Machine Learning Sandbox for future networks including IMT-2020: requirements and architecture framework

Posted as ML5G-I-238-R1

Authors: Francesc Wilhelmi (UPF)

Vishnu Ram (Independent Research Consultant)

Presenter: Francesc Wilhelmi (UPF)
francisco.wilhelmi@upf.edu

1. Introduction

Problem

- Lack of **reliability** of ML algorithms (black-box systems)
  - Unexplainable operation ≈ Unpredictable behaviour
  - Exploration-based approaches
  - Issues related training data (e.g., limited datasets, noisy data, unseen situations)
  - Competitive settings

Sandbox

- Isolated domain to **train/test/evaluate** ML models before being applied to production
- To provide *trustworthy* ML-assisted applications for future networks
- Simulator, Emulator, Model, Testbed …

Scope

- Requirements and derived architectural aspects for ML Sandbox
- Interfaces to allow the manageability of the simulation, the execution of test cases, and the evaluation of ML models.

2. Requirements

- Up to 24 requirements
- Split into the following categories:
  - Simulated ML underlay requirements
  - ML operations requirements
  - Communication requirements
  - Metadata requirements
- Aligned with other documents:
  - Y.3172
  - Y.3173
  - Y.3174
  - Y.Supp55
  - Y.ML-IMT2020-MP, ML5G-I-248, ML5G-I-227-R2

REQ-ML-SANDBOX-001: ML Sandbox is required to simulate heterogeneous sources of data and SINKs of ML output.

REQ-ML-SANDBOX-002: ML Sandbox is required to support the dynamic instantiation of new simulated SRCs and/or SINKs.

…

REQ-ML-SANDBOX-008: ML Sandbox is required to support multiple ML pipelines, which may be chained and which potentially are interfaced with simulators from different levels of the network.

…

REQ-ML-SANDBOX-017 : ML Sandbox is required to support data handling reference points towards technology-specific simulated ML underlays.

3. High-level Architecture

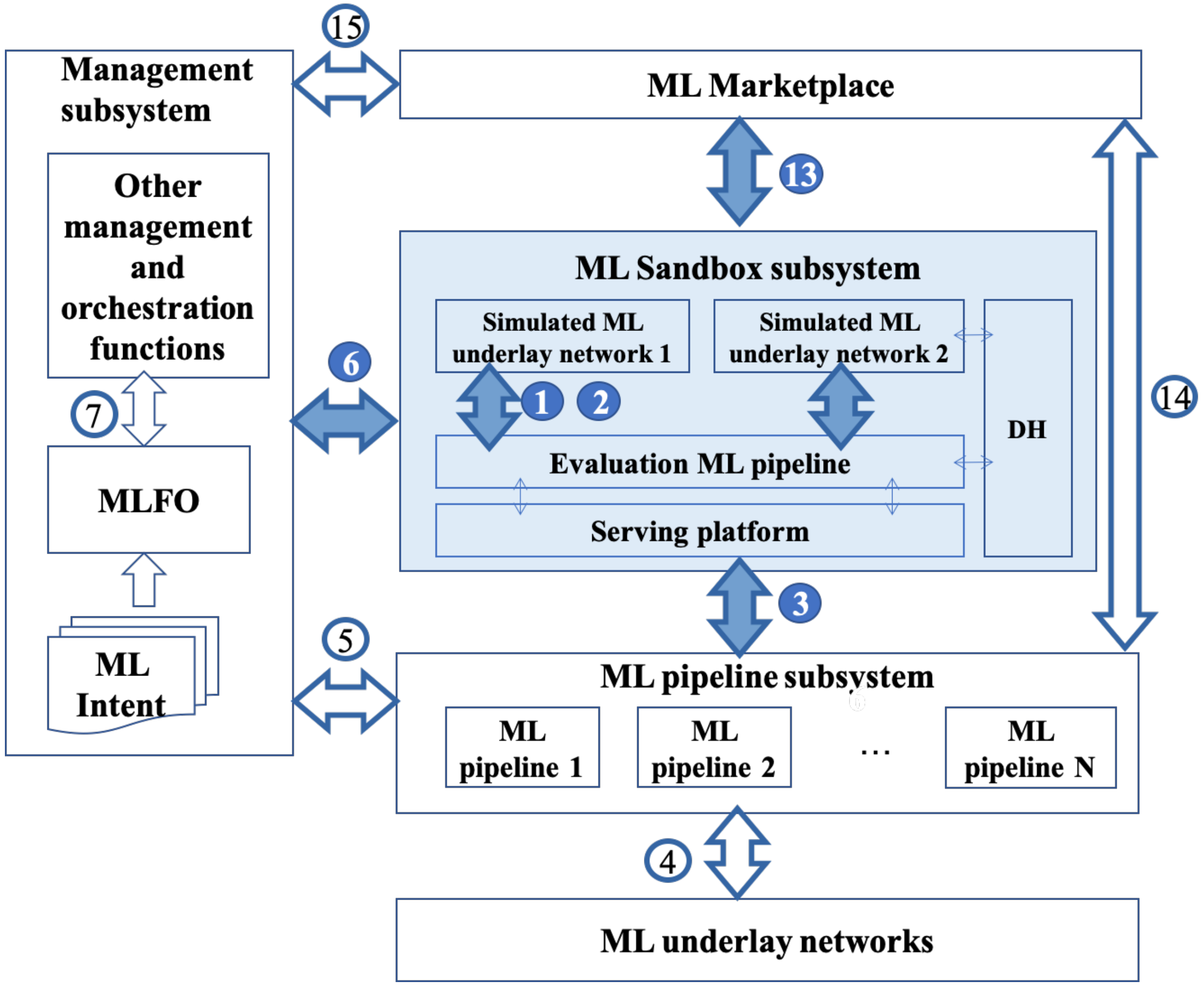

Figure 1: Overview of the main functional components and reference points of the ML Sandbox subsystem within the logical ML-aware architecture.

3.1 High-level Architecture
Components

- Simulated ML underlay
  - Simulation designer
  - Simulation composer
  - Evaluation plugin
  - Simulation post-processor
- Evaluation ML pipeline
- Data handling components
- Monitoring agent

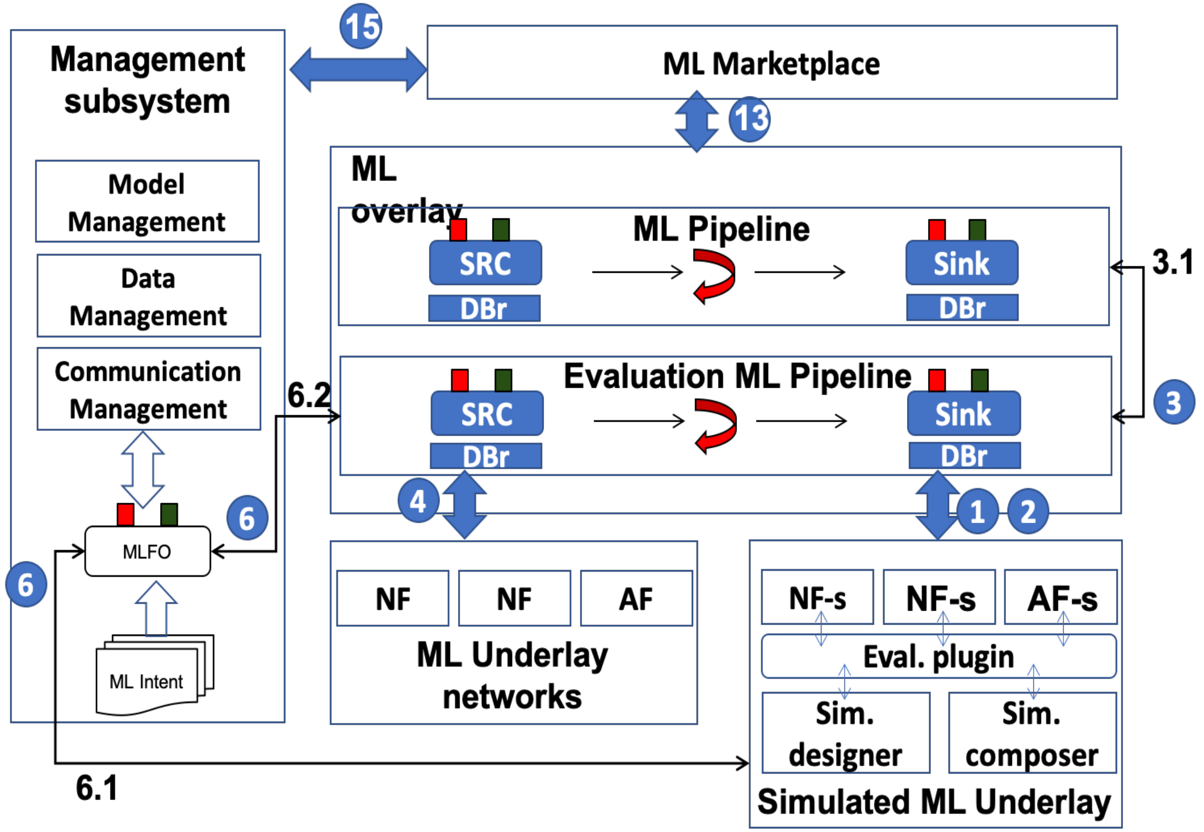

Figure 2: High-level architecture of the ML Sandbox components within the ML architecture.

3.2 High-level Architecture
Sequence diagrams

1. Capability discovery/negotiation for third-party simulation components
2. Monitor health
3. Validate input/output data
4. MLFO-triggered operations
5. Sandbox asynchronous messages

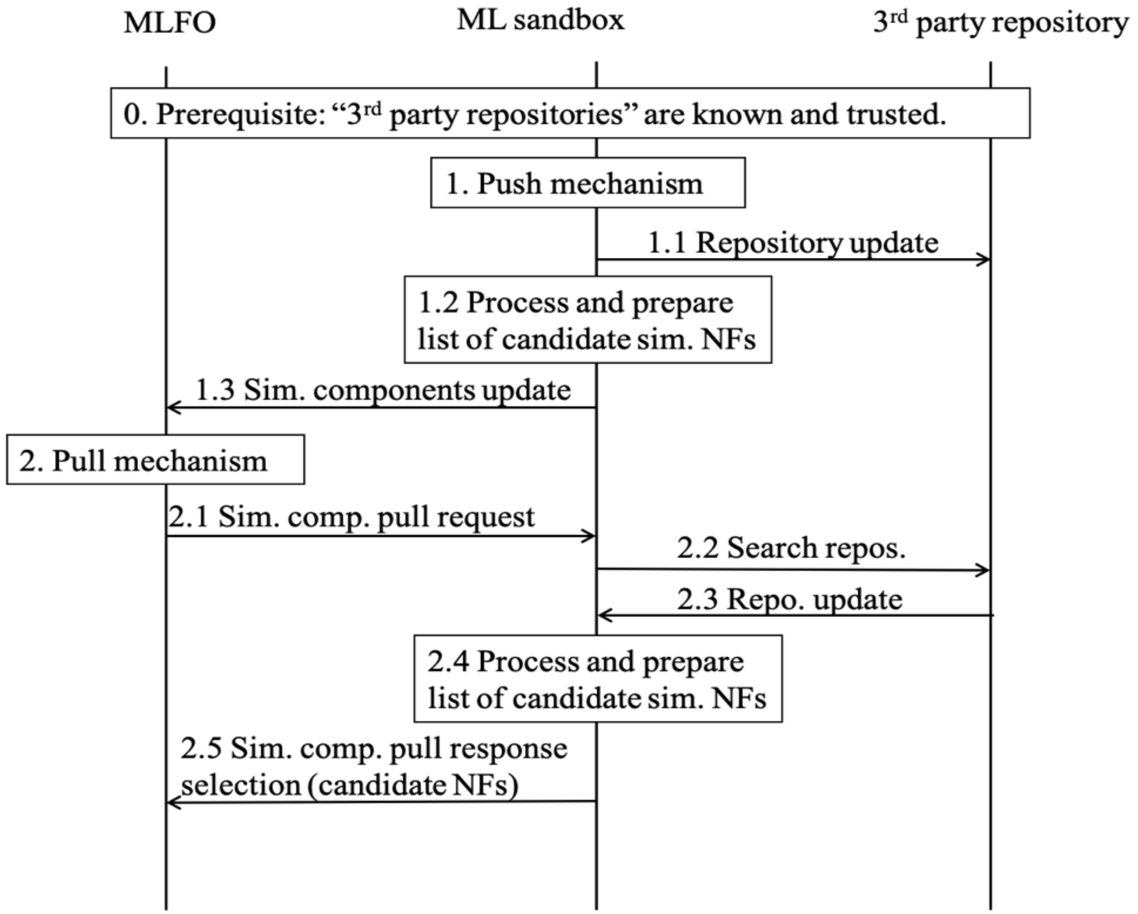

Figure 3: Sequence diagram of Capability discovery/negotiation for third-party simulation components

3.3 High-level Architecture
MLFO-triggered operations

Code	Parameters	Time sync / dependencies	Description
SANDBOX-TRIGGER-001	Request type, ML profile	Time sync = yes (evaluation blocks)	Request to prepare the simulation environment (both simulated ML underlay and evaluation ML pipeline) to train, test, and evaluate ML models in the ML sandbox.
SANDBOX-TRIGGER-002	Request type, validation type, acceptance criteria	Time sync = yes (sanity checklist, test suite, etc.)	Request to validate (sanity check) the deployed simulation environment. Used by the MLFO to determine whether to take action to fix potential deployment issues, proceed with ML model evaluation in the sandbox, etc.
SANDBOX-TRIGGER-003	Request type, update type, updated ML profile	Time sync = yes (updated simulation components)	Request to modify/update the simulated ML underlay according to updates in policies, changes in the live ML underlay, potential failures of previous simulated functions, etc.
SANDBOX-TRIGGER-004	Request type, ML profile, ML model output	Time sync = yes (evaluation results)	Request to evaluate the impact of the output of an ML model in the simulated ML underlay, so that some insights can be provided before applying that output on the live ML underlay.
SANDBOX-TRIGGER-005	Request type, ML profile	Time sync = no	Request to train an ML model in the ML sandbox.

3.4 High-level Architecture
Sandbox async. messages

Code	Parameters	Conditions	Description
SANDBOX-ASYNC-001	Status code / Error code / detailed report (conditional)	Threshold-based alert is fired / miss-behavior is detected / keep-alive message	Report the health status of the simulation components
SANDBOX-ASYNC-002	Update type / changelog / additional information on implications of update	Updated on simulation components is notified/discovered to/by the ML Sandbox	Report an update on security, accounting, licensing requirements of simulation components
SANDBOX-ASYNC-003	Alert type / forecasted results / additional information on potential failure points	Threshold-based alert based on trend analysis (e.g., service at risk)	Proactive behavior trend identification of simulation components and sandbox resources
SANDBOX-ASYNC-004	Report type / Updated simulation component metadata	Change on simulation component is noticed	Report an update on simulation component metadata
SANDBOX-ASYNC-005	Report type / Updated level of intelligence of simulation components	Change on the intelligence level of a simulation component	Report the simulation environment intelligence level

4. Extra

- *“Usage of Network Simulators in Machine-Learning-Assisted 5G/6G Networks”*
- Magazine paper submitted to IEEE Wireless Comm. Magazine
    - https://arxiv.org/pdf/2005.08281.pdf
- More details on simulated ML underlay (ITU-T approach)
- Use case testbed implementation

Usage of Network Simulators in Machine-Learning-Assisted 5G/6G Networks

Francesc Wilhelmi, Marc Carrascosa, Cristina Cano, Anders Jonsson, Vishnu Ram, and Boris Bellalta

***Abstract*—Without any doubt, Machine Learning (ML) will be an important driver of future communications due to its foreseen performance in front of complex problems. However, the application of ML to networking systems raises concerns among network operators and other stakeholders, especially regarding trustworthiness and reliability. In this paper, we devise the role of network simulators for bridging the gap between ML and communications systems. Network simulators can facilitate the adoption of ML-based solutions by means of training, testing, and validating ML models before being applied to an operative network. Finally, we showcase the potential benefits of integrating network simulators into ML-assisted communications through a proof-of-concept testbed implementation of a residential Wi-Fi network.**

***Index Terms*—Future Networks, ITU, Network Simulator, Machine Learning, Wireless Local Area Networks**

I. Introduction

Beyond the fifth-generation (5G) of mobile communications systems, namely the sixth generation (6G), Artificial Intelligence (AI), and more precisely Machine Learning (ML), are expected to be pervasively included as part of the network operation, which would entail a huge leap towards optimization, automation, and self-healing. This is possible thanks to the paradigm shift driven by the softwarization of networks – achieved through Software Defined Networks (SDN) and Network Function Virtualization (NFV) – which provides the necessary flexibility to empower data-driven approaches.

The integration of ML to communications has started to be considered for the upcoming versions of 5G. This fact is

use cases comprise heterogeneous scenarios with mobility, a huge number of devices, and high-bandwidth and low-latency requirements. In particular, ML may offer substantial performance gains due to the inherent flexibility of automatically learning diverse situations, thus allowing to solve problems related to interference management, improving spatial reuse, or efficient resource allocation.

While ML promises significant productivity gains, it also raises serious challenges and concerns. First of all, the successful application of ML depends on the quality of the training data provided. These data, by nature, can often be limited or noisy, and draw insightful conclusions might be challenging for many problems. Apart from that, dealing with non-stationary data is still an open challenge, which casts doubts on the validity of potentially learned models. A prominent example is that of IEEE 802.11 Wireless Local Area Networks (WLANs). The typical decentralized nature of WLANs (e.g., residential deployments) affects data collection and also leads to complex and highly non-stationary environments.

These challenges put into question the worthiness of introducing ML to networking systems. In particular, network operators and other stakeholders may have concerns regarding architectural (e.g., how to train and transfer ML models across a network) and operational aspects (e.g., how to provide trustworthy ML optimizations). While significant efforts have been put towards designing ML-based network architectures [1–4], only a small number of works have been devoted to study and address the side effects that ML can produce when applied to networks.

Thank you

Authors: Francesc Wilhelmi (UPF)

Vishnu Ram (Independent Research Consultant)

Presenter: Francesc Wilhelmi (UPF)
francisco.wilhelmi@upf.edu