O-RAN next Generation Research Group (nGRG)

Contributed Research Report

# Principles and Methodologies for AI/ML Testing in Next Generation Networks

**Report ID: RR-2024-04**

**Contributors:**

**VIAVI Solutions**

**Keysight Technologies**

**Dell Technologies**

**Release date: 2024.04.04**

## Authors

**Mohammad Alavirad; Dell Technologies (Editor-in-Chief)**

**Balaji Raghothaman; Daniel Garcia-Ulloa, Keysight Technologies**

**J. Gordon Beattie, Jr., Paul Harris; Viavi Solutions**

## Reviewers

**Aloizio Pereira da Silva; Virginia Tech**

**Ravi Sinha; Reliance Jio**

**Rajat Agarwal; Tech Mahindra**

**Bernard Guarino, O-RAN ALLIANCE**

## Disclaimer



## Copyright

## Executive summary

Testing next generation networks, which are new and unexplored, requires a thorough approach using advanced methods and principles. This is especially important in the changing world of radio access networks (RAN). The integration of Artificial Intelligence and Machine Learning (AI/ML) framework within the next generation network fabric will soon be a universal requirement, enabling cognitive decision-making processes and real-time adaptation to the complexities of modern communication scenarios. However, the presence of non-deterministic outcomes poses a significant challenge in this dynamic environment, necessitating the development of robust AI driven testing framework. This Framework, which incorporate advanced statistical models and simulation techniques, are crucial to ensure the reliability and performance of AI/ML algorithms in next generation networks.

This research report addresses how AI/ML systems can be seamlessly integrated into the next generation network architecture, ensuring compatibility and scalability across various network paradigms, including Open RAN, and how testing methodologies can be adapted to accommodate the intrinsic uncertainties associated with AI/ML algorithms in the network, while remaining adaptable for both traditional RAN and Open RAN configurations, as well as how the safety implications of AI integration in next generation networks can be addressed, ensuring that these technologies enhance user privacy, security, and trust. It concludes by discussing how predictive analytics and advanced testing techniques can be employed to proactively identify network bottlenecks, optimize resource allocation, and enhance overall network efficiency, regardless of the specific network infrastructure. By addressing these questions and embracing iterative development methodologies, this research not only contributes to the broader next generation testing landscape but also holds relevance in shaping the future of intelligent, adaptive, and efficient communication networks across diverse network architectures.

# Table of Contents

## List of abbreviations

| | |
|---|---|
| 3GPP | 3rd Generation Partnership Project |
| AI | Artificial Intelligence |
| API | Application Programming Interface |
| AR | Augmented Reality |
| BS | Base station |
| CNN | Convolutional Neural Networks |
| CSI | Channel State Information |
| DL | Downlink |
| DNN | Deep Neural Network |
| DUT | Device Under Test |
| HAPS | High-Altitude Platform Station |
| IDS/IPS | Intrusion Detection/Prevention Systems |
| IoT | Internet of Things |
| KPI | Key Performance Indictor |
| MIMO | Multiple Input Multiple Output |
| ML | Machine Learning |
| MNO | Mobile Network Operator |
| nGRG | Next Generation Research Group |
| NNEF | Neural Network Exchange Format |
| NFVI | Network Function Virtualization Interfaces |
| O-CU | Open Central Unit |
| O-DU | Open Distributed Unit |
| ONNX | Open Neural Network Exchange |
| O-RAN | Open RAN |
| O-RU | Open Radio Unit |
| QoE | Quality of Experience |
| QoS | Quality of Service |
| RAN | Radio Access Network |
| RIC | RAN Intelligence Controller |
| RIS | Reflective Intelligence Surfaces |
| RNN | Recurrent Neural Network |
| RSU | Roadside Unit |
| SDN | Software Defined Networking |
| SDO | Standards Development Organizations |
| SHAP | Shapley Additive exPlanations |
| SIEM | Security Information and Event Management |
| TE | Test Equipment |
| UE | User Equipment |
| UL | Uplik |
| URLLC | Ultra Reliable Low Latency Communications |
| VR | Virtual Reality |
| WI | Work Item |
| XAI | explainable AI |

## List of figures



## List of tables

# 1 Introduction

## 1.1 Overview of next generation networks and components

The upcoming next generation networks mark an essential moment in the telecommunications landscape. They bring forth sophisticated architectural designs to cater to a wide range of practical needs. Unlike mere iterations of previous networks, next generation network represents a ground-breaking advancement, integrating state-of-the-art technologies and innovative communication approaches. In this domain, AI/ML infuse every facet, augmenting intelligence, and responsiveness.

This deep integration of AI/ML creates a complex environment, presenting significant challenges to conventional testing methods. The testing of next generation involves complicated processes, where AI/ML algorithms are rigorously examined for their efficiency and accuracy. Researchers and engineers delve into comprehensive analyses, probing the AI-driven components to ensure they function seamlessly within the network architecture. These careful evaluations are crucial to guaranteeing the reliability and performance of next generation networks, paving the way for a future where telecommunications reach unprecedented levels of sophistication and efficiency.

## 1.2 How will next generation testing differ from 5G testing?

Native AI/ML fabric, High Performance Compute (HPC) platform, heavily distributed and virtualized infrastructure and cognitive Management are key technologies shaping the future of next generation testing in telecoms. As these technologies become integral to RAN functions and the air interface, operating in real-time, new testing and validation techniques will be necessary, requiring the creation of specific datasets for training and validation of AI/ML functionalities.

Currently, only the core network has a virtual architecture in 3GPP standards, but it is expected that the Radio Access Network (RAN) will also transition to a virtual format. Industry groups like the O-RAN ALLIANCE have developed guidelines for virtualizing 5G RAN, and next generation network is expected to inherently include virtual RAN functions as standard, just as with the 5G core network. A fully virtualized RAN will mandate the development of virtualized diagnostic and measurement tools to accompany these virtual functions.

## 1.3 Testing challenges and paradigm shifts

The arrival of next generation networks marks a significant shift in how we approach testing in telecommunications and radio access networks. The increased complexity brought about by these advanced networks necessitates a complete rethink of our testing strategies. In contrast to traditional systems, next generation networks driven by AI/ML framework introduce an element of unpredictability in their outcomes. While occasional inaccuracies may occur, the primary goal remains the improvement of the network's overall performance.

The dynamic nature of AI/ML systems in next generation network, which includes their training and potential retraining phases, introduces a level of complexity that was previously unseen in the testing landscape. Furthermore, ensuring fairness and efficiency in test results becomes critically important in an environment characterized by diverse network architectures.

Moreover, the next generation landscape promotes the adoption of AI/ML algorithms from different providers. This diversity necessitates a "black box" testing approach. This method ensures that these algorithms perform optimally without requiring an in-depth understanding of their internal workings. Therefore, comprehensive testing methodologies are indispensable to guarantee the seamless integration and optimal performance of AI/ML systems in next generation networks.

## 1.4 Considerations in designing next generation testbeds

When designing a next generation test setup, it is important to build a system that can test both the real, physical parts of the network and the virtual parts that can be changed and moved around. This means the testbed must work well with both the hardware that next generation runs on and the software that shifts according to needs. Since next generation networks will use AI/ML to manage things and solve problems, the test setup needs to be able to check how well this is working. AI/ML in the testbed can help predict issues before they happen, making sure the next generation network can be trusted and work smoothly. So, the testbed should be intelligent and flexible to keep up with the next generation network as it grows and changes.

## 1.5 Use cases from nGRG research streams RS01-RS04

The O-RAN ALLIANCE nGRG RS01 has been the forum where many new use cases for the next generation of communications have been discussed [1]. The use cases where the application of AI/ML have been postulated include [2]:

- optimization of energy efficiency in all aspects of the communication network
- intelligent techniques for emerging indoor use cases such as multimedia streaming, 3D Communication rendering, immersive virtual shopping etc.

The O-RAN ALLIANCE nGRG RS02, the Architecture-focused research stream, has discussed many aspects that involve AI/ML processes. These include closed-loop autonomous decision making in network operations, federated and distributed models for programmable intelligent RAN [3], and the use of ML-based distributed apps (dAPPs) for real-time RAN control. The O-RAN ALLIANCE nGRG RS03 is focused solely on AI, and hence will be the forum for most of the AI/ML work in nGRG. Some of the use cases discussed thus far have been cross-domain data analysis, intent-driven smart slicing, digital twins (DTs), and cross-domain QoE estimation [4]. The O-RAN ALLIANCE nGRG RS04 has discussed the use of AI in security, such as for adversarial techniques etc.

In this research report, we discuss understanding the unique methodology and paradigm for testing AI/ML applications, defining barriers in identifying testing items for different research streams. Also, we talk about determining the extent of testing required by assessing additional components or systems with native AI/ML, defining levels of testing to meet various operational requirements, analyzing the potential impact of reduced testing on overall security and performance. This research report describes challenges and differences in testing AI/ML systems compared to conventional systems, highlighting the paradigm shift in testing approaches and methodologies. It also emphasizes the importance of establishing a robust testing ecosystem that can accommodate algorithms from various equipment vendors, operators and incorporate industry standards.

# 2 Testing AI/ML in mobile telecommunications

The integration of AI/ML algorithms is becoming prevalent in the field of telecommunications. These algorithms will start to be used in a wide range of telecommunication applications, including predictive maintenance, network optimization, and personalized user experiences. Mobile telecommunication networks generate vast amounts of data, and AI/ML algorithms are increasingly being used to extract valuable insights and optimize network performance.

Integrating AI/ML algorithms into mobile telecommunications presents unique challenges in terms of testing and validation. Unlike conventional systems, AI/ML systems are characterized by their ability to learn from data and adapt their behavior over time. This dynamic nature poses challenges in ensuring reliability, and stability of these algorithms. As a result, traditional white-box testing methodologies become less applicable, necessitating the adoption of innovative techniques that focus on the system's outputs and behavior. Additionally, the complexity and interdependence of AI/ML systems require specialized testing strategies and tools to evaluate their performance and reduce potential risks.

## 2.1 Introduction to AI/ML testing in mobile telecommunications

With the increasing adoption of AI/ML algorithms by various service providers, it becomes evident the need to establish a robust test framework that can accommodate these algorithms and incorporate industry standards. This would ensure the reliability, performance, and interoperability of AI/ML systems in mobile telecommunications. By testing the algorithms from different providers, network operators can gain confidence in their functionality and make informed decisions about their deployment. In the following sections, we will delve into establishing a testing ecosystem that can evaluate AI/ML algorithms effectively.

## 2.2 Challenges in testing AI/ML systems

Testing AI/ML in the mobile telecommunications domain presents unique challenges. Specifically, one of the challenges is the dynamic nature of the network environment. Mobile networks are subject to constant changes, including fluctuations in network traffic, varying signal strengths, and evolving user behaviors. These dynamic characteristics make it challenging to validate and certify AI/ML algorithms, as their performance may vary under different network conditions.

Another challenge is the need for real-time decision-making in mobile telecommunications. AI/ML algorithms are often deployed to make critical decisions in real-time, which introduces additional complexities, as their performance must be evaluated not only in terms of accuracy but also in terms of speed, latency, and responsiveness.

The mobile telecommunications industry relies on a diverse ecosystem of network equipment vendors, service providers, and AI/ML algorithms providers.

Integrating these algorithms from multiple providers into a cohesive testing framework can be challenging, as it requires addressing compatibility issues, ensuring data privacy and security, and establishing standardized evaluation criteria.

Moreover, the complexity of AI/ML algorithms often results in them being treated as a “closed box”, where the focus is more on the input and output, rather than understanding the internal workings of the system. In contrast, “open-box testing” involves having access to the internal working of the model, including its architecture, parameters, and intermediate representations. This allows for a deeper understanding of the model’s decision-making process. Methods used in open-box testing include gradient-based attribution methods like Integrated Gradients [5] and DeepLIFT [6], which provide insights into the importance of input features. SHAP (Shapley Additive exPlanations) [7] is another method that assigns feature importance values to each input.

AI/ML algorithms can be categorized in a broad continuum in terms of interpretability vs explainability. Interpretable systems are those for which the humans have more minute knowledge of inner functioning, which explainable systems are those where the human can understand the broad mapping of outcomes to the data. Open-box testing provides better results in terms of being closer to interpretability in this continuum, which can be crucial for ensuring transparency and trust in ML models.

Closed-box testing focuses on testing the inputs and outputs of a model (In this case the ML model) without any knowledge of its internal workings. This approach is used when the model is complex or proprietary, and its internal details are not accessible. LIME (Local Interpretable Model-agnostic Explanations) [8] is a popular method for closed-box testing, as it approximates the model’s behavior locally by training an interpretable model on perturbed samples. Other methods for closed-box testing include SHAP, surrogate models, and adversarial attacks. In the context of AI in mobile telecommunications, closed-box testing might be more suitable due to the complexity and proprietary nature of the ML models used. A combination of closed- and open-box testing, known as translucent-box testing, is a method that involves having a partial knowledge of the internal workings of the system, striking a balance between transparency and practicality.

### 2.3 Ensuring the reliability of AI/ML systems

To ensure the reliability of AI/ML systems, the testing strategies need to be comprehensive and robust, taking into account the unique characteristics of the telecommunications systems. One such strategy is conducting simple tests for behavioral conformance, which involves evaluating the AI/ML system’s response to different inputs and scenarios. For example, in the context of network optimization algorithms, a simple test for behavior could involve simulating different network conditions, such as high traffic or network congestion, and observing how the algorithm adjusts its parameters or makes decisions to optimize the network dimensioning and improve performance.

Another crucial strategy is the validation of models' outputs against ground truth data. Ground truth data, which represents actual real-world scenarios and outcomes, is used to verify the accuracy of AI/ML predictions. Through techniques such as cross-validation, where the model is trained on a subset of the data and tested on the remaining unseen data, developers can assess the model's performance against real-world situations. Anomalies and discrepancies between predicted outcomes and ground truth data can indicate potential issues in the model's training or the quality of input data.

Code coverage refers to the extent to which the code of the system has been executed during testing. This involves testing the AI/ML system under all possible conditions and scenarios to ensure that all parts of the code have been executed. This is particularly important for AI/ML systems with complex code structures. A comprehensive code coverage helps identify hidden bugs or issues that might not surface in regular testing scenarios. It can be achieved through techniques like unit testing, integration testing, and system testing, where different parts of the code are evaluated against expected outcomes.

Testing for robustness and stability involves evaluating the system under various conditions and stress levels, such as high loads, extreme inputs, or other such adverse conditions, to ensure consistent and reliable performance. In the next section we delve into the importance of establishing a robust testing ecosystem that can help developers identify and address potential vulnerabilities or performance issues, ensuring the AI/ML system can operate effectively in real-world scenarios.

## 2.4 Establishing a robust testing ecosystem

Different providers may offer unique approaches and solutions to AI/ML challenges, and a robust testing ecosystem should be capable of accommodating these diverse algorithms. This enhances the flexibility of the ecosystem and promotes innovation and continuous improvement.

Adapting to industry standards is another important aspect to establishing a robust testing ecosystem, since they provide a common framework and guidelines for developing and testing AI/ML systems. By adhering to these standards, developers can ensure the compatibility, reliability, and interoperability of their algorithms. Industry standards should reflect the best practices in the field, which also helps developers avoid common pitfalls and provide better testing solutions.

To ensure this ecosystem is comprehensive, we focus on a methodology that covers all stages of the machine learning lifecycle, including problem analysis, validating the training data and feature engineering, testing the model training, model evaluation, model deployment and inferencing, and model monitoring. Each stage requires specific tests to ensure the reliability, accuracy, and performance of the system. For example, tests related to validating the training data and feature engineering ensure the quality and representativeness of the data by examining the data distribution, checking for bias, identifying missing or duplicate values, and

detecting outliers or anomalies. The quality of the training data has a direct impact on the quality of the trained model.

Another stage to emphasize is testing the model training, whose objective is to ensure that the model can appropriately generalize and perform well on unseen data. To achieve this, various techniques are used, such as regularization methods and hyperparameter optimization [9]. Hyperparameter optimization involves tuning the model's hyperparameters such as learning rate, batch size, and number of layers to find the optimal configuration that maximizes the performance. Regularization methods, such as L1 or L2 regularization, are used to prevent overfitting and obtain the right balance between bias and variance.

The main role of the monitoring stage is to continuously monitor the models' performance and detect any change that may occur over time. One of the main aspects of model monitoring is testing for data drift and concept drift [10], [11]. Data drift refers to situations where the statistical properties of the incoming data change over time, leading to degradation in the model's performance. Concept drift refers to the scenario where the underlying relationship between the input features and the target variable change over time. Both data and concept drift can significantly impact the effectiveness of ML models in mobile telecommunications. Therefore, the model monitoring stage involves regularly testing for these drifts and implementing appropriate strategies to adapt the model or even retrain if necessary.

In the next section we shift our focus to testing AI/ML particularly in the context of mobile telecommunications. We explore the various techniques and methodologies that can be employed to ensure the reliability, performance, and adaptability of these systems in the telecommunications environment.

## 2.5 Testing strategy for AI/ML algorithms in mobile telecommunications

The immediate strategy is to provide tools for quantizing the performance of a deep learning algorithm in terms of accuracy, when expected values are known, or in terms of optimizing a key performance indicator (KPI). The testing applies the methods described above. However, when it comes to explaining what the deep learning algorithm is doing, or why it decides to perform in a particular way, then surveys, like [12] conclude that as the technical performance increases the ability to explain the model actions, the explainability, tend to go down. A Deep Neural Network (DNN) is not based on a mathematical model, which means that it is inherently less transparent. It is therefore required to design tests for evaluating performance in terms of various KPIs, as well as tests to quantize explainability.

Another practical strategy is to optimize the complexity vs. performance tradeoff of an ML implementation. Studies have found that NNs generally include too many parameters, and that some are with lesser importance, which allows for pruning the network, while maintaining very similar performance. There are several ways that this may be done.

1. Removing weights, i.e., connections between neurons, to lower the number of parameters.

2. Re-designing the network, like reducing the dimension by replaying fully connected layer by convolutional layer.

3. Quantizing weight values, like using inter values with lower bit-width limit, hence more granular and less accurate values.

One objective is to shrink the DNN to a manageable size that requires less resources, like storage for parameters and time for evaluating the network. This execution time is critical, especially in the in physical layer where the overall operations must take less than a fraction of a millisecond. The performance and explainability tests will show that as a simplification of the DNN, or of any model, it generally improves the explainability. The performance testing may then be used to quantize the loss in some KPI versus the gain in complexity, to identify an acceptable compromise.

### 2.6 AI/ML applications in physical layer and RAN

Deep learning can be applied to several places on the physical layer. Equalization of nonlinear distortion in signal detection may be achieved with received signal as input and known transmitted data, pilot etc. as target. This can outperform implementations based on MMSE. Channel correlation over time implies that a bidirectional recurrent neural network (BRNN) is suitable for obtaining a model where output layers can make use of “inner state” information from both the past and the future. In this case the Channel State Information (CSI) is not required, and the solution can outperform Viterbi detection. Several studies in mmWave massive MIMO and end-to-end channel estimation solutions have found good performance of DNNs, generally outperforming existing mathematical and model-based solutions. The following presents examples of cases where deep learning can be used at the physical layer:

**Source and channel coding**. CSI feedback that transfers information about the state of a wireless channel from the receiver to the transmitter that adapts transmission parameters for best performance with the current channel condition. Channel coding add bits achieving redundancy that improves the detection and correction of the original data. An autoencoder is an unsupervised deep learning method applying a coder for mapping input to a lower or higher dimensional representation and a decoder for estimating the original input. An autoencoder may replace the above encoding and adds redundancy when the intermediate dimensionality is increased. This is transmitted and the decoder applied on reception. An autoencoder may also replace the modulation and detection blocks. And a single autoencoder may substitute both encoding and modulation. A recurrent neural network (RNN) autoencoder captures temporal features and performs better [ 13 ]. A bidirectional RNN performs even better, as it captures temporal information from both past and future states [14].

**Channel estimation**. In channel estimation there is a tradeoff between reliability and spectral efficiency, as more resources for channel estimation, like for pilots, increases the accuracy of detection but tend to lower the throughput of the system. Innovations in channel estimation based on ML are promising as the

complexity increases with more antennas and specialized configurations. One example is Reflective Intelligence Surfaces (RIS) that are passive elements that can change the phases of impinging waves. This adds a new dimension to the channel estimation problem that can be implicitly captured with deep learning -based methods that can achieve significant gains.

**MIMO**. Multiple input multiple output (MIMO) and specifically massive MIMO was introduced at large scale in 5G and has a potential for large gains in diversity, array and multiplexing, when applying ML methods. An example application is pseudo-localization utilizing unsupervised learning. AI/ML is also used to address MIMO challenges, like MIMO detection and cell-free MIMO. MIMO-GAN: Generative MIMO Channel Modeling and AI-Empowered Hybrid MIMO Beamforming are also some of the domain specific GenAI based approaches very relevant to MIMO.

**Beam selection**. Beam Selection is challenging with MIMO that require selection of multiple optimal transmission vectors for MIMO. Deep Learning algorithms using Convolutional Neural Networks (CNNs) can predict the optimum number of beams. ML methods are particularly promising in dynamic environments, like with varying channel conditions, mobility, and interference. The objective is to adapt the beamforming vectors to the changing environment in order to improve the signal quality, increase coverage, and mitigate interference.

On the data link layer, the following are the most prominent examples:

**Scheduling and multiple access**. Scheduling is a critical task and will be more challenging with the explosion in the number of devices. Methods for this include time division multiple access (TDMA), orthogonal frequency division multiple access (OFDMA), and code division multiple access (CDMA). Non-orthogonal multiple access (NOMA) allows for mutual interference between devices, while using successive interference cancellation (SIC) to separate the signals at reception. Although NOMA was a prospective technique considered for 5G, it was ultimately omitted from the finalized specifications. It is indeed anticipated to play a significant role in next generation networks. Next generation multiple access (NGMA) addresses the evolution of NOMA for next generation network, replacing the relatively simple NOMA techniques used in 5G by ML-based techniques, including DNNs, federated learning, reinforcement learning, and Multi-Deep Q-Network (MDQN).

**Handover**. The exponential increase in density of users in urban scenarios implies that next generation networks must have many more cells and be generally heterogenous, like with both micro and picocells. This causes a much higher number of handovers, due to mobility and variations in the local environment, combined with many small and overlapping cell coverage areas.AI/ML is essential for capturing the complex environment and minimize the number of handovers, more conservatively staying with current cell longer, and apply handoff predictions in services. Some 5G/next generation services, like ultra-reliable low-latency communications (URLLC), are not feasible with excess handovers. Also improved estimations of blockage may be used to mitigated by proactive handovers.

On the network layer we have the following examples:

**Network traffic monitoring and analysis**. Understanding and controlling network traffic will be of large importance in next generation network in order to monitor networks, manage resources efficiently and detect faults. These tasks fall under the label network traffic monitoring and analysis (NTMA), which will be of increasing importance to future networks transferring large amounts of data in heterogeneous networks. Here, traffic prediction is of particular interest to the deep learning community, where predictors based on classical time-series analysis are challenged by the recurrent neural networks, which can potentially achieve higher accuracy in model complex and non-stationary traffic patterns.

**Route planning**. Finding the optimal path for a packet is an essential task of the network layer. Optimizing for the best path in a network should consider not only the length (latency) of the route but also account for power consumption, network congestion, and specialized network typologies. Finding the shortest path (by some measure) in a network is a well-studied problem, but as the scale and complexity of modern networks increase, the classical methods based on graphs fall short. Solutions based on Q-learning, known as Q-routing, which seeks to maximize a reward by taking an action given the state of the system, is a popular approach. A new promising approach to ML-based routing uses graph neural networks, which are neural networks designed specifically to capture graph structures and capable of efficiently handling large graph structures.

**Prediction of QoS and QoE**. The Quality of Service (QoS) of a connection can be measured at the network layer based on key performance indicators (KPIs), such as throughput, error rate, jitter, etc. Quality of Experience (QoE), on the other hand, is a subjective measure based on the user experience. Both are important measures; however, the current 3GPP standard is mostly focused on QoS. An interesting area is how to map the correlation between QoS, which can be measured directly, and QoE, which is much more difficult and resource intensive to measure. These models are known as QoS/QoE correlation models and are often based on ML methods. A related topic is a prediction of QoS, which is important for use cases such as Internet of Things (IoT) and scenarios with mobile users.

**RAN intelligence controller (RIC):** Central components specified by O-RAN are the Near-Real-Time and non-Real-Time RAN Intelligent Controllers (Near-RT RIC and non-RT RIC). The Near-RT RIC is a logical function that enables querying, control and optimization of RAN elements and resources with data collection and actions over the E2 interface that operates at 10ms to 1s intervals. This interface communicates directly with base station entities. The non-RT RIC communicates with the Near-RT RIC via the A1 interface that operates at higher than 1s intervals. It may provide the Near-RT RIC infrequent information, like network information, subscriber data, AI based recommendations, and generally policy-based guidance. The non-RT RIC can monitor detailed per device performance and QoS/QoE statistics.

The Near-RT RIC provides interfaces for applications, xApps, that can connect using Service Models, like E2SM-KPM for Key Performance Metrics, that specifies data elements and actions that are provided for the implementation of various microservices. As an example, an AI-based traffic steering xApp based on the E2SM-

KPM was demonstrated on O-RAN India plugfest in 2021. The non-RT RIC provides similar interfaces for applications, rApps, that as the non-RT RIC operates within the Service Management and Orchestration (SMO) framework.

Procedures are required for testing and monitoring of services provided by xApps and rApps that may be developed by 3rd party software providers.

## 2.7 Characterization

ML is used for multiple purposes. We list some examples:

**Estimation of hidden state**. The ML estimates information that is not directly available but can be deduced from the information that is available. The results may be used for further analysis or simply provided to an operator or end user.

**Predicting future state**. The ML extrapolates and predicts future state, like for early alarms and input to control algorithms.

**Decision making**. The ML outputs values that indicate a decision, like turning features on and off, modifying set-points and other settings, selecting beam, routing packet and prioritizing traffic flows and users. The list is long, as most applications of ML are as part of a control loop where the ML is taught how to extract information and make a sensible decision, whether this is choosing a modulation scheme, or determining the amount and location of resources to be used for a transmission.

In any case, it is required to have a testing framework that allows for exercising a solution under specific conditions while analyzing the performance, including statistical tools for characterizing the ML outputs as well as the process that it is impacting. Simple descriptive statistics may reveal an excess variance that is an indicator for inaccuracy in estimation or instability in a control loop. Additionally, characterization of the dynamics is important, as metrics like gain, rise time, settling time and steady state error are relevant also when applying a non-linear ML-based controller. This includes quantizing the explainability as well as performance of the solution.

It is expected that solutions will be delivered by third-party developers as xApps and rApps calls for standardized testing procedures and metrics. It is even more challenging when multiple such services are used, as the behaviour of one may couple to the behaviour of another.

**Data quality and integrity**: In addition to characterizing ML outputs and ensuring standardized testing procedures, it is of very importance to address the quality and integrity of the data used in ML applications. High-quality, unbiased, and representative data is the basis upon which ML models are built. Data anomalies, inaccuracies, or shifts in data distributions can significantly affect the performance and reliability of ML algorithms [15]. Therefore, while standardized testing procedures are essential for evaluating ML models, ensuring data quality is a prerequisite that underpins these tests. Monitoring data quality in real-time and implementing mechanisms for data validation and cleansing should be an integral part of any

comprehensive testing framework for ML applications. By maintaining high data quality, we can ensure that the standardized testing procedures yield meaningful and accurate assessments of ML model performance.

### 2.8 Digital twins (DT)

A DT is a virtual representation of objects, like a radio, a base station, a RAN. It must represent the processes of relevance by simulation, possibly by use of hardware modules, sufficiently accurate for this to be considered behaving same as the real system. In a communications system this includes application programming interfaces (APIs), traffic models, environment models, possibly artificial load on various systems. It must behave same as the real thing and provide the same interfaces (APIs).

The benefit of using a DT is that it can be used to generate realistic data that is distributed same as by the real system, which can be used for training the ML features of a (sub-)system that is under test. This may be new features considered for deployment in the real system, which means that decisions on how to extend and improve the real system can be made based on evaluations based on the DT. Supervised training is used for most current ML solutions, and data is collected under some assumptions. A DT allows for the ML training to explore ways unlikely to be envisioned by persons producing training data for supervised learning. In terms of testing a DT allows for more efficient testing that may include cases that would be destructive when performed on the real system. An AI-based testing system may be used for wider exploration of the way that the tested system can be operated.

Additionally, DTs can significantly enhance cybersecurity measures. By creating a virtual replica of the system, cybersecurity experts can simulate various cyber-attacks, vulnerabilities, and breach attempts. This simulated environment allows them to proactively identify weak points, anticipate potential threats, and develop robust security solutions before they are deployed in the real system [16].

Within the O-RAN architecture, an AI/ML Server is a component that provides orchestration and service functions for the AI/ML closed-loop in the O-RAN ecosystem. The AI/ML Server, also called, AI/ML Framework includes functions such as data collection, data validation, data training, and data model packing. O-RAN Alliance specified two ways to locate the AIML framework in the O-RAN ecosystem: internal or external to the SMO.

Today, a single and reliable AI/ML framework that can be used across the O-RAN-based architectures does not exist. Alternative existing frameworks such as ClearML, Acumus, OpenGYM, and AIMLFW are some of the options. Some of them despite claiming O-RAN-based or being providers of the service functions previously mentioned. They lack several features and capabilities. The lack of flexibility, security, and interoperability across O-RAN components and interfaces demands appropriate testing methodology to evaluate such frameworks before testing the intelligence by itself. In this direction, testing the conformance, performance, and interoperability of the AIML framework is key. For example, testing the accuracy and precision of the AIML framework on data training mechanism, testing how secure and trustable is the

data collection mechanism and testing the capability of the AIML framework to keep the RICs time constraints, are some examples that need to be addressed when approaching AIML testing in the O-RAN ecosystem. Once the AIML by itself has been tested and validated, the next step is to test the AIML solutions and microservices that have intelligence embedded in them. In this case, DTs can play a fundamental role.

## 2.9 AI/ML testing for the air interface

As part of Release 18, 3GPP studied the application of AI to the air interface for the first time considering three initial use cases: beam management, positioning enhancement and CSI feedback enhancement [17]. Beam management considers how AI could be used to enhance selection of the optimal downlink (DL) beam in scenarios where a grid of beams is used. The positioning accuracy enhancement use case considers how AI could improve the accuracy of user equipment (UE) location measurements in the network. Finally, CSI feedback enhancement considers how both compression and prediction could be enhanced by AI to reduce the CSI feedback overhead for the network, which could be particularly valuable for a technology like massive MIMO where there is a high channel count.

The above use cases can be categorized as either one-sided or two-sided model deployments. Beam management and positioning enhancement come under the one-sided deployment category, where an AI model will typically be deployed in the UE and infer outputs independently of the network. CSI feedback enhancement, and particularly CSI compression, comes under the two-sided category due to the use of an autoencoder. For example, with uplink (UL) CSI compression, the UE would need to employ the encoder part of the model and the base station (BS) would need to have the associated decoder.

From a test perspective, the one-sided cases are arguably simpler since everything is contained on either the UE or BS side, and so long as the appropriate stimuli, metrics and KPIs can be agreed, this fits a more conventional test equipment (TE) setup. For CSI compression, a new dimension of complexity is introduced for test since the test equipment vendor needs to incorporate a decoder within their systems for testing the encoder of a device-under-test (DUT), or vice versa. This gives rise to the question of the source of the test decoder or encoder. If a UE vendor implementing an encoder provides their own decoder for testing, this could arguably not be representative of the real-world conditions since they do not know which decoder will be deployed by different BS vendors in the field. Alternatively, fully specifying a test encoder or decoder in the standard could alleviate this issue, but further technical evaluation is needed to understand the feasibility and limitations of this approach. 3GPP started its evaluation of these options and others within the Release 18 study.

For Release 19, a new work item (WI) for AI/ML at the air interface has been approved which will carry out normative work for the beam management and positioning enhancement use cases [18]. This WI will also conduct further study of the CSI feedback enhancement use case and attempt to converge on a feasible approach for model development and test.

## 2.10 A comprehensive methodology for testing AI: A use-case example

As described in previous sections, testing ML models is needed for a wide variety of reasons. Reliable and stable models instill confidence in their users that it will produce accurate results and behave predictably, leading also to a faster adoption and usage. These models are often deployed in real-world scenarios and several processes are automated with machine learning. In this section we show a comprehensive methodology for testing AI, which takes into consideration the entire lifecycle of the ML model. To illustrate this methodology, we also show an example use-case centered around a 5G Beam Selection model.

### 2.10.1 Machine learning model testing

Here we show an overview of the stages in a ML lifecycle and how to test ML models. The regular AI/ML Model development consists of the following steps:

1) Problem Analysis
2) Data Analysis and feature engineering
3) Model Training
4) Model Evaluation
5) Model deployment and Inferencing
6) Model monitoring.

These are well-defined stages, and each of them should be tested. This process creates a standardized way of testing machine learning models, which would lead to reliable results. Each stage should have a set of pre-defined tests to perform, which may vary depending on the model under test. At all stages, the tests would provide feedback on the current development of the model, and there might be the need to go back to a previous stage to fix issues from previous stages (feedback and iterate arrow). Figure 1 shows an overview of the testing methodology.

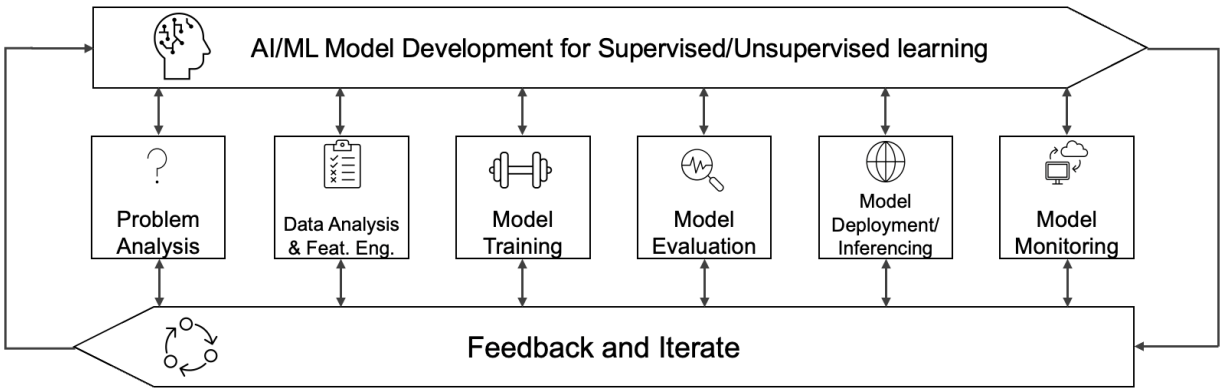


**Figure 1 Overview of the testing methodology. During the ML model development, the different stages are used for testing the model. At any stage**

**there is feedback and there might be a need to iterate and go back to a previous stage.**

### 2.10.2 Use-case example

The use case we will use to exemplify this methodology is a 5G Beam Selection problem. Here, a mmW cellular network is considered to be deployed in an urban street canyon environment. There are multiple RUs mounted on tops of buildings, communicating with UEs in moving vehicles on the street. The problem is to select the optimal RU and beam to transmit to any particular UE, while minimizing interference to other UEs. An ML algorithm is used for this beam selection, with channel strength data being the primary input.

It is a supervised learning model that consists of a multi-class classification model. It uses a CNN, and it is open source. The data for this model is publicly available. It is data published in IEEE in the Information Theory and Applications Workshop (ITA) [19]. The model under test is also publicly available in Github, and it was developed by LASSE (see [20])

### 2.10.3 Problem statement

Authors in [19] describe a methodology that combines a traffic simulator with a ray-tracing simulator to generate channel realizations in 5G scenarios. The traffic simulator consists of buildings of different heights (going from short blue ones to tall, red ones) made of ITU concrete and vehicles made of metal. The material choice is significant for the simulation as it impacts the propagation and reflection of the electromagnetic waves. There is a Roadside Unit (RSU) that transmits a beam to a receiver (a vehicle) and the problem consists in findings the best pair of transmitter and receiver antennas that optimizes the beam strength. Figure 2 (Left) shows the 3D environment used for generating the dataset. The colors indicate the heights of the buildings, with blue being the shortest and red being the tallest buildings. Ray tracing simulates waves from the transmitter to the receiver, and all the information regarding various rays, traffic simulation, vehicle dimension, angles of arrival, etc. are all stored for later generation of the dataset.

These traffic and ray-tracing simulation is simplified to a matrix to relate the beam-selection problem only on the positions and sizes of the vehicles. The RSU receives, through an error-free channel, the position, and a unique index of all vehicles in each scene. The RSU, based on this vehicle index, understand each vehicle’s dimensions, and may incorporate this information for the ML algorithm. Figure 2 (middle) shows the ray tracing simulation from the transmitter to the receivers. For visualization purposes, the camera in the middle figure was rotated with respect to the one in the left figure.

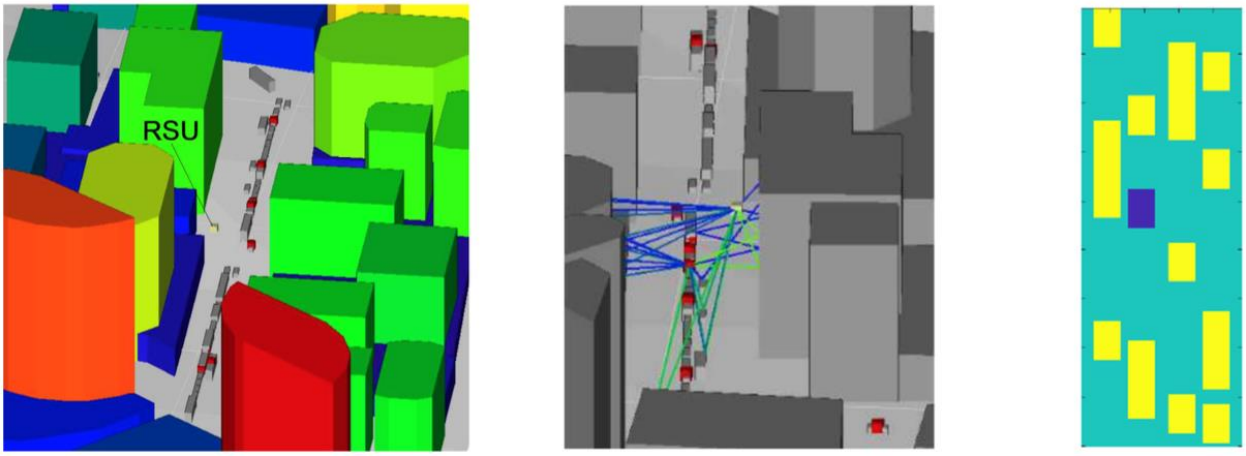


**Figure 2: Left: Urban 3D ray-tracing sim with height-coded buildings from 0 (blue) to 101 meters (red). Middle: Rays obtained during traffic jam with 25 significant rays reaching a vehicle. Right: Image corresponding to the matrix depicting 13 vehicles in four lanes where +1 values represent the receiver vehicle (blue) and -1 values represent the position of other vehicles (yellow). The rest of the pixels (turquoise) are 0 and represent the road.**

For the beam-selection, a CNN is proposed. CNN is used to identify features of a multi-dimensional input. The input is filtered by a kernel of same dimensions as the input. Each application outputs the convolution between the kernel and the input samples around points at intervals in the dimensions. This implies a down sampling, thus shrinking dimensionality. Many samples for accuracy, but correlation allows for sparse application of the filter.

Multiple layers of CNN are used for first extracting or enhancing small features that are in subsequent layer combined into larger features that are based on a hierarchy of sub-features. The outputs of CNN layers, called feature maps, contain samples that are very dissimilar to the original data. They process the data, while enhancing sub-structures, that allows for efficient mapping into targeted features by the final classification layer.

In this case, the input to the CNN is the spatial data of the vehicles, such as their positions and sizes, as well as the RSU's location. The final layer of the CNN is the targeted features are the 61 pairs of beams that are predicted to be the best for a given set of vehicle positions and RSU locations. The transmitters and receivers have 4x4 uniform planar antenna arrays, implying 16 antenna elements each, and thus a theoretical maximum of 256 pairs. The classification identifies the 61 optimum pairs are identified. Figure 2 (right) shows the input to the CNN model created by reducing the 3D environment.

### 2.10.4 Data analysis: bias in the target value

Figure 3 shows the proportions of given pairs beams in the dataset. For each pair of transmitter and receiver beams on the x-axis, the bars show the proportion of

this beam pair in the dataset. The dataset is divided into a set for learning and one for subsequent testing. The proportions in each set are shown by the blue and red bars, respectively.

The dataset is imbalanced, as the proportions of the 61 identified pairs vary. An extreme is that more than half the pairs are (2,14), which biases the learning towards selecting this pair irrespective of the actual performance. No learning, always selecting this pair, is correct more than half the time. Imbalance in samples may lead to a bad model, as some have little weight during learning.

This can be mitigated by under-sampling the many and design collection scenarios to favoring the few. Ideally the proportions of outcomes should be even. The proportions observed in the training and testing dataset must be close to equal to ensure consistency. Even proportions show a good random split of original dataset.

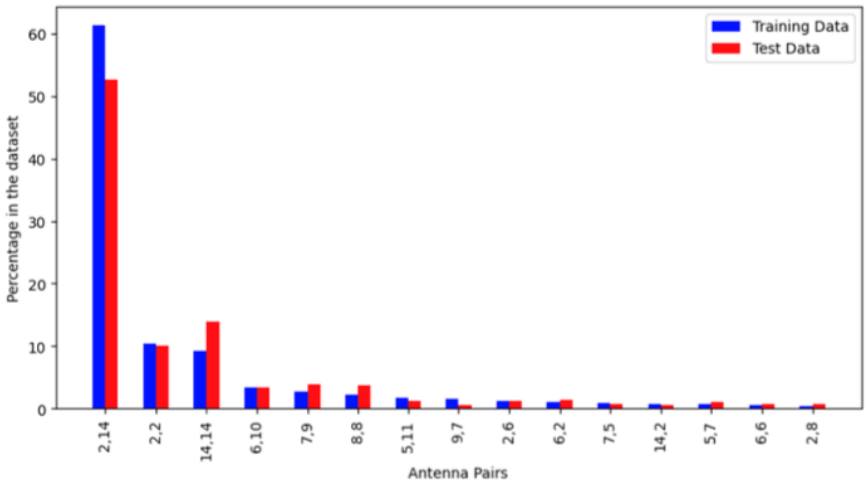


**Figure 3: Distribution of classes in the dataset indicating some classes are over-represented compared to others.**

### 2.10.5 Data analysis: spatial bias

Spatial bias occurs when the sampling space is represented unevenly. Here, space refers to values that may be physical distances or state values not included directly in the trained model. To ensure that the trained model is performing well for any combination of states, as are defined as feasible, then the data collection must be done for all such combinations.

The distribution of states within the collected dataset must be close to even to avoid bias. Combinations of states that are poorly represented have little weight in both learning and testing, leading to a model that may not perform well in these areas. Figure 4 shows a heat map of the position of receivers during data collection. The model, intended to perform in the whole square, is trained using data collected only along the diagonal. Validation using the testing dataset with same distribution does not verify performance outside the diagonal, in the top left and bottom right corners.

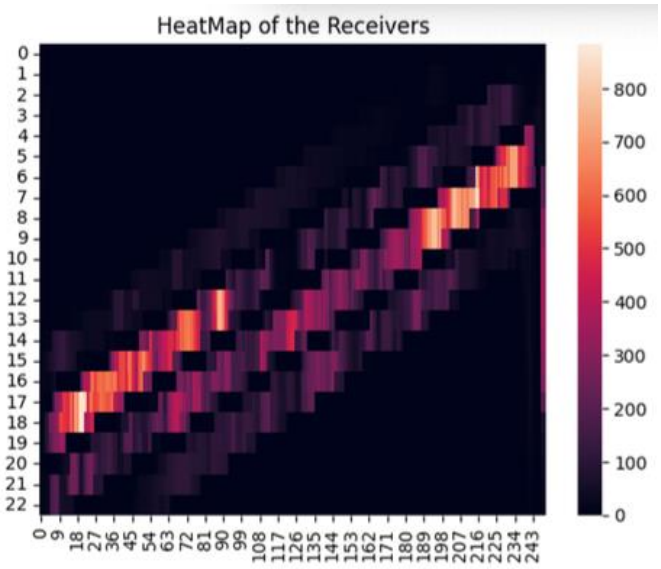


**Figure 4: Heatmap of receivers in the model features. Most receivers are found in the diagonal, which might indicate spatial bias.**

### 2.10.6 Model training: performance

Learning is repeated in what is called epochs, repeatedly updating model weights for a better fit when processing samples from the training dataset and evaluating the performance when processing samples from the testing dataset used for validation.

We measured the model's performance in terms of accuracy and evaluation of the loss function. Other metrics could be used too, like precision, recall, and F1-score. Loss is a measure of difference between the predicted values and the correct values, and as we train the model, we expect these values to decrease. Accuracy is a measure of the number of successes. For classification, success means selecting the correct class. If something is misclassified, accuracy does not take into consideration the differences between false positives and false negatives and treats them equally as failures. Figure 5 shows the evolution of accuracy and loss over a few epochs of training.

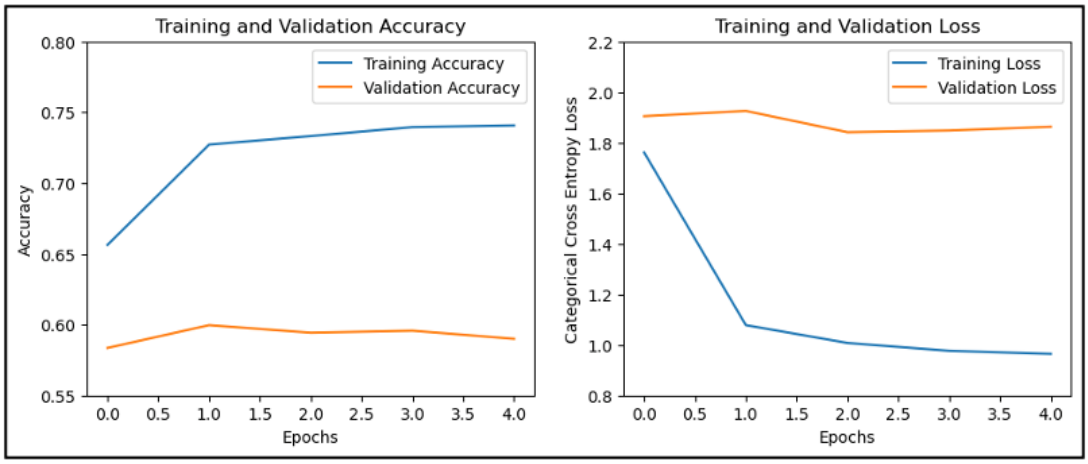


**Figure 5: Accuracy and Loss of the model under test. Training accuracy increases, but validation set remains the same, indicating overfitting. Similarly, the loss function decreases with the training set, but remains the same with the validation set, which indicates the model does not generalize appropriately.**

The right plot shows that the training loss is going down, but it does not improve the validation loss. This means that the training set is fitted very well, while the validation loss does not improve. The model has been overfitted, and it does not generalize to fit unknown samples. The left figure shows that the training accuracy is going up, but it does not improve the validation accuracy. This also shows overfitting.

The overfitting may be mitigated by simplifying the model or- introducing regularization by using a modified training error measure penalizing high weights.

### 2.10.7 Model evaluation: robustness

We also checked for robustness of the model using an adversarial technique, which tries to fool the model by generating carefully crafted noise to the input to make the model incorrectly categorize the input. Figure 6 shows the results of this adversarial attack. In this case, we see that the model's accuracy drops considerably. From the figure, we see that class 11 was misclassified as class 19 almost 400 times. Class 19 was misclassified as class 20 around 375 times, and so on. We conclude that the model is not robust towards adversarial attacks.

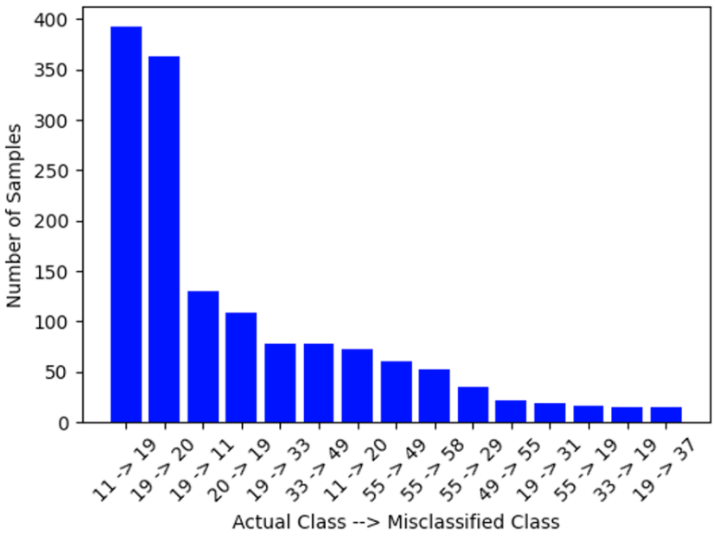


**Figure 6: Results of adversarial attack. The most likely classes to be misclassified after the attack were 11 and 19, which were misclassified as 19 and 20, respectively.**

One way to counteract this lack of robustness is to include adversarial training, which consists in adding these carefully crafted new inputs into the training of the model. Another way is through defensive distillation, which is a technique where there is a 'teacher model' trained with the original dataset. The teacher model outputs probabilities for each class for a given input, instead of just the class itself. Then a student model is trained on the same dataset, but this time using the output probabilities from the teacher model as labels, instead of the original hard labels. These new labels contain information about the likelihood of each class according to the teacher model, providing the student model with more information about the data's structure, and making the student model less sensitive to small perturbations in its input.

### 2.10.8 Model evaluation: explainability

Shap Values help us understand why a specific data point gets classified as a particular class. Figure 7 shows a particular case. In our example, we're focusing on a particular data point that seems to belong to class 19. Shap Values reveal how nearby pixels either support or oppose this classification. While there are 59 other Shap Value graphs for other possible classes, we won't delve into all of them here. What's interesting is that the pixels influencing the decision, whether positively or negatively, are mostly found along the diagonal line. This observation coincides with what we've learned from the heatmap regarding the positions of receivers. Notably, when a pixel has a positive impact on the most likely class, it has a negative impact on the least likely class, creating a color contrast between red (positive impact) and blue (negative impact).

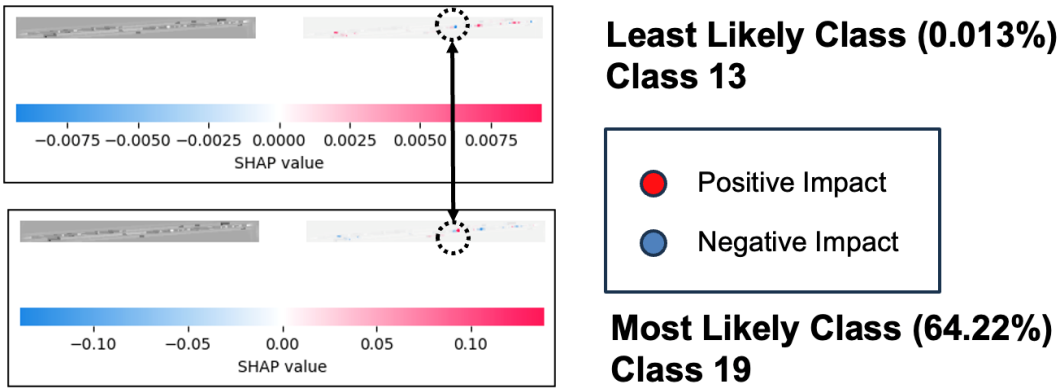


**Figure 7: SHAP values for the least and most likely classes of a particular sample. Red pixels indicate positive contributions and blue pixels are negative contributions. Values are most in the diagonals which is consistent with our previous analysis of spatial bias.**

### 2.10.9 Model deployment: robustness to data drift

After deploying the model, it is important to evaluate its performance for data drift. Data drift occurs when the input data distribution changes. To simulate this, we horizontally flip images, creating new data points that shift the data distribution. Unfortunately, we lack labels for these flipped images, making direct evaluation challenging. However, we can measure model uncertainty using entropy, which quantifies prediction uncertainty. A well-performing model should have consistent entropy between original and flipped images. Yet, when we plot entropy histograms for both datasets, they are not the same. The distribution from the original set and the flipped dataset are different. This suggests that the model's training was incomplete, specifically in recognizing receivers in certain image areas, reinforcing findings from prior data analysis.

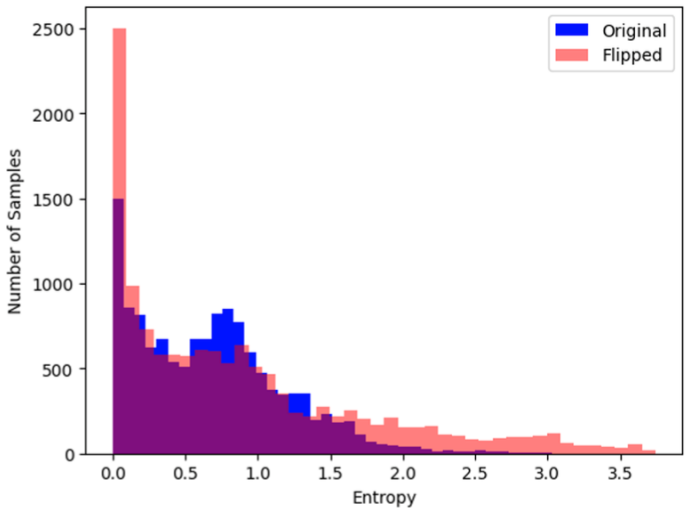


**Figure 8: Histogram of Prediction Entropy of the original dataset and the 'flipped' dataset, which includes images in the opposite diagonal. Since we do**

**not have the labels for these images, we can only estimate how certain the model is about these predictions.**

### 2.10.10 Conclusion

Controlled sampling is required to in the development of ML models to mitigate bias. It is important to ensure that the sample space, which represents the environment from which the data is drawn, and the state space, which represents all possible states or configurations that the system can achieve, are properly represented.

Thorough testing is required not only before deployment to validate the model's performance, but also after deployment to continuously monitor and mitigate any type of drift that might emerge.

Testing must be an integral part of every stage in the ML lifecycle, from initial design to final output. Moreover, these tests need to be adapted on a case-specific basis, acknowledging that different applications may require different testing protocols.

# 3 AI/ML Integration in next generation networks: challenges and solutions

This chapter provides a comprehensive overview of the challenges and solutions related to integrating AI/ML in next generation networks:

## 3.1 Data-driven and intent-driven ML in next generation networks

Integrating ML algorithms, both data-driven and intent-driven, into next generation testing presents unique challenges, but also offers promising solutions. One of the primary challenges lies in the complexity and dynamic nature of next generation networks. Data-driven ML algorithms require vast amounts of data for training, which can be difficult to collect and process in a constantly evolving next generation environment. This is compounded by the need for these algorithms to adapt to new data patterns and network configurations in real-time, making continuous learning and adaptation a necessity.

Intent-driven ML algorithms face their own set of challenges, primarily in accurately interpreting and implementing user intentions in a network environment as complex as next generation networks. Ensuring these algorithms understand complex user requirements and translate them effectively into network actions requires sophisticated natural language processing and decision-making capabilities.

Another significant challenge is the validation and reliability testing of these ML models within next generation networks. Ensuring that these algorithms perform accurately and consistently under various network conditions is critical, but doing so requires advanced testing methodologies that can simulate a wide range of scenarios and conditions.

In terms of solutions, one approach is the development of more sophisticated data processing techniques. Leveraging edge computing can help manage the data load by processing data closer to its source, thereby reducing latency and bandwidth usage. For continuous adaptation, employing online learning and federated learning approaches can enable ML algorithms to learn and evolve in real-time based on new data inputs [21], [22].

To address the challenges with intent-driven ML, the use of advanced semantic analysis and contextual understanding techniques can enhance the ability of algorithms to interpret user intentions more accurately. This can be complemented by user feedback mechanisms, allowing the system to refine its understanding over time [23].

For testing and validation, simulation-based testing environments can be highly effective. These environments can replicate a wide range of network conditions and user scenarios, allowing for comprehensive testing of ML algorithms. Additionally, implementing AI-driven testing tools can automate the testing process, making it more efficient and thorough.

Finally, the integration of explainable AI (XAI) principles in ML models can also be beneficial. XAI can provide insights into how ML decisions are made, which is

crucial for troubleshooting and refining algorithms, as well as for ensuring transparency and trust in these systems.

In conclusion, while integrating ML algorithms into next generation network testing poses several challenges, there are a variety of innovative solutions available. By combining advanced data processing techniques, sophisticated learning algorithms, and comprehensive testing methodologies, it is possible to overcome these hurdles and fully harness the power of ML in next generation networks.

## 3.2 Protocols and interfaces for monitoring and testing next generation systems

The next generation network is expected to integrate the terrestrial, aerial, high-altitude platform station (HAPS) and maritime communications into a robust network. This integration will be more reliable, fast, and can support a massive number of devices with ultra-low latency requirements [24]. To ensure the smooth introduction of next generation network, and the ability to monetize on it from day one, early alignment on a common set of principles that will lead toward a more focused ecosystem is advocated.

### 3.2.1 Implementing monitoring capabilities at all network layers

The high-level vision for next generation network is to deepen the connection and integration between the digital, physical, and human worlds [25]. This vision is built on the desire to create a seamless merging of the digital and physical worlds. To achieve this, monitoring capabilities need to be implemented at all network layers [24]. This enables and enhances support of continuous service assurance monitoring, to handle the growing complexity in managing and optimizing the network automation transition toward zero-touch network operation. However, this presents a significant challenge due to the complexity and diversity of the network layers.

Sample required Interfaces are Network Function Virtualization Interfaces (NFVI) (to monitor virtual network functions across different layers), and Software-Defined Networking (SDN) Controllers (interfaces for SDN controllers can provide a centralized view of the network state across all layers).

### 3.2.2 Ensuring continuous testing and security against attacks

The significant evolution of next generation network will also impact the threat landscape, adding new threat actors, and leading to a new set of cybersecurity challenges. A new concept to security monitoring and attack detection in next generation network-enabled IoT systems is introduced, leveraging on hierarchical and collaborative approaches. This approach satisfies the main next generation network's KPIs such as security, latency, connectivity, data rate, and energy consumption [26]. However, ensuring continuous testing and security against attacks is a significant challenge due to the evolving threat landscape and the complexity of next generation network-enabled IoT systems.

Sample required interfaces are Security Information and Event Management (SIEM) Systems (interfaces to SIEM systems for real-time analysis of security alerts) and Intrusion Detection/Prevention Systems (IDS/IPS) (interfaces for monitoring network traffic and identifying potential threats)

### 3.2.3 Requirements for AI/ML development cycle

Data-driven operations, distributed intelligence, continuous learning, intent-based automation, explainable and trusted AI, and cognitive system are the key enablers for the development cycle in next generation network. These enablers focus on the expected results of applying them. The tasks of the human workforce will eventually shift towards monitoring and supervising, guiding the network in its learning and decision processes as the network gradually becomes more autonomous and self-managed. However, managing the data-driven development cycle in next generation network is a significant challenge due to the complexity and volume of data involved.

Sample required interfaces are APIs for Big Data Platforms (such as Hadoop or Spark, to handle and process large volumes of data), and Machine Learning Model Management Systems: Interfaces (for managing the lifecycle of ML models, including deployment, monitoring, and updating).

## 3.3 Acquiring and validating reference evaluation datasets for next generation network AI/ML

The acquisition and validation of reference evaluation datasets are crucial steps in the development and deployment of AI/ML models in next generation networks. These datasets serve as the foundation for training, testing, and validating the performance of these models.

### 3.3.1 Dataset acquisition

In the context of next generation networks, acquiring relevant datasets can be achieved through live networks and testbeds. Live networks provide real-world data that reflects the dynamic and complex nature of next generation networks. This includes data on network traffic, user behavior, and various network conditions. Testbeds, on the other hand, offer a controlled environment where specific network scenarios can be replicated and studied. Both live networks and testbeds are invaluable sources of data for training and testing AI/ML models in next generation networks [27].

### 3.3.2 Dataset validation and certification

Once acquired, these datasets need to be validated to ensure their quality and relevance. This involves checking the datasets for completeness, consistency, and accuracy. It’s also important to certify the behavior of AI/ML models using these datasets. Certification involves verifying that the models behave as expected when

applied to the datasets. This step is crucial in ensuring the reliability and robustness of AI/ML models in next generation networks [28].

#### 3.3.3 Leveraging reference evaluation datasets

Reference evaluation datasets serve as a standard for assessing the performance of AI/ML models. By comparing the performance of a model against these reference datasets, developers can gauge the model's effectiveness and identify areas for improvement. These datasets also facilitate the comparison of different models, enabling the selection of the most suitable model for a particular application in next generation networks [27].

### 3.4 Addressing bias in AI/ML systems across different next generation network layers

The integration of AI/ML systems across different layers of next generation networks presents a unique set of challenges, one of which is addressing bias. This section discusses the understanding of sample distribution imbalance and its implications, strategies for mitigating bias at various layers of next generation networks, and the importance of ensuring fairness and equity in AI/ML-driven next generation networks.

#### 3.4.1 Understanding sample distribution imbalance and its implications

Sample distribution imbalance is a common issue in AI/ML systems, where certain classes of data are overrepresented while others are underrepresented. This imbalance can lead to biased predictions, as the AI/ML models tend to favor the majority class. In the context of next generation networks, this could mean that certain network scenarios or user behaviors are not adequately represented in the training data, leading to biased network operations [29].

#### 3.4.2 Mitigating bias at various layers of next generation networks

Mitigating bias in AI/ML systems involves adjusting the training process to ensure that all classes of data are adequately represented. This can be achieved through various techniques such as resampling, cost-sensitive learning, and ensemble methods [30]. In the context of next generation networks, this could involve adjusting the data collection process to ensure a balanced representation of different network scenarios and user behaviors.

#### 3.4.3 Ensuring fairness and equity in AI/ML-driven next generation networks

Fairness and equity are crucial considerations in the deployment of AI/ML systems in next generation networks. This involves ensuring that the AI/ML models do not favor or disadvantage any particular group of users or network

scenarios. Techniques for ensuring fairness and equity in AI/ML systems include fairness-aware learning, discrimination-aware learning, and fairness regularization.

# 4 Test-bed requirements for AI/ML

When designing a next generation testbed, it is important to consider how virtualized network functions will shape the necessary test equipment. The test architecture must align closely with the next generation system's mix of physical and virtual elements. It is not just about checking if the physical parts that hold up the next generation network are working right; it's also about making sure that the virtual parts that shift and change within the network are functioning smoothly.

For the next generation era, where AI/ML are integral, the test setup must be intelligent enough to test these technologies too. This means the testbed must have the capability to learn from the network's behavior, predict how it will act under different conditions, and adjust itself for the best testing outcomes. It should be able to simulate real-world scenarios where AI/ML applications are making decisions, to ensure that when these technologies are deployed, they will work as expected. The environment will need to handle complex algorithms and be able to validate the vast data sets that AI might use to make decisions. This attention to AI/ML within the test setup is essential because they are expected to be the brains of the next generation network, controlling its operations and optimizing its performance. In this chapter, we will delve into the crucial test-bed requirements for AI/ML implementation in network environments.

## 4.1 High-fidelity simulation/emulation

### 4.1.1 Accurately representing the network environment

Modern telecommunications, especially with the advent of next generation networks, require extremely realistic simulation environments. A misalignment between the simulated and real-world scenarios can skew results, leading to sub-optimal AI/ML solutions [31]. As an example, in Table 1, we draw a comparison between an example real-world network parameters and their simulated counterparts. It highlights the critical need for simulation environments to mirror the real-world as closely as possible, especially when preparing AI/ML systems for deployment in real network conditions. In terms of node density, the close approximation of simulated values to real-world metrics (a difference of only 50 devices/sq.km) indicates a high degree of fidelity in the simulation. Achieving this level of granularity ensures that the simulated environment can account for network congestion, device-to-device interactions, and other dynamics that are influenced by device density. In terms of bandwidth, the narrow gap between real and simulated values (only a 0.2% decrease in the simulated environment) is promising. It implies that AI/ML algorithms can be tested at nearly real-world data transmission rates, ensuring they are optimized for real network conditions [32].

On the other hands, Interference is a crucial factor in network performance, caused by overlapping signals that disrupt communication. Given that real-world interference patterns can be complex and contingent on numerous externalities (other devices, physical obstacles, etc.), "closely replicated" indicates that the simulation

strives to mirror this unpredictable nature. This is invaluable for AI/ML models that need to understand and mitigate such interference in real-time.

**Table 1 (Example) Comparison of real vs. simulated network parameters**

| Parameter | Real-world Value | Simulated Value (example) |
|---|---|---|
| Node Density | 10,000 devices/sq.km | 9,950 devices/sq.km |
| Bandwidth | 100 Gbps | 99.8 Gbps |
| Interface | Varied with environment | Closely replicated |

These three factors along with other important factors play an important role in accurate representation of network environment in a testbed.

### 4.1.2 Types of traffic, devices, and network conditions

Traffic patterns, types of devices, and network conditions can vary significantly in a next generation network environment. Table 2 provides a hypothetical breakdown of the various types of traffic we might expect on a next generation network. Devices can range from tiny IoT sensors to high-data-demanding devices like AR and virtual reality (VR) headsets or autonomous vehicles. This diverse ecosystem must be effectively simulated to understand and anticipate the demands on the real network, so a proper simulations and testbed has to be developed to anticipate these challenges and design networks that seamlessly handle them [33]. Here is a deeper dive into its columns and the listed traffic types:

**Table 2 Estimated next generation traffic patterns (hypothetical)**

| Traffic Type | Percentage of Total Traffic | Estimated Data Consumption (Exabytes/month) |
|---|---|---|
| IoT bursts | 20% | 2.0 |
| Streaming (Video/Audio) | 40% | 4.0 |
| Browsing | 15% | 1.5 |
| AR/VR sessions | 10% | 1.0 |
| Autonomous vehicles | 5% | 0.5 |
| Smart city systems | 5% | 0.5 |
| Other (gaming, communication, etc.) | 5% | 0.5 |
| **Total** | **100%** | **10.0** |

### 4.1.3 AI/ML algorithm performance evaluation

Continuous validation and recalibration of AI/ML models are essential. Three important metrics such as response time, accuracy, and reliability must be constantly monitored to ensure that AI/ML models function optimally within a telecommunication environment [34]. Table 3 provides an understanding of these metrics and what the ideal values are, especially in the context of a demanding next generation telecommunication environment. The response time KPI determines the speed at

which an AI/ML model processes input data and arrives at a decision. In telecommunications, especially in next generation where URLLC is one of the defining features, ensuring swift decision-making is essential. For instance, real-time applications such as augmented reality (AR) or autonomous driving would necessitate split-second decisions, thus demanding a response time of less than 5ms.

Accuracy is a KPI of how often the AI/ML model's decision or prediction aligns with the actual or expected outcome. High accuracy ensures that the model can be trusted to make the right decisions, a crucial attribute for mission-critical applications. For telecommunication networks, where every incorrect decision might result in significant resource wastage or even service downtime, an accuracy rate of more than 99% is desirable. Reliability KPI assesses the model's consistency in performance over time. If an AI/ML model is exposed to the same or similar input data multiple times, it should consistently provide the same or closely similar output. Such reliability ensures that once the AI/ML model is trained and deployed, it remains dependable. The ideal value for reliability is a bit conceptual, as it is more of a qualitative measure. However, "No significant variance" means that the outcomes of the model should not vary wildly over different periods or situations [35].

**Table 3 Key AI/ML Performance Metrics**

| Metric | Description | Ideal Value |
|---|---|---|
| Response Time | Time taken to make a decision | <5 ms |
| Accuracy | Correctness of decision | >99% |
| Reliability | Consistency over time | No significant variance |

## 4.2 Data collection and processing

In the next generation network era, the vast amount of data produced by numerous devices is overwhelming. Thus, efficient methods for collecting, storing, processing, and retrieving this data are crucial.

### 4.2.1 Generating and recording data

AI/ML models are only as good as the data they are trained on. As devices grow in number and variety, especially with IoT, the volume of data generated escalates. This brings forth challenges related to data collection, storage, and processing. Edge computing has emerged as a solution to these challenges. By processing data closer to the data source (like an IoT device), edge computing minimizes latency, thereby making real-time or near-real-time feedback feasible.

### 4.2.2 Handling large datasets for training and testing

In next generation telecommunication, datasets are not just vast; they are tremendous. These datasets might contain information ranging from user behavior patterns to device connection logs [36], [35]. Table 4 highlights some of the storage

solutions tailored for these substantial AI/ML datasets. Distributed Storage involves storing data across a distributed network of computers or servers, ensuring scalability (can easily add more storage as required) and fault tolerance (if one storage node fails, the system can still retrieve data from other nodes). However, its complexity lies in its setup and management.

Utilizing cloud providers' infrastructure, cloud storage is good for scalability and accessibility from anywhere. However, a recurring cost is associated with usage, and depending on the data retrieval requirements, there can be potential latency issues. The efficient indexing is not a storage solution intrinsically, but a *method* to enhance data retrieval. By indexing vast datasets, you can quickly fetch specific data subsets without scanning the entire dataset. The challenge here is the initial setup, where decisions on what, how, and when to index can get complex [37].

**Table 4 Storage solutions for AI/ML datasets**

| Solution | Pros | Cons |
|---|---|---|
| Distributed storage | Scalable, Fault-tolerant | Complexity |
| Cloud storage | Scalable, Accessible | Recurring costs, potential latency |
| Efficient indexing | Quick data retrieval | Initial setup complexity |

#### 4.2.3 Real-time data processing

The world is moving towards real-time or near-real-time analytics. Consider scenarios like autonomous driving: an autonomous car cannot wait for several seconds to decide how to react to an obstacle; it needs insights instantly. Therefore, real-time data processing is not just a luxury; it is a necessity in many next generation applications. However, the real challenge lies in optimizing algorithms for minimal latency, ensuring data integrity during rapid processing, and managing potential bottlenecks arising from simultaneous multi-node processing. Integrating these capabilities seamlessly into the existing network infrastructure is also a complex endeavor, warranting a thorough understanding of both hardware limitations and software intricacies [38].

### 4.3 Flexibility and Interoperability

#### 4.3.1 Supporting different AI/ML algorithms and configurations

As AI/ML continues to grow, it is becoming increasingly diverse and specialized. Different AI/ML algorithms are designed to solve specific problems, and therefore, the infrastructure that supports these algorithms needs to be adaptable. For instance, a neural network designed for image recognition (like a CNN) may differ considerably from one created for time series prediction (like an LSTM). The need for a flexible testing environment, capable of seamlessly transitioning between these different configurations, is therefore paramount [35].

Containerization, as mentioned, becomes an essential tool in this context. Through containerization (using technologies like Docker or Kubernetes), each AI/ML

algorithm, with its unique set of dependencies and configurations, can be isolated into a "container". This approach, as depicted in Figure 9, ensures that the algorithm runs consistently across various environments, from a developer's local machine to large-scale cloud infrastructure [39]. Additionally, it enables easy scalability, as containers can be quickly replicated or taken down as demand fluctuates.

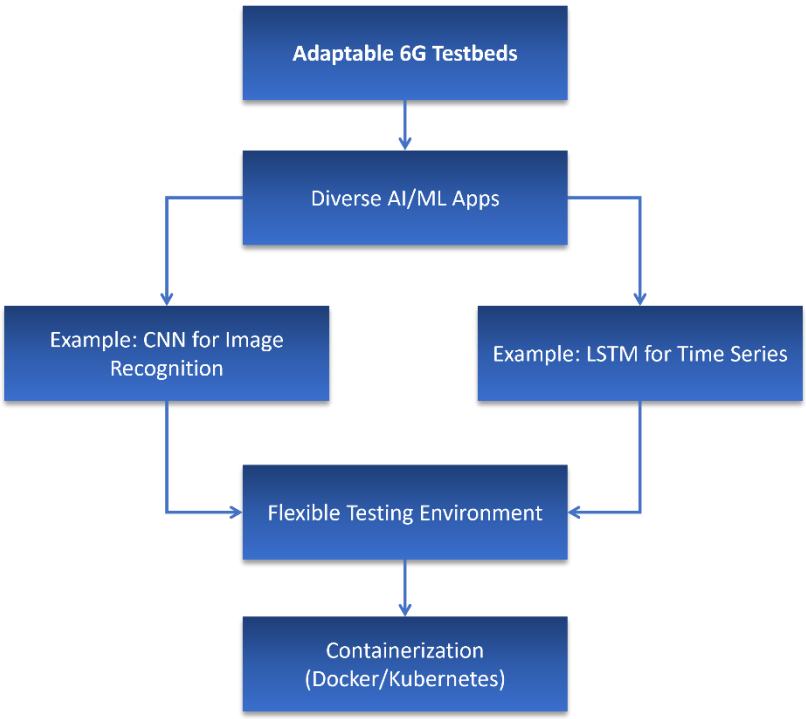


**Figure 9 An adaptable next generation network testbed supporting different AI/ML algorithms**

### 4.3.2 Compatibility with common model formats

AI/ML research and development often involve multiple tools and platforms. For example, a researcher might develop a model in TensorFlow but would want the flexibility to deploy or further refine it using PyTorch. Formats like ONNX (Open Neural Network Exchange) and NNEF (Neural Network Exchange Format) have emerged to address this need. Table 5 breaks down these formats, emphasizing how they enable AI/ML models to be more universally accepted and integrated across different platforms:

**Table 5 Common AI/ML model formats and their features**

| Format | Description | Supported Platforms |
|---|---|---|
| ONNX | Open format for ML models | TensorFlow, PyTorch, etc. |
| NNEF | Format for neural networks | Most deep learning tools |

**ONNX**: is an open standard for representing machine learning models. ONNX provides a shared model representation for frameworks like Microsoft's Cognitive Toolkit, Caffe2, and TensorFlow, allowing developers to switch between tools seamlessly.

**NNEF**: developed by the Khronos Group, it aims to enable data exchange between training systems and inference engines, especially in embedded systems and other environments where computational resources might be limited.

### 4.3.3 Interoperability with other network components

In the world of telecommunications, the scenario is further complicated with legacy systems. Many of the older infrastructures were not initially designed with the current generation of AI/ML tools in mind. Therefore, ensuring AI/ML solutions can communicate effectively with these older systems is essential.

One of the significant movements addressing this need is Open RAN. Open RAN aims to ensure that the various components in a telecommunications network, from the antennas to the base stations, can work together regardless of their manufacturer. This initiative pushes for more standardized interfaces and flexible deployments, enabling a seamless integration of AI/ML tools into the network. When AI-driven solutions are introduced into Open RAN architectures, it can lead to more dynamic network management, automated traffic routing, and enhanced user experiences [40].

This integration becomes crucial, especially when considering real-time demands like latency reduction in tasks such as video streaming or gaming. By ensuring that AI/ML solutions can interact with both current and legacy systems, we can leverage the best of both worlds: the cutting-edge capabilities of AI and the robustness and established presence of legacy systems [41].

# 5 Coordination with other standards organizations

In this chapter, we will explore how the document's research and work align with existing standards, collaborations, and initiatives led by various standards development organizations (SDOs) and entities in the field. We bring an overview of the need for coordination with other SDOs and entities. Also, we point to importance of referencing work from different bodies to ensure compatibility and interoperability. Over time we will conduct outreach initiatives to expand the engagement of user communities and expand upon the pool of traditional role players currently comprised of network operators, network equipment manufacturers and in some cases, governments.

## 5.1 Standards and organizations relevant to the research

ISO/IEC JTC 1/SC 42 Artificial Intelligence

ITU-T Y.AI4SC and Y.qos-ml

3GPP AI/ML Work Items

TMForum Artificial Intelligence Standards

Other bodies related to current 5G, and NextG communities of interest related to user and market segments including, but not limited to specialty NTNs, smart cities, energy, transportation, medicine, education, personal automation, first responders, critical infrastructure and military. The goal is to enable private network ecosystems leveraging the work of these focused bodies and those of the traditional 3GPP/O-RAN players to create as seamlessly operating environment as possible. The goal is to streamline interoperability and integration. Further, the goal is to create frames of reference and ultimately standards that allow for hardware/software re-use and flexibility of deployment of functionality across devices and the network edges, RANs and cores. This will allow for opportunities to reduce the costs of implementation and operations.

## 5.2 Challenges and benefits of coordination

Traditionally, we have grown our cellular wireless ecosystem around the efforts and structure of 3GPP, O-RAN ALLIANCE and other groups with interest in cellular wireless standards and interoperability. As has been widely noted, the marketplace has "xG fatigue" and so something transformative in both technology and ubiquitous availability and service assurance needs to be put forward to interest and accelerate adoption of 6G. The 6G architecture needs to expand and leverage AI/ML abundantly to help manage the breadth of applications and services at a cost-effective scale.

As we evolve into 6G platforms the opportunities and requirements will evolve from that of the current cellular wireless communications-centric model and features to include sensing, ubiquitous coverage, even higher speeds, security and more

developed, deployed and operated by a broader community of interests. As we expand the scope of interested parties, we will need to include them.

The insights derived from the underlying network AI/ML will be valuable and essential to those with specialized applications and edge network ecosystem ownership or operational responsibilities as will the testing methodologies, training data and "guardrails" put into place to regulate their behavior. To optimize the operations and the capital expense of these application and network platforms they will need to be integrated with those of the cellular wireless community. A lack of integration with them will cause those user communities to relegate the cellular wireless community to role of being providers of "wireless fiber" and will miss an opportunity efficiently and more comprehensively integrate and scale.

These application and edge computing communities have standards and practices associations of their own. Lacking integration with the cellular wireless community largely managed by 3GPP and O-RAN ALLIANCE, they will create "over the top" ecosystems that will often duplicate embedded 6G (or 4G/LTE & 5G) functions to provide sufficient levels of assured services and security. Further, coordination and integration with multiple standards organizations and entities would allow the 6G community to grow in influence and scope by highlighting potential benefits, requirements, ensure ensuring consistency, and leveraging expertise from diverse sources. The result would be a more tightly integrated and pervasive 6G ecosystem.

# 6 Conclusion

In our research report, we extensively explored the advanced methodologies for testing AI/ML technologies within the sphere of next generation networks, emphasizing the detailed approach required to ensure the reliability, effectiveness, and secure deployment of these systems. Highlighting the integration of AI/ML in mobile networks, the document delves into innovative strategies for creating a robust testing ecosystem. This includes the development of frameworks that accommodate the dynamic and complex nature of next generation environments, leveraging RICs, DTs, and optimizing the air interface through AI-driven approaches.

The report further elaborates on methodological frameworks tailored for evaluating AI models, from training and validation to real-world deployment. It addresses critical aspects such as mitigating biases, ensuring the robustness of AI/ML models, and maintaining their explainability. The significance of adapting testing strategies to manage the non-deterministic outcomes of AI/ML systems is underscored, highlighting the importance of predictive analytics and advanced simulation techniques in optimizing network performance and resource management.

By focusing on these areas, the report lays a comprehensive foundation for the future of telecommunications, envisioning next generation networks that are not only more efficient and secure but also inherently intelligent and adaptive. The detailed examination of testing methodologies for AI/ML in this context aims to advance the integration of these technologies, ensuring that next generation networks are equipped to meet the evolving demands of connectivity and serve as a catalyst for technological innovation and societal progress.

## References


[1] O-RAN ALLIANCE: https://public.o-ran.org/display/NGRG/Introduction

[2] O-RAN Towards 6G, O-RAN ALLIANCE, [Online]. Available: https://mediastorage.o-ran.org/ngrg-rr/nGRG-RR-2023-01-O-RAN-Towards-6G-v1_3.pdf

[3] O-RAN ALLIANCE RS02 Research Report on O-RAN Native AI Architecture Description. [Online]. Available: https://mediastorage.o-ran.org/ngrg-rr/nGRG-RR-2023-02-Native%20AI%20Architecture%20Description-v1.2.pdf

[4] O-RAN ALLIANCE RS03 Research Report on Native and Cross-domain AI: State of the art and future outlook. [Online]. Available: https://mediastorage.o-ran.org/ngrg-rr/nGRG-RR-2023-03-Research-Report-on-Native-and-Cross-domain-AI-v1_1.pdf

[5] M. Sundararajan, A. Taly, Y. Qiqi. Axiomatic attribution for deep networks. International conference on machine learning (pp. 3319-3328). PMLR, 2017.

[6] A. Shrikumar, P. Greenside, A. Kundaje, Learning important features through propagating activation differences. International conference on machine learning (pp. 3145--3153). PMLR, 2019.

[7] M. Lundberg, S.I. Lee. A unified approach to interpreting model predictions. Advances in neural information processing systems, 30, 2017

[8] M. Ribeiro, S., Singh, C. Guestrin, “Why Should I Trust You?” Explaining the Predictions of Any Classifier, In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1135--1144). 2016.

[9] Li Yang and Abdallah Shami, “On Hyperparameter Optimization of Machine Learning Algorithms: Theory and Practice”, arXiV, Oct 2022

[10] Lu, Liu, et.al., “Learning under Concept Drift: A Review”, arXiv Apr 2020

[11] S. Ackerman *et.al.,* “Automatically detecting data drift in machine learning classifiers”, arXiv Nov 2021

[12] W. Guo, Explainable Artificial Intelligence for 6G: Improving Trust between Human and Machine. IEEE Comm. Magazine, 7(1), 114-117. 2020.

[13] I. Sutskever, O. Vinyals, and V. Le Quoc, "Sequence to sequence learning with neural networks." Advances in neural information processing systems 27, 2014.

[14] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," in IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673-2681, Nov. 1997

[15] M. K. Shehzad, L. Rose, M. M. Butt, I. Z. Kovács, M. Assaad and M. Guizani, "Artificial Intelligence for 6G Networks: Technology Advancement and Standardization," in IEEE Vehicular Technology Magazine, vol. 17, no. 3, pp. 16-25, Sept. 2022

[16] L. U. Khan, W. Saad, D. Niyato, Z. Han and C. S. Hong, "Digital-Twin-Enabled 6G: Vision, Architectural Trends, and Future Directions," in IEEE Communications Magazine, vol. 60, no. 1, pp. 74-80, January 2022

[17] 3GPP, “TR 38.843 Study on Artificial Intelligence (AI)/Machine Learning (ML) for NR air interface”, 5 Jan. 2024. [online]. Available: https://www.3gpp.org/ftp/Specs/archive/38_series/38.843/38843-201.zip

[18] 3GPP, “New WID on Artificial Intelligence (AI)/Machine Learning (ML) for NR Air Interface”. 5 Jan.2024. [online]. Available: https://www.3gpp.org/ftp/tsg_ran/TSG_RAN/TSGR_102/Docs/RP-234039.zip

[19] A. Klautau, P. Batista, N. González-Prelcic, Y. Wang and R. W. Heath, "5G MIMO Data for Machine Learning: Application to Beam-Selection Using Deep Learning," Information Theory and Applications Workshop (ITA), San Diego, CA, USA, 2018, pp. 1-9. 2018.

[20] LASSE – Telecommunications, Automation and Electronics Research and Development Center. http://lasse.ufpa.br

[21] M. Al-Quraan et al., "Edge-Native Intelligence for 6G Communications Driven by Federated Learning: A Survey of Trends and Challenges," in IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 7, no. 3, pp. 957-979, June 2023

[22] J. Huang, G. Li, J. Tian and S. Li, "Accurate Interpretation of the Online Learning Model for 6G-Enabled Internet of Things," in IEEE Internet of Things Journal, vol. 8, no. 20, pp. 15228-15239, 15 Oct.15, 2021

[23] W. Yang et al., "Semantic Communications for Future Internet: Fundamentals, Applications, and Challenges," in IEEE Communications Surveys & Tutorials, vol. 25, no. 1, pp. 213-250, Firstquarter 2023

[24] M.A. Matin, S.K Goudos, S. Wan “Artificial intelligence (AI) and machine learning (ML) for beyond 5G/6G communications”. J Wireless Com Network, 22, 2023.

[25] A Patil, S Iyer, RJ Pandya, “A survey of machine learning algorithms for 6g wireless networks”. arXiv preprint arXiv:2203.08429. 2022 Mar 16.

[26] AI-powered 6G networks will reshape digital interactions. [online]. Availbale: https://www.technologyreview.com/2023/10/26/1082028/ai-powered6g-networks-will-reshape-digital-interactions/

[27] A. PATIL S. IYER, R. PANDYA, A survey of machine learning algorithms for 6g wireless networks. arXiv preprint arXiv:2203.08429, 2022.

[28] SHEHZAD, Muhammad K., et al. Artificial intelligence for 6G networks: Technology advancement and standardization. IEEE Vehicular Technology Magazine, 2022, 17.3: 16-25.

[29] A. Patel, A. Shukla and J. Bhalani, "A Comprehensive Survey on 6G Networks: Key Technologies and Challenges," 2021 International Conference on Simulation, Automation & Smart Manufacturing (SASM), Mathura, India, 2021, pp. 1-6

[30] V. K GNANAVEL, A. SRINIVASAN. Effective power allocation and distribution for 6 g–network in a box enabled peer to peer wireless communication networks. Peer-to-Peer Networking and Applications, 14: 2351-2360. 2021

[31] M. Nagy, R. Molontay, Network classification-based structural analysis of real networks and their model-generated counterparts. Network Science. Jun;10(2):146-69. 2022

[32] How We Simulate Real-World Network Conditions for Testing (testdevlab.com)

[33] C de Alwis, Q-V Pham, M Liyanage, "6G Requirements," in 6G Frontiers: Towards Future Wireless Systems, IEEE, pp.21-33, 2023

[34] M. Katz, P. Pirinen and H. Posti, "Towards 6G: Getting Ready for the Next Decade," 2019 16th International Symposium on Wireless Communication Systems (ISWCS), Oulu, Finland, pp. 714-718, 2019

[35] L. Bonati et al. "OpenRAN Gym: AI/ML development, data collection, and testing for O-RAN on PAWR platforms." Computer Networks 220: 109502, 2023

[36] 6G Technology & the Next Generation of Wireless Networks – NI, 2021. [online]. Available: https://www.ni.com/en/perspectives/6g--the-next-generation-of-wireless-communication.html

[37] The Architect's Guide to Storage for AI - MinIO Blog" discusses various storage options for machine learning and serving, including cloud storage. https://blog.min.io/the-architects-guide-to-storage-for-ai/

[38] H. A. Kholidy and S. Hariri, "Toward An Experimental Federated 6G Testbed: A Federated Leaning Approach," IEEE/ACS 19th International Conference on Computer Systems and Applications (AICCSA), Abu Dhabi, United Arab Emirates, 2022, pp. 1-6, 2022

[39] C. V. Nahum et al., "Testbed for 5G Connected Artificial Intelligence on Virtualized Networks," in IEEE Access, vol. 8, pp. 223202-223213, 2020.

[40] T. Hoeschele, F. Kaltenberger, A. I. Grohmann, E. Tasdemir, M. Reisslein and F. H. P. Fitzek, "5G InterOPERAbility of Open RAN Components in Large Testbed Ecosystem: Towards 6G Flexibility," European Wireless 2022; 27th European Wireless Conference, Dresden, Germany, 2022, pp. 1-6.

[41] L Bonati, M Polese, S D'Oro, S Basagni, T Melodia, "OpenRAN Gym: AI/ML development, data collection, and testing for O-RAN on PAWR platforms," Computer Networks. 2023 Jan 1; 220:109502.