T Y.3100] Recommendation ITU-T Y.3100 (2017), *Terms and definitions for IMT-2020 network.*

[b-ITU-T Y.3172] Recommendation ITU-T Y.3172 (2019), *Architectural framework for machine learning in future networks including IMT-2020.*

[b-ITU-T Y.3174] Recommendation ITU-T Y.3174 (2020), *Framework for data handling to enable machine learning in future networks including IMT-2020.*

[b-ITU-T Y.3179] Recommendation ITU-T Y.3179 (2021), *Architectural framework for machine learning model serving in future networks including IMT-2020.*

[b-ETSI GR ENI 004] ETSI GR ENI 004 V1.1.1 (2018), *Experiential Networked Intelligence (ENI); Terminology for Main Concepts in ENI.*

[b-IEEE 2807.1] IEEE 2807.1-2024, *IEEE Standard for Technical Requirements and Evaluating Knowledge Graphs*.

[b-IEEE 3198] IEEE 3198-2025, *IEEE Standard for Evaluation Method of Machine Learning Fairness*.

[b-ISO/IEC 24029-2] ISO/IEC 24029-2:2023, *Artificial intelligence (AI) – Assessment of the robustness of neural networks Part 2: Methodology for the use of formal methods*.

[b-ISO/IEC 42005] ISO/IEC 42005:2025, *Information technology – Artificial intelligence (AI) – AI system impact assessment*.

[b-ISO/IEC TS 4213] ISO/IEC TS 4213:2022, *Information technology – Artificial intelligence - Assessment of machine learning classification performance*.