AI/ML for Beyond 5G Systems: Concepts, Technology Enablers & Solutions

Tarik Taleb[a], Chafika Benzaïd[a], Rami Akrem Addad[b], Konstantinos Samdanis[c]

[a]tarik.taleb@oulu.fi, chafika.benzaid@oulu.fi
University of Oulu, Oulu, Finland
[b]rami.addad@aalto.fi
Aalto University, Espoo, Finland.
[c]ksamdanis@lenovo.com
Lenovo, Munich, Germany

Abstract

5G brought an evolution on the network architecture employing the service-based paradigm, enabling flexibility in realizing customized services across different technology domains. Such paradigm gives rise to the adoption of analytics and Artificial Intelligence/Machine Learning (AI/ML) in mobile communications with the ease of collecting various measurements related to end-users and the network, which can be exposed towards consumers, including 3rd party applications. AI/ML may influence network planning and optimization considering the service life-cycle and introduce new operations provision, paving the way towards 6G. This article provides a survey on AI/ML considering the business, the fundamentals and algorithms across the radio, control, and management planes. It sheds light on the key technologies that assist the adoption of AI/ML in 3rd Generation Partnership Project (3GPP) networks considering service request, reporting, data collection and distribution and it overviews the main AI/ML algorithms characterizing them into user-centric and network-centric. Finally, it explores the main standardization and open source activities on AI/ML, highlighting the lessons learned and the further challenges that still need to be addressed to reap the benefits of AI/ML in automation for beyond 5G/6G mobile systems.

Keywords: `AI/ML, Data Analytics, Intelligent Networks, Network Automation.`

1. Introduction

The 5th Generation of mobile communications (5G) and beyond [1] aims to accelerate the digital transformation across diverse business sectors, including manufacturing, entertainment, transport, agriculture, retail, logistics and health. Beyond 5G (B5G)/6G is expected to facilitate further services such as smart cities, self-sustainable operations in work-sites, drones, biosensors, transferable human skills over the Internet and allow a mixed-reality experience [2]. Such emerging business landscape drives new service requirements with extreme capacity and flexibility, diverse latency (i.e., immediate, bounded and cooperative), tight synchronization and a nearly zero packet loss that is not necessarily coupled with latency [3]. In addition, the deployment of 5G/B5G/6G may co-exist with 4G/3G, while introducing various advanced radio technologies (e.g., millimeter Wave (mmWave), New Radio (NR), massive Multiple-Input Multiple-Output (MIMO)) and a cloud-native core based on softwarization. The support of these diverse service requirements across heterogeneous networks significantly increases the operational complexity of B5G [4]. To this end, AI/ML can optimize the network and service performance, while reducing costs by enabling automation [5, 6].

AI/ML introduces the capability to learn without being explicitly programmed and can facilitate analytics, which can assist autonomous decisions making. AI/ML can bring value for Mobile Network Operators (MNOs) during network planning, optimization and operations. To this end, the reduction of operational costs is only a short term target. New revenue streams leveraging the service benefits of B5G in combination with big data are the ultimate goal, where AI/ML can play a significant role in differentiating the customer experience as well as creating innovative services. Indeed, MNOs should explore value generation in relation with new applications and platforms that offer services based on data analytics and AI/ML [7]. Offering AI/ML services for premium subscribers and towards 3rd party applications, e.g., for assuring the desired performance, can assist MNOs to enhance their services beyond connectivity and likewise the earnings [8]. For instance, autonomous driving applications can benefit from AI/ML services by receiving proactively network conditions knowledge for future vehicular locations. Hence, an autonomous driving application can effectively control the level of vehicular automation considering the expected network performance quality. Similarly, mixed-reality applications may rely on AI/ML services for assuring proactively synchronization among distributed application sources leveraging on the benefits of resource flexibility. In principle, mobile networks can adopt AI/ML services in different network segments, including:

- *Network management and orchestration* with the objective to improve network resource allocation, assure network performance and analyze efficiently failures. The use of AI/ML is expected to assist long-term optimizations, e.g., configuring Network Functions (NF) or scaling up/down resources [9]. In addition, it can benefit root cause analysis and alarm correlation.

- *Radio Access Network (RAN)* that relies on real-time or nearly real-time data for predicting and analyzing user access and radio conditions that are highly dynamic in nature. The goal is to optimize, e.g., scheduling, interference control and radio resource sharing.

- *Core network* provides control plane AI/ML services concentrating on specified sessions, flows, or User Equipment (UE), with the objective to analyze or predict users' communication behavior and mobility, security risks and assure the desired network performance.

- *Application* that focuses on optimizations (e.g., re-configuring a video codec), assessing the Quality of Service (QoS) / Quality of Experience (QoE), policy negotiation and synchronization of distributed application sources.

Typically, providing analytics is a complex process and may require a combined insight across different network segments. For instance, determining the level of user plane congestion in an area of interest requires an insight of RAN, core network resource utilization and UE throughput. The analytics data itself can also be a significant commercial asset for MNOs provided that the relevant privacy is respected (e.g., via anonymity). Customer data can be exposed to application providers or vertical segments, (e.g., for smart grid applications). In addition, customer data can feed model training and validation.

The compelling role of AI/ML in communication systems and its ability to provide network optimization and foster service intelligence is captured in several state of the art contributions. However, the focus is mainly on the algorithmic aspects considering specific network technologies. The fundamental gap still remains on the AI/ML practice and deployment solutions, especially for beyond 5G and 6G. Several limitations concentrate on the system architecture that facilitates data collection and delivery of analytics across different network segments and towards 3rd parties. Shortcomings also relate to the key AI/ML enablers, which allow consumers to discover, select, request and control AI/ML services as well as on the conditions and processes for maintaining an accurate AI/ML model.

This survey aims to fill these gaps by: (i) elaborating the applicability of AI/ML data analytics and system-level architecture components in B5G/6G mobile networks; (ii) investigating the adoption of AI/ML algorithms in various practical user centric solutions and network optimizations; (iii) summarizing the key enablers and mechanisms to automate the use of of AI/ML allowing a consumer to discover, request and control analytics services and the MNO to control the data collection and distribution efficiently; (iv) exploring AI/ML results reporting mechanisms; (v) elaborating the configuration and maintainance of AI/ML models across different vendors' equipment and among multiple MNOs; and (vi) overview the current AI/ML Standards Developing Organizations (SDOs) and open source initiatives and highlighting the potential challenges facing the enablement of AI/ML data analytics in future mobile networks. Table 1 summarizes the telecommunication network abbreviations used in the article.

The remaining of the paper is organized as follows: Section 2 overviews other state of the art surveys related to AI/ML. Section 3 elaborates the automation concepts and architecture. Section 4 explores the key AI/ML technologies. Section 5 introduces the main AI/ML algorithms and techniques, while Section 6 details the adoption of AI/ML algorithms in the 5G and beyond mobile network system. Section 7 analyses the use cases and business insight related to the adoption of AI/ML in mobile communications. Section 8 provides the lessons learned and finally Section 9 concludes this article.

2. State of the Art Surveys

Driven by the anticipated key role of AI/ML in mobile networks, numerous surveys were conducted covering various aspects in network planning, optimization and operations. Wang *et al.* [10] offers a historic overview of ML providing an in-depth analysis of various algorithms including heuristics for intelligent decision making in complex heterogeneous networks. Mao *et al.* [11] detailed an extensive insight of advanced ML techniques, i.e., Deep Learning (DL), with respect to different network layers, including physical, data link, routing, security and data compression. In particular, the survey elaborates the use of DL in modulation, coding, error correction and signal detection, channel allocation, scheduling, resource management and routing, flow identification and intrusion detection. Similarly, Kaur *et al.* [12], elaborate ML techniques for 5G and beyond detailing their impact on application and network infrastructure layers.

Preliminary efforts that adopted AI/ML techniques in networking concentrate on traffic classification, a complex problem considering the plurality of traffic

Table 1: List of telecommunication network abbreviations used in the manuscript.

Abbreviation	Description	Abbreviation	Description
3GPP	3rd Generation Partnership Project	5G	5th Generation of mobile communications
5GC	5G Core	6G	6th Generation of mobile communications
ADAES	Application Data Analytics Enablement Service	ADRF	Analytics Data Repository. Function
AF	Application Function	API	Application Programming Interface
B5G	Beyond 5G	CR	Cognitive Radio
CSI	Channel State Information	D2D	Device-to-Device
DCCF	Data Collection Coordination Functionality	DDoS	Distributed Denial of Service
EGMF	Exposure Governance Management Function	ENI	Experiential Network Intelligence
ETSI	European Telecommunication Standards Institute	gNB	Next generation NodeB
IDS	Intrusion Detection System	IoT	Internet of Things
KPI	Key Performance Indicator	LoA	Level-of-Automation
LTE	Long Term Evolution	LTE-U	LTE-Unlicensed
MDAS	Management Data Analytic Service	MDT	Minimization of Drive Test
MIMO	Multiple-Input Multiple-Output	MLFO	ML Function Orchestrator
mmWave	millimeter Wave	MNO	Mobile Network Operator
MnS	Management Service	MOS	Mean Opinion Score
NE	Network Element	NEF	Network Exposure Function
NF	Network Function	NFV	Network Function Virtualization
NFVI	NFV Infrastructure	NIDS	Network Intrusion Detection System
NOMA	mmWave Non-Orthogonal Multiple Access	NRF	Network Repository Function
NWDAF	Network Data Analytics Function	OAM	Operations Administration and Maintenance
ONAP	Open Network Automation Platform	OPNFV	Open Platform NFV
ORAN	Open RAN	PDU	Protocol Data Unit
QoE	Quality of Experience	QoS	Quality of Service
RAN	Radio Access Network	RIC	RAN Intelligent Controller
RRC	Radio Resource Control	RSRP	Refrence Signal Receive Power
RSRQ	Refrence Signal Received Quality	RSS	Received Signal Strength
RSU	Road Side Unit	S2S	Sequence-to-Sequence
SBA	Service Based Architecture	SBMA	Service Based Management Architecture
SBRA	Service Based RAN Architecture	SEAL	Service Enabler Architecture Layer
SDN	Software Defined Network	SDO	Standards Developing Organization
SFC	Service Function Chaining	SINR	Signal to Interference and Noise Ratio
SLA	Service Level Agreement	SON	Self-Organized Networks
UAV	Unmanned Aerial Vehicle	UE	User Equipment
VM	Virtual Machine	VAL	Vertical Application Layer
VNF	Virtual Network Function	WSN	Wireless Sensor Network

patterns. Initially, traffic patterns were identified based on supervised and unsupervised learning via clustering [13], while more complex methods based on DL followed applied for encrypted traffic classification as presented by Rezaei *et al.* [14]. Various ML-based solutions for traffic classification considering the different steps of ML workflow (i.e., data collection, feature extraction, dimensional reduction, model deployment), are analyzed by Pacheco *et al.* [15].

Mobility prediction received significant attention from the early AI/ML stages due to the key role it plays in enabling various optimizations in mobile networks. An overview of the state-of-the-art AI/ML algorithms pertained to mobility prediction, including Markov models, artificial intelligence and probabilistic graphical models is detailed in [16]. Network optimizations are respectively addressed by introducing AI/ML in traffic control, routing and QoS/QoE, as elaborated by Fadlullah *et al.* [17] and combined with mobility prediction as detailed by Usama *et al.* [18]. Boutaba *et al.* [19] also extensively investigated ML techniques for various key areas of networking focusing on traffic engineering, QoS/QoE, performance optimization and network security. Sun *et al.* [20] provided a survey of the recent advancements of ML in wireless communications focusing on resource management, routing, caching and mobility management, as well as energy saving and localization. Zhang *et al.* [21] studied the usage of DL in resource management, routing, scheduling, security, mobile applications and monetization. The authors also analyzed how to tailor DL to mobile environments, pointing out emerging hardware and software enablers for efficient deployments.

Various surveys are devoted to integrating AI/ML techniques into the design of SON functions (i.e., self configuration, self-optimization and self-healing). Klaine *et al.* [22] surveyed the applicability of ML including supervised and unsupervised learning as well as reinforcement learning, transfer learning and heuristics in SON. Wang *et al.* [23] reviewed the different AI techniques to devise SON functions in heterogeneous networks, including heuristics, i.e., genetic algorithms, ant colony optimization and fuzzy system. The use of distributed ML for 5G and beyond is explored by Nassef *et al.* [24] aiming to address ultra-low latency requirement and optimize communication, computation and resource distribution, while assuring privacy and security.

The use of AI/ML for network security has also attracted considerable attention in the context of risk analysis and anomaly detection. Hodo *et al.* [25], Xin *et al.* [26], Moustafa *et al.* [27] surveyed ML-based Network Intrusion Detection Systems (NIDS), considering DL and ensemble learning algorithms. Other surveys focus on the NIDS application on IoT [28] and cloud systems [29], or leverage the benefits of Software Defined Networks (SDN) to implement NIDS [30]. Mishra *et al.* [31] provided an analysis of shallow ML for anomaly, misuse or hybrid detection mechanisms, highlighting the attack detection capability. Deep Reinforcement Learning (DRL) based security methods have been surveyed in [32], while Husák *et al.* [33] surveyed forecasting methods for cyber security.

A number of surveys have attempted to encompass a broad range of potential AI/ML-driven applications. Chen *et al.* [34] analyzed the adoption of AI Neural Networks (NNs) for addressing IoT, considering resource management among Multiple Radio Access Technologies (Multi-RAT) as well as emerging applications, including Unmanned Aerial Vehicles (UAVs)-based communication, Virtual Reality (VR) and mobile edge caching. Jiang *et al.* [35] delved into the potential of leveraging ML in 5G for optimizing resource management for smart grids and Device-to-Device (D2D) networks, SON functions for small cells and energy harvesting. Jagannath *et al.* [36] reviewed the usage of ML to tackle key problems in IoT with respect to wireless communication system layers. Luong *et al.* [37] focused on DRL considering decentralized autonomous networks, wireless caching, data offloading, adaptive rate control/streaming, localization, resource management, network security and crowdsourcing. Table 4 summarizes and classifies the aforementioned surveys according to the main network attributes considered.

Table 2: AI/ML surveys in mobile networks: State-of-the-art summary.

Network Attributes	**Survey**
Clustering	[10, 20, 35]
Edge Computing	[20, 24, 34, 37]
Energy Saving/Harvesting	[20, 35]
IoT	[10, 18, 28, 36, 34]
Traffic Classification	[13, 14, 15, 17, 18, 19, 21]
Localization	[20, 21, 37, 38, 39]
Mobile APPs	[21]
Mobility Prediction	[16, 20, 21, 34, 38]
NFV & SDN	[18, 29, 30]
QoS/QoE	[18, 19, 38]
Resource Management	[10, 11, 18, 19, 20, 21, 34, 35, 37, 38, 39]
Security & NIDS	[11, 18, 19, 21, 25, 26, 27, 28, 29, 30, 31, 32, 33]
SON	[17, 18, 22, 23, 35]
Traffic Control & Routing	[11, 17, 18, 19, 20, 21, 37, 38]

The state-of-the-art surveys mainly focus on analyzing the AI/ML algorithm aspects emphasizing how to apply them into the mobile network to optimize a network attribute or resolve a specific problem. An exception is 5GPPP[1] [40], which provides an insight primarily on the network management architecture enhancements related to network planning, orchestration and diagnostics, optimization and control. However, none of the current works elaborate how to incorporate AI/ML in the 5G advanced architecture, (i.e., considering the 3GPP mobile network architecture), detailing its use in the core network, management and orchestration as well as application plane. The AI/ML architecture enhancements are related to new NFs, services and interfaces that allow a consumer to discover, request and obtain AI/ML intelligence in the form of file reports, streaming or

[1] 5G Infrastructure Public Private Partnership (5G PPP) is a joint initiative between the European Commission and European ICT industry.

notifications. In addition, this survey provides an analysis on the adoption of AI/ML algorithms for providing statistics/predictions or recommendations in both the model network and the user device considering the data needed, potential location limitations and the interaction with other network entities. It also provides a categorization of the AI/ML algorithms considering the benefits and limitations in enhancing service quality and resource optimization in 5G systems and provides an insight on the MNO mechanisms to control and automate AI/ML services.

3. Network Automation Architecture

3.1. Origins of Automation & SON

The use of automation in mobile networks has a long tradition with origins from SON. SON was introduced as a network management feature for Long Term Evolution (LTE) to achieve self-optimization, self-healing, and self-configuration. SON has applicability in various use cases such as load balancing, handover optimization, fault management, equipment configuration, etc., [41]. SON functions rely on automation, which consists of monitoring, analysis (e.g., using AI/ML), decision and execution, enabling continuous optimization. SON functions can be characterized as centralized, distributed or hybrid, i.e., with distributed operation and centralized coordination. The notion of automation is an integral component of SON, i.e., only available for use, providing execution actions. Similarly, AI/ML that assists a SON function is not visible to a consumer nor feasible to control its logic and operation. Currently, automation and AI/ML have been widely adopted in 5G with the advent of new services in the 3rd Generation Partnership Project (3GPP), allowing consumer interaction. A consumer can be a network logical or physical entity, an automation or assurance function, a service optimization tool, a human operator, or an application. The applicability of AI/ML spreads across 5G core focusing on the control plane and application, the Operations Administration and Maintenance (OAM) influencing network planning and resource configuration, and the RAN for optimizing the user experience considering radio conditions.

3.2. AI/ML & Network Automation

Network automation that relies on automation loops can be broadly categorized into two types, namely: (i) *open loops*, where a manual or another separate process intervenes in taking decisions; and (ii) *closed loops*, which execute all steps autonomously [42]. Closed loops take decisions based on the limits of a given goal, which consists of a set of parameter boundaries that can be adjusted considering the outcome of the loop. The notion of automation can be applied into the different life-cycle phases of a communication service, which include:

- *Preparation phase* focusing on the design, pre-planning, feasibility check, negotiation of service attributes.
- *Commissioning phase* that converts the communication service to network requirements.
- *Operation phase* allowing run-time operations, maintaining optimization of the communication service.
- *Decommissioning phase* de-activating network resources once no longer needed.

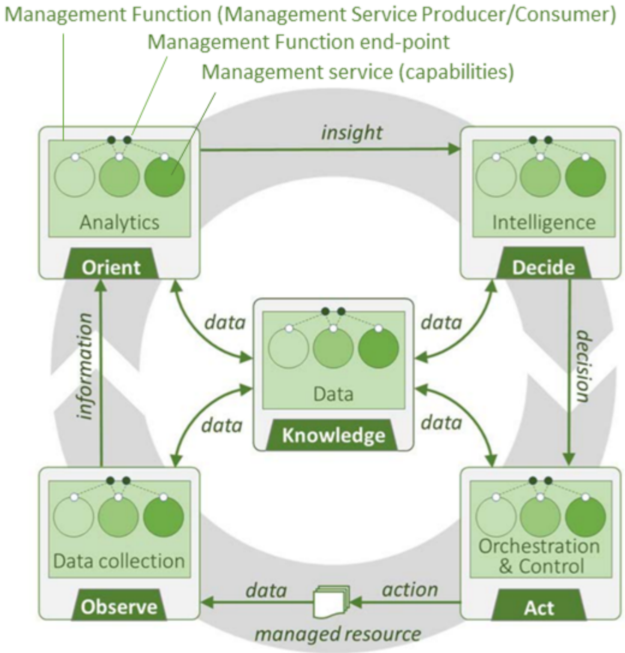


Figure 1: Closed-loop building blocks and operations [43].

Closed loops continuously observe the behavior of the entities in charge. This enables a closed loop to analyse and detect ongoing or potential deviations from a given target goal, and make decisions if actions are needed to adjust the current state accordingly [44]. A closed loop consists of the following building blocks as illustrated in Figure 1:

- *Data collection* observes and gathers data from relevant data sources.
- *Analytics* formats data before analysing it to orient and derive an insight of the past, current or future.
- *Decision* executes algorithms to achieve intelligence, i.e., recognize patterns before planning different actions (e.g., determining root causes).
- *Execution* plans to orchestrate and control actions while resolving conflicts between different goals.
- *Data lake* maintains the collected data and knowledge derived after each step is completed.

Closed loops interact with network resources via continuous iterations as well as with other peers, i.e., upper or lower level closed loops. The configuration and management of a closed loop is bounded by governance information, which can assist in configuration adjustments of the closed loop components. Network automation can be applied at different scopes, also referred to as automation layers [45], as illustrated in Figure 2. The complexity of network automation is related with the residing scope. For instance, the domain layer automation is more complex compared to a corresponding Network Element (NE) layer, since it needs to coordinate a set of NEs.

Each automation layer provides optimizations that takes place in the respective operation scope with the capability of interacting with other automation services in neighboring automation layers. ZSM elaborates the notion of network automation in GS ZSM 009-1 [46] considering governance and coordination, while it sheds light into various solutions in GS ZSM 009-2/3 [47] [48] considering resource upgrades, service deployment and configuration, as well as coordination among multi-domain loops. The autonomy scope of different automation layers may include the following:

- *Autonomy in NE layer* by introducing an automation mechanism executed in the NE; e.g., a SON functions at a base station.

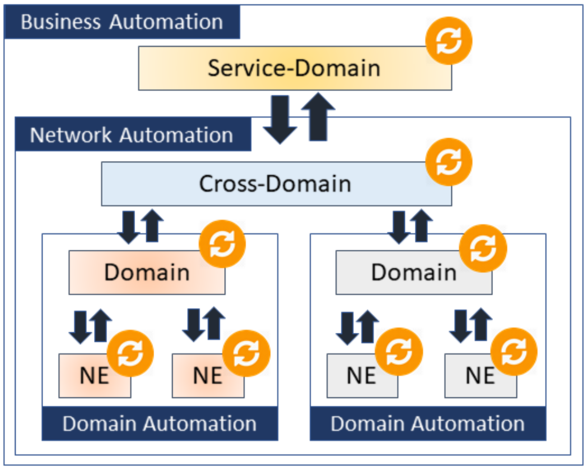


Figure 2: Scope of automation.

- *Autonomy in domain layer*, an automation mechanism executed in a 5G system domain; e.g., element manager of RAN or a NF responsible for automation in 5G core.
- *Autonomy in cross domain layer*, involving mechanisms for automation executed across different domains, e.g., across the RAN and 5G core.
- *Autonomy in communication service layer* related, e.g., to the communication service management function.

Every autonomy level can be applied in both physical and virtual resources and support distinct AI/ML models, which can be executed independently. Autonomy in NE, domain and cross-domain layers relate to network automation, while the autonomy in communication service comprise the business automation enabling the interaction with the service consumer.

3.3. AI/ML-pipeline

An AI/ML service consists of numerous logical processes or components, which are combined forming a pipeline [49]. The main components of a typical AI/ML-pipeline in future mobile networks, as suggested by ITU-T Focus Group on ML for Future Networks including 5G (FG-ML5G) [49], involve the following:

- *Source* that generates raw data (e.g., performance measurements or alarms) to feed into the AI/ML *Model*.

- *Collector*, which collects data from various sources.

- *Pre-processor* responsible for preparing the data to fit the AI/ML model by performing data processing operations, cleansing, formatting and/or aggregation.

- *Model* representing an AI/ML logic or algorithm.

- *Policy* that leverages the output of the *Model* and apply a suitable set of rules depending on the corresponding use case.

- *Distributor* in charge of identifying the *Sinks* and the distributing *Policy* to forward the output of the *Model* towards the corresponding *Sinks*.

- *Sink* is the target node of the *Distributor*.

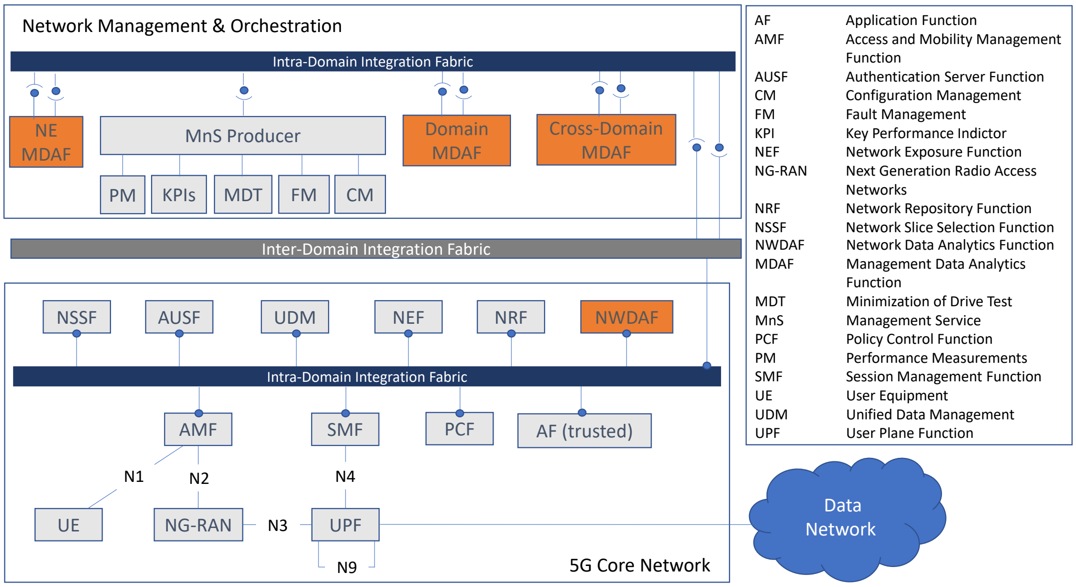
The life-cycle management of an AI/ML-pipeline relies on orchestration; i.e., provided by *ML Function Orchestrator (MLFO)*, which takes care of the configuration, scale-up/down and re-location of AI/ML-pipeline components. The *MLFO* is responsible for AI/ML service composition based on an input request or *Intent* by provisioning a flexible AI/ML service chaining. An AI/ML-pipeline can serve as a sandbox for simulation or can be applied in a real network environment directly or both. An AI/ML-pipeline deployment may span across multiple domains, i.e., RAN, 5G core and transport, which may belong to different administrative entities.

3.4. 5G Network Architecture & Micro-Services

5G architecture adopts micro-services with the advent of Service Based Architecture (SBA) in the 5G core [50]. SBA allows NFs to interact via a communication fabric relying on representational state transfer interfaces, also called RESTful interfaces [51], enabling a consumer-producer paradigm. Such architecture enables flexibility in service provision and modular upgrade of NFs in a multi vendor environment. Similarly, the Service Based Management Architecture (SBMA) [52] introduces a Management Service (MnS)[2] component toolset

[2] In 3GPP terminology, a MnS represents a management plane interface.

for building 5G management and orchestration solutions. Such an MnS component toolset consist of: (i) create, read, update and delete (CRUD) operations related to a MnS, (ii) the Network Resource Model (NRM) that allows MNOs to control and monitor the configuration of network resources, and (iii) the reporting format, e.g., related to measurements or analytics.



AF	Application Function
AMF	Access and Mobility Management Function
AUSF	Authentication Server Function
CM	Configuration Management
FM	Fault Management
KPI	Key Performance Indictor
NEF	Network Exposure Function
NG-RAN	Next Generation Radio Access Networks
NRF	Network Repository Function
NSSF	Network Slice Selection Function
NWDAF	Network Data Analytics Function
MDAF	Management Data Analytics Function
MDT	Minimization of Drive Test
MnS	Management Service
PCF	Policy Control Function
PM	Performance Measurements
SMF	Session Management Function
UE	User Equipment
UDM	Unified Data Management
UPF	User Plane Function

Figure 3: An integrated micro-services architecture for 5G core and network management.

Both architecture paradigms, facilitate service acquisition, modification and termination, while enabling access for 3rd parties, e.g., verticals, leveraging the exposure capability to assure security, service mapping and abstraction [53] [52]. An overview of an integrated micro-services architecture across the 5G core and network management is illustrated in Figure 3. In 5G core, the main SBA NF components are shown considering both the control plane and data plane (UE, NG-RAN, UPF and data network). In the network management and orchestration the MDA components are shown considering various MnS data producers, and the applicability of MDAF in the network element, domain and cross-domain levels. An inter-domain integration fabric facilitates access capabilities for endpoints across the network management and 5G core, following the Zero Touch network and service Management (ZSM) paradigm [43]. The inter-domain integration fabric is a logical entity, which represents a functionality responsible for controlling the exposure of services beyond domain boundaries and access to services exposed allowing each interconnected domain to connect via native domain interfaces.

The notion of service based can also be extended into the RAN. A service based RAN architecture may focus on Next generation NodeB (gNB) (i.e., 5G base station) split scenarios supporting a flexible combination of Distributed Units (DUs) with Centralized Units (CUs), involving also gNB control functions. AI/ML can exploit the benefits of micro-services to enrich the quality of analytics, by allowing to collect and combine analytics and observation data from different technology domains including:

- *5G core*: control plane or user-centric data, e.g., user mobility, communication patters, service experience.
- *Radio*: near-real time or real time data, e.g., interference, signal strength, pilot congestion.
- *Network management*: performance measurements, (e.g., throughput), KPIs, (e.g., end-to-end delay), fault management (e.g., alarms) and configuration management.
- *Computing and virtualization*: Central Processing Unit (CPU) load, storage, memory.
- *Application*: QoE, service sustainability, security.

The adoption of a service based architecture enables the discovery, selection and invocation of AI/ML analytics even across different domains as well as the delivery of inter domain anayltics results [43].

3.4.1. 5G Core Network Data Analytics Function

The notion of AI/ML in 5G core is introduced with the advent of Network Data Analytics Function (NWDAF) [54] [55]. NWDAF can leverage the benefits of knowledge of the UE identity to deliver various types of analytics related to, e.g., UE mobility, communications patters, positioning, traffic steering and abnormal behavior. A NWDAF can be discovered and selected using an identifier, i.e., an Analytics ID, that indicates the type of analytics. NWDAF collects data and delivers analytic results from and towards other NFs including 5G repository functions, Application Functions (AFs) and the OAM as depicted in Figure 4. It takes advantage of the 5G core counters related to, e.g., user location, session establishment/release, QoS flow monitoring, (e.g., packet delay, bit rate) and traffic volume [50][53] to derive statistics and prediction analytics. In addition, NWDAF relies on OAM data including performance measurements [56], radio conditions, (e.g., interference, received signal power), trace data and KPIs such as end-to-end latency[57], for providing complex analytics, e.g., congestion experience.

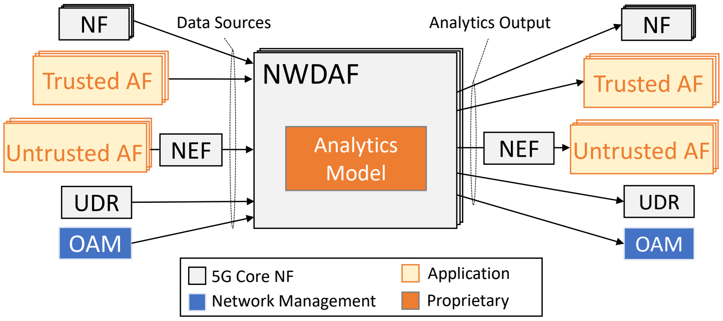


Figure 4: Overview of NWDAF.

NWDAF can also interact with applications, i.e., AFs from untrusted 3rd parties, via the exposure capability of 5G core, i.e., via the Network Exposure Function (NEF). Applications may provide NWDAF with performance observation data, e.g., with the Mean Opinion Score (MOS); data that cannot be obtained neither from the user nor from the network indirectly. NWDAF may correlate such application data with user analytics, e.g., mobility prediction, and network conditions to estimate for instance the expected QoE or QoS sustainability in future locations. NWDAF may also assist an AF to re-negotiate a policy based on the expected network conditions, which may impact, e.g., the background data transfer or the selection of the optimal edge computing location.

NWDAF is decomposed into the: (i) Analytics Logical Function (AnLF) that performs inference, derives analytics and provides the output results to subscribers, and (ii) Model Training Logical Function (MTLF), which is responsible for ML model training [58] [55]. The AI/ML model that is used in the training process is typically proprietary and may vary among equipment vendors. Network analytics has also provided training based on Federated Learning (FL) in [59], where a FL server MTLF assigns training towards FL client MLTFs handling consecutive iterations. To manage model re-training [59] introduced the notion of accuracy check by comparing the ground truth with predicted data including consumer experience, in both inference phase performed by the AnLF and training that is handled by MTLF. A comprehensive study on model accuracy considering detection, interpret, and compensation for potential performance drifts is provided in [60].

3.4.2. 5G Management Data Analytics

MDA (Management Data Analytics) enables analytics in the 3GPP management and orechstration plane [61] [62]. MDA offers analytics related to resource optimization, feasibility check, optimal resource re-configuration, average user performance (e.g., latency, jitter, throughput) and root cause analysis. Unlike NWDAF that specifies a function, MDA introduces a service, i.e., MnS or interface, called MDA Service (MDAS) or MDA MnS. MDAS can be contained in a wide variety of management functions introducing deployment flexibility, including also dedicated MDA Functions (MDAFs). An overview of MDA, pointing out the roles of MDAS and MDAF is shown in Figure 5.

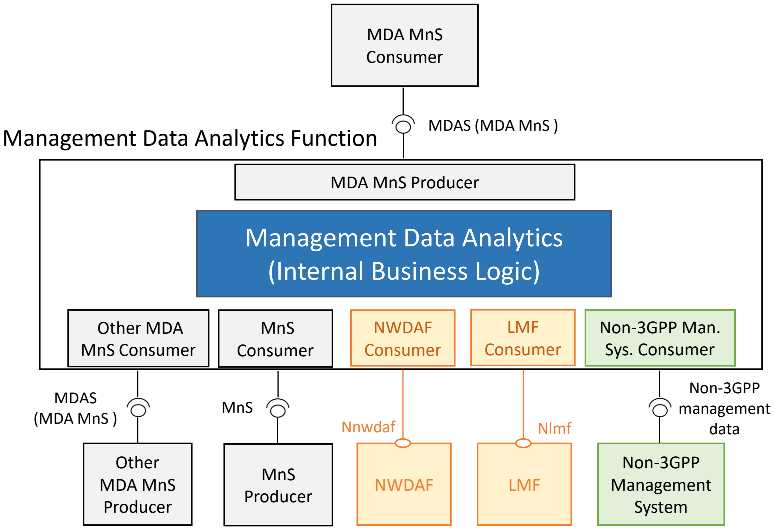


Figure 5: Overview of MDA.

A MDAF contains the respective MnS consumer part, the MDA internal business logic that represent the AI/ML model and the MDA MnS producer responsible for feeding analytics results towards other MDA MnS consumers (as shown in Figure 5). A MDA MnS consumer can potentially be a management function, a SON function, an optimization tool or even a 5G core NF. Analytics in the management plane can be acquired using an identifier, i.e., MDA type, that specifies a unique MDA capability corresponding to a predefined service. A MDAF collects OAM performance measurements [56], KPIs [57], trace data including Minimization of Drive Tests (MDT) measurements [63], alarm information [64] and configuration management data from other MnS producers. More complex analytics, e.g., service experience in relation with resource management, can be obtained from interacting with NWDAF. For instance, NWDAF may provide an indication of QoE for a group of UEs, i.e., when a certain percentage of UEs experience a MOS score below a given limit, it may trigger MDA to analyze the resource utilization. MDA on the other hand can feed NWDAF with RAN or network load analytics to complement estimations of the expected service performance.

Similarly, user location can enhance the MDA quality when correlated with configuration management or can assist analytics related to resource allocation. MDAF may use the 5G core conventional SBA interfaces, i.e., *Nnwdaf* and *Nlmf*, to collect core network data, while it provides information towards the 5G core using MDAS or MDA MnS interfaces. Certain analytics may also benefit from non-3GPP management data, which may include coverage information related to different types of networks or even data, e.g., from cameras, to complement the perception of user behaviour. Unlike NWDAF, which focuses on statistics and predictions, MDA may additionally provide recommendation options, e.g., identify a potential issue or type of problem, e.g., in terms of location, objects involved, or the optimal network configuration, e.g., endorse the use of too-late or too-early handover or dual connectivity. MDA may further need to enable analytics towards 3rd parties, exploiting the service exposure mechanisms in the management plane, called Exposure Governance Management Function (EGMF).

3.4.3. 5G Application Data Analytics Enablement Service

Application Data Analytics Enablement Service (ADAES) [65] introduces application specific analytics, i.e., predictions and statistics, for verticals or edge applications providing an insight of service parameters. The ADAE client in the UE, can provide application specific data to the ADAE server, which may also interact with the 5G core and management plane, collecting additional data if needed through network exposure. The application enablement layer hence, can expose in a unified manner analytics related to application, 5G core and network management towards verticals. ADAES comprises a Service Enabler Architecture Layer (SEAL) supporting application data analytics towards the Vertical Application Layer (VAL). Typical SEAL services towards VAL may include location management, group management, configuration management, identity/key management, network resource management and data delivery, which can be reused among different vertical applications. The ADAES architecture relies on: (i) a data collection and distribution function to coordinate efficiently the data requested or provided by ADAE server and (ii) a repository function to store historical data and analytics for future use. ADAES may support numerous value-add capabilities as documented in [66] for enabling analytics including the following:

- *Application server or session performance analytics* proactively identify application service adaptations and trigger adjustments at the communication layer.
- *Edge load analytics* related to computing and platform load, assist applications to decide when to scale-in, scale-out or migrate.
- *UE-to-UE session performance analytics* predict the performance of an application, i.e., QoS attributes, among UEs in a service group with VAL capabilities, allowing the VAL layer to pro-actively adapt.
- *Slice-related App performance analytics* provide performance insights for VAL applications utilizing a network slice and recommendations for slice (re-)configuration.
- *Location accuracy analytics* related to the deviation of a UE location from the expected one can assist applications that need positioning to decide if service adaptation, e.g., automation, is needed.
- *Service API analytics* including statistics on the successful/failed API invocation or predicted API availability can comprise a tool to be used by the API provider to help optimizing the API usage.

3.4.4. AI/ML Applications on UEs

The development of AI applications for UEs, which may include, e.g., video or speech recognition, picture optimization, intelligence assistanceand robot control, can introduce advanced 5G network requirements for supporting AI/ML operations [59]. The use of 5G network aims to anticipate UE device limitations as illustrated in Figure 6 related to storage, computing, energy and privacy, when handling AI/ML:

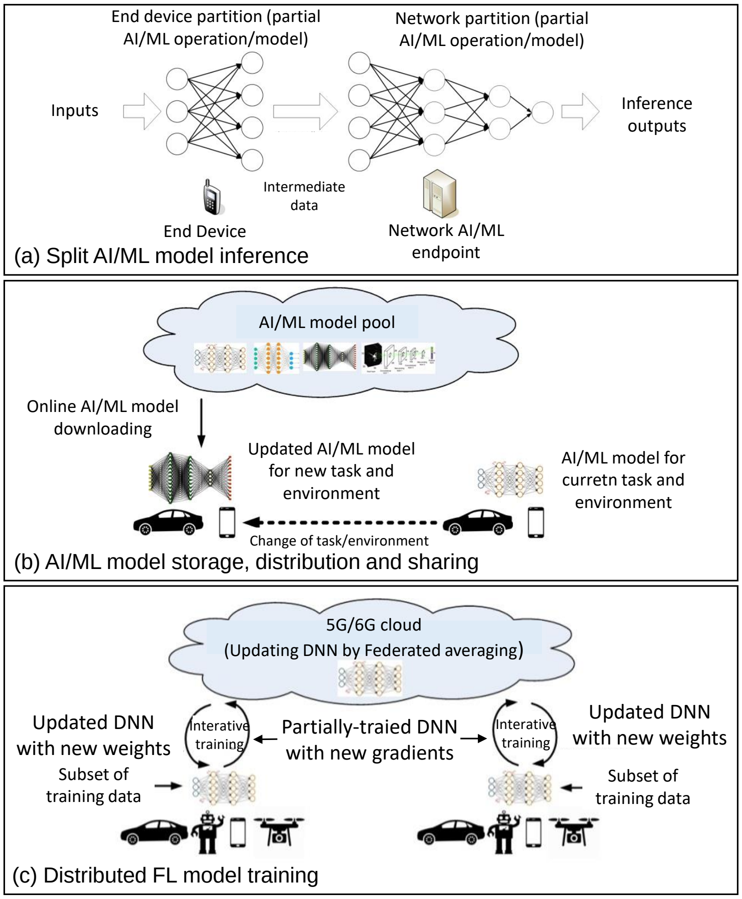

End device partition (partial AI/ML operation/model)
Network partition (partial AI/ML operation/model)
Inputs
Inference outputs
Intermediate data
End Device
Network AI/ML endpoint
(a) Split AI/ML model inference
AI/ML model pool
Online AI/ML model downloading
Updated AI/ML model for new task and environment
AI/ML model for curretn task and environment
Change of task/environment
(b) AI/ML model storage, distribution and sharing
5G/6G cloud
(Updating DNN by Federated averaging)
Updated DNN with new weights
Interative training
Partially-traied DNN with new gradients
Interative training
Updated DNN with new weights
Subset of training data
Subset of training data
(c) Distributed FL model training

Figure 6: Overview of various scenarios for supporting AI/ML applications in UEs [59].

- *Split model inference* since UEs may face certain performance or battery limitations that provide a barrier to run an AI/ML model for inference. In this circumstances, splitting an AI/Ml model across the device and an edge cloud to provide inference in a combined manner can resolve this issue as long as the network service in terms of latency and UL/DL throughput can satisfy the requirements of model inference. To achieve this a new QoS profile and policy enhancements are introduced to support application AI/ML operational traffic.

- *Multi-model storage, distribution and sharing* for UEs that may use different models for distinct situations, e.g., peak/off-peak times, on the move or at specific locations, but have no capacity to store them. In this case, UEs may fetch a model when needed without disrupting other services. To achieve this monitoring support of network resources for timely model transfer can assist in selecting the time and connectivity link for retrieving the desired model.

- *FL model training* performed by different UEs using local data to assure privacy. An aggregator application server combines local updates to create a global model which is then distributed towards the subscribed UEs. Selecting the appropriate UEs is the main challenge, since FL training needs to be performed within a specified short time window in where UEs and network shall have sufficient resources. A network assisted UE selection is proposed in [50], in where the exposure function, i.e., NEF, recommends UEs for FL training considering the network conditions and network analytics, while the application server checks the UE availability and handles the consecutive training iterations.

3.5. *Edge Computing & Open Platforms*

Edge computing [67] can facilitate AI/ML edge services closer to the user leveraging the low latency benefits of proximity. An AI/ML edge can also process data requests providing intelligence at a reduced network cost. For instance, every time somebody asks Siri or Alexa a question, the voice recording can be processed by AI/ML edge that translates the voice to text, allowing a command processor to generate the desired answer. AI/ML edge may enable on-premise services, e.g., for Industrial IoT, to improve the network performance and provide application layer services such as camera analytics. An AI/ML edge resides in the data network and can be an integral part of the 5G core being connected on the data plane via the User Plane Function (UPF) as shown Figure 7.

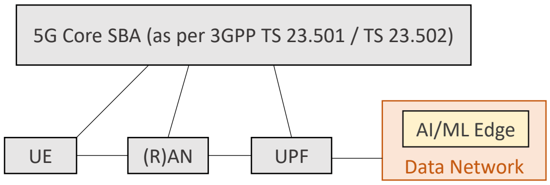


Figure 7: AI/ML edge within the 5G core SBA.

The UPF may provide accessibility to the AI/ML edge following the supplied rules based on user subscription, UE location and application information. The role of the edge in AI/ML operations can permit the following:

- *Enriched data* by allowing to collect more raw data closer to the data source.
- *Reduced data transfers via pre-processing* by filtering out irrelevant data, or by aggregating data before feeding AI/ML services.
- *Real-time data insights* by transforming raw data into analytics instantly.
- *Local intelligence* by enabling the edge to process local data for ultra-low latency applications without central cloud involvement.
- *Federated learning* by sharing local learning experience.

Besides its benefits, edge computing may introduce significant challenges in power consumption, data storage and security/privacy, since the edge may hold the majority of data and transfer only a small fraction to the cloud. Hence, careful considerations are needed in the planning phase and deployment of edge AI/ML services.

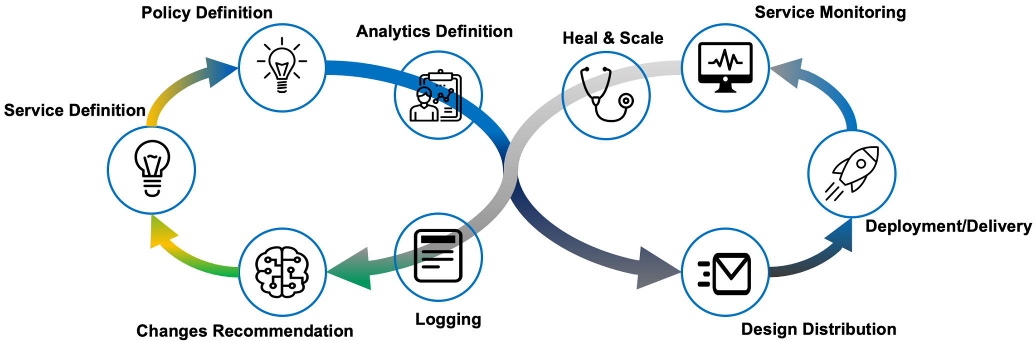


Figure 8: ONAP automation management [68].

The rapid development of AI/ML solutions in the mobile edge relies on the openness of business opportunities, i.e., allowing 3rd parties via open interfaces to enable and modify services and related features. The Open Network Automation Platform (ONAP) [69] offers an open-source life-cycle management facilitating

AI/ML capabilities, policy-aided analytics, information models and an orchestration layer, which can configure and manage services across legacy and emerging networks. An overview of automation management, mainly inspired by ONAP as defined in [68], is illustrated in Figure 8, highlighting the various phases of the service life-cycle. A de facto standard related to open source NFV platforms is defined by the Open Platform for NFV (OPNFV) [70] project, which aims to shape the open source community, e.g., OpenStack and Kubernetes. OPNFV introduces automation solutions driving the adoption of closed loop over NFVI layer and supports cloud native models and APIs considering various types of closed loops, e.g., real time, near real time and offline.

3.6. Open RAN

ORAN defines an eco-system that sheds light into open interfaces to enable multi-vendor deployments, allowing application providers and operators to introduce their own services. ORAN adopts the 3GPP RAN disaggregation paradigm, which splits a base station into a Central Unit (CU), Distributed Unit (DU), and Radio Unit (RU) component, with the CU being further split into two logical components, i.e., one for control plane and another for the user plane. This logical split allows a flexible RAN deployment on different cloud platforms and locations, with the RAN components connected via open interfaces. The O-RAN architecture [71] is the foundation for developing a virtualized RAN on open hardware introducing programmability into the RAN to optimizing radio components using closed-loop control. ORAN specifies two type of logical RAN Intelligent Controller (RIC): (i) non-real time RIC that facilitates control and RAN optimization operations greater than 1 second and (ii) near-real time RIC with control capability less than 1 second for optimizing RAN elements and radio resources.

The non-real time RIC can also provide policy-based guidance or other features in near-real-time RIC and support exposure. Each RIC type enables applications, i.e., xApp on near-real-time RIC and rAPPs on non-real-time RIC, by different vendors or 3rd parties, which are used to facilitate distinct radio features and optimizations. An overview of the ORAN architecture showing the operation of different RIC types in relation with rAPPs and xAPPs is illustrated in Figure 9. In the context of AI/ML, different xAPPs and rAPPs can offer micro-services related to AI/ML models, inference or training functionality, data collection and preparation, model management, verification and monitoring. Non-real time RIC receives the performance requirements and provides RAN configuration and analytics, supporting the deployment, training and update of AI/ML models. AI/ML models and real-time control functions are then distributed towards the

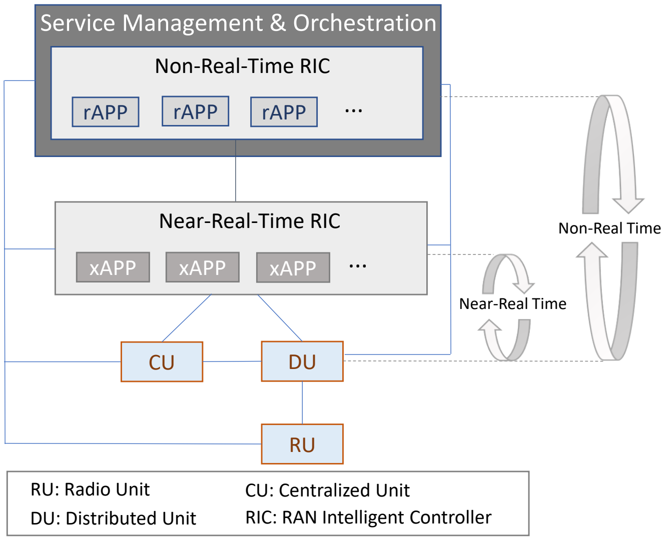


Figure 9: ORAN architecture and xAPPs/rAPPs.

corresponding near-real time RIC. The main use cases related to non-real time RIC including interface are described in [72], while the AI/ML workflow showing the roles and inter-relation of AI/ML comments is elaborated in [73]. The architecture description of the near-real time RIC is detailed in [74]. The main requirements related to the network, functional, inter-working, performance and operations are detailed in [75].

4. AI/ML Enabler Technologies

The AI/ML enabler technologies allows the use of a service considering the selection criteria, the means to request and receive analytics, the data collection for determining analytics results and the distribution for circulating the results. Specifically, the enabler technologies consist of a set of mechanisms that offer:

- a consumer the capability to discover the desired AI/ML service, and then use and control it throughout the service life-cycle,
- the MNO to control the data collection and distribution efficiently,
- policy and governance mechanisms between the consumer and the MNO to simplify and automate the use of the AI/ML services.

4.1. Discovering AI/ML Analytics

A consumer that requests an AI/ML analytics service, needs to know that it can fulfil its needs. This depends on the AI/ML service capabilities that has two-folds: (i) the type of AI/ML service, algorithm, location and other potential filtering options, (e.g., time schedule, reporting modes, target objects) and (ii) the hardware and software availability on a specific location, e.g., computing power, storage and queuing. The discovery process can take place in the following two different ways:

- *Centralized repository* wherein AI/ML analytics can register the entire range of its serving capabilities. A consumer can then place a discovery request towards the repository and receive a set of AI/ML analytics that can fulfill the request. The consumer can then select the most appropriate one based exclusively on the information received.

- *Hybrid approach* where the consumer can request a centralized entity, e.g., a Domain Name Server (DNS), to get some basic information like the IP address of a set of AI/ML analytics. In this case the centralized repository only holds basic information regarding the AI/ML service. The consumer can then request the detailed capabilities directly from the entity that offers the AI/ML service, before making a selection.

It shall be noted that the centralized approach is more simple but can only offer non-real time information regarding the AI/ML service capabilities. The hybrid one on contrary, allows the consumer to obtain the AI/ML service capabilities directly and hence get a real-time perspective. The deployment of the discovery approach depends on the trade-off between the complexity, i.e., including the singling overhead, and the AI/ML service information accuracy needs.

4.2. AI/ML Service Request & Reporting

AI/ML consumers may request a customized service by fine tuning a number of different parameters related to timing, location or particular events, shaping in this way the output reporting [55] [61]. A consumer can subscribe, issue a single request on-demand, or create a reporting job for an AI/ML service, which may contain:

- *AI/ML service type identifier* indicating the desired service, e.g., mobility prediction, NF load prediction, etc.

- *Target of an AI/ML service* that relates to the object(s) involved, including a UE, group of UEs, session or flow, area of interest, slice, or a combination, e.g., once a UE enters an area of interest.

- *Filter information*, specifies the conditions, which should be fulfilled before triggering an AI/ML service considering timing, i.e., create a report periodically or at specific times, or upon a particular event, e.g., exceeding a load threshold.

- *Immediate reporting flag* that indicates an urgent notification related to the current status of the subscribed AI/ML service, provided that it is available.

- *Target time duration* related to an AI/ML service, indicating the desired time duration in the past that a statistics report should consider or the future time duration where a prediction report should be valid for.

- *Feasibility time* that specifies until when a requested AI/ML service is needed by a consumer; this parameter limits unnecessary reports.

- *Ratio or accuracy*, (high, medium, low), related to the sampling ratio of the target objects, e.g., 60% of the UEs or 50% of the base stations in a given area of interest; this parameter regulates the cost of the AI/ML report.

- *Group reporting* indicates that a AI/ML service should be processed or aggregated before being reported to the consumer, specifying the method (e.g., average).

- *Reporting mode* that characterizes whether the reporting type should be continuous, i.e., streaming, or file based or notifications considering the filtering information.

- *Subscription reporting information* that regulates the volume of reporting by indicating the reporting times, periodicity, duration or the number of expected reports.

- *Notification address* that is used to provide the AI/ML report, which can potentially be different from the one that subscribes or creates the AI/ML service.

A consumer can modify a request by altering the involved parameters or the respective values. Since multiple instances of an AI/ML service may be deployed inside a mobile operator's network, such parameters together with the geographical context can be used to assist the AI/ML service discovery. A consumer may subscribe to an on-going or request to set-up a new AI/ML service. In other words, there are two modes of operation including:

- *Synchronous* assuming that the AI/ML producer is continuously providing analytics results using a specific AI/ML model under a regular schedule based on input data from predetermined sources. AI/ML results are constrained in terms of the input data sources, AI/ML model in use and the regularity schedule of input data. One the other hand, results are always ready and available immediately towards interested consumers.

- *Asynchronous* allows a consumer to place a new service request to select on-demand an AI/ML model and input data providing customization, i.e., offering the capability to pick the input data sources with the desired KPIs, location and time schedule, i.e., real-time or non-real-time. However, the collection of data and the AI/ML inference may introduce a delay in delivering results towards the corresponding consumer.

Once an AI/ML service producer prepares the report that contains the requested anylytics, it exposes it towards the consumer including also the following parameters:

- *AI/ML service reporting type*, indicating whether the reporting contains: (i) statistics based on past measurements, (ii) a prediction of future behavior, or (iii) recommendations of optimal parameters or configuration, with the final decision remaining at the consumer.

- *Validity period* that specifies until when a report is useful, e.g., NF load prediction for the next 20 minutes.

- *Timestamp of an AI/ML report* that defines a record related to the report generation time.

- *Confidence degree* of statistics or prediction indicating the accuracy of data reported.

- *Reporting expiration information*, which indicates on the report the corresponding AI/ML service termination in terms of timing or number of remaining reports.

4.3. Data Collection, Preparation & Distribution

Data collection is naturally the first action to take before using an AI/ML service. Typically, a consumer may indicate a target that can be a UE or group of UEs or alternatively a geographical area, network equipment or a network slice. Potentially a target is related to a network condition or event, e.g., an alarm, or application. Each analytics service, i.e., Analytics ID or MDA type, consume as input, data from pre-reconfigured, requested, i.e., customized, or private data sources. A wide variety of data can be collected using control plane related measurements [55], OAM performance measurements and KPIs [56][57] as well as user reporting such as Minimization of Drive Tests (MDT) [76] and application performance. Once data is collated, there is a need to analyze it and prepare it for use by the respective AI/ML model. Data preparation involves: i) data recovery and cleaning considering both systematic and random errors, and ii) data formatting for specific AI/ML models. The main mechanisms considered for collecting data are classified into the following categories:

- *Counter based data collection* focusing on the rate of a network procedure, e.g., handover success/failure.
- *Network resource related data collection* considering the access medium, e.g., RAN, link utilization and VNF processing and memory measurements.
- *Packet data collection* capturing per packet performance, e.g., latency or loss [77][78].
- *Flow-based data collection* related to a cluster of packets with the same characteristics, e.g., QoS flow [79] [80].
- *MDT report based data collection* based on individual UE feedback related to radio conditions, location and service performance.
- *Logs based data collection* gathering information from logs files stored in the network entities [81].

The effectiveness of data collection lies in the usefulness of information, the authenticity of data and trustworthiness of the source. Data collection and distribution can bring scalability challenges since the same data may be needed by various AI/ML services. In this case data sources need to handle multiple subscriptions and send multiple notifications containing the same data. To avoid this, 3GPP has introduced the Data Collection Coordination and Delivery (DCCF) function [55] to coordinate the collection and distribution of data. Data consumers including NWDAF may send a request to DCCF instead of the corresponding NF data source. Once DCCF receives a data request it first determines the status of data collection, i.e., it checks prior records or data profile registrations. If DCCF determines that the requested data is already being collected, it forwards it to the consumer, or if it is already stored in a data repository it provides information to retrieve it. Besides the advancement of DCCF in 5G core, the management plane has also considered to specify an equivalent data coordination service in [82]. Three distinct ways to collect and distribute data are currently considered including:

- *Real-time data* collection and distribution via streaming, can improve the AI/ML performance and responsiveness to dynamically changing conditions, e.g., related to RAN. Nevertheless, it may prove to be costly in terms of network resource consumption.
- *File based data* transfer in conjunction with filtering conditions, i.e., threshold oriented or periodic updates, can regulate the required network resources, but may impact the AI/ML accuracy [83][84][85].
- *Notification based* that handles small data, which can be transferred immediately once produced towards the consumer.

Data collected from various sources including analytics results can be stored for future used in a data repository, referred to as Analytics Data Repository Function (ADRF) in the context of 3GPP 5G core. AI/ML models can also be stored in ADRF, which can be transferred using containers, via serialization or by sharing the address from where an AI/ML model can be retrieved.

4.4. Policy & Intent Based AI/ML Services

When requesting AI/ML services, a consumer may specify certain conditions either in the form of a policy or as an intent. A policy indicates is a set of rules, typically modeled around events or conditions related to data collection, processing and reporting, e.g., once a user starts moving. A policy-based framework that obtains performance and configuration parameters to take decisions, reflecting dynamic resource alternations and varying service requirements is considered by ETSI Experiential Network Intelligence (ENI) [86] [75]. The ENI reference architecture [87] facilitates automation, service orchestration and security [88]. To assure compatibility with other systems and data formats the ENI architecture relies on the Application Programming Interface (API) broker to serve as a gateway providing translation mechanisms.

On the other hand, the notion of intent aims to simplify the networking environment by capturing the business insights. It is defined as a declaration of operational goals that a network should meet specifying what to accomplish, without indicating how to achieve them [89]. Since there is a plethora of ways to deliver an intent across the network, e.g., assuring the desired application performance may be achieved by various combinations of latency, loss and thought, it is necessary to plan and assess the effectiveness of actions taken by receiving feedback. To achieve this, closed loops are employed, which consist of the following functional building blocks:

- *User interaction* allows to communicate an intent and receive feedback, enabling users to assess whether the imposed intent has the desired effect.
- *Translation* captures an intent into policies including the relative algorithms, while it provides feedback abstracting observations to validate compliance with the intent.
- *Operations* configures the policies and course of actions across the network infrastructure and assures the desired performance using AI/ML and orchestration is aligned with the desired business outcomes.

Intent-based networking represents a learning system, which is subject to reasoning, before implementing changes over the course of time. Such learning abilities can be applied to different tasks such as translation, planning, optimization and refinement processes, enabling a continuously evolving system. GR ZSM 005 [90] elaborates various mechanisms that enable automation considering the implications on the ZSM framework architecture [43].

5. AI/ML Algorithms

This section provides an overview of the various AI/ML algorithms considering the different types of learning and the depth of learning. Table 3 summarizes the AI/ML algorithms abbreviations used in the article.

Table 3: List of AI/ML algorithmic abbreviations used in the manuscript.

Abbreviation	Description	Abbreviation	Description
AdaBoost	Adaptive Boosting	AI	Artificial Intelligence
ANN	Artificial Neural Network	ARIMA	Auto Regressive Integrated Moving Average
BRNN	Bidirectional Deep Recurrent Network	CNN	Convolutional Neural Network
DAE	Deep Auto-Encoder	DBM	Deep Boltzmann Machine
DBN	Deep Belief Network	DDPG	Deep Deterministic Policy Gradient
DL	Deep Learning	DM	Data Mining
DNN	Deep Neural Network	DP	Dynamic Programming
DQN	Deep Q-Network	DRF	Distributed Random Forest
DRL	Deep Reinforcement Learning	DT	Decision Trees
ELM	Extreme Learning Machine	EM	Expectation Maximization
ESN	Echo State Network	FDL	Federated Deep Learning
FNN	Feed-back Neural Network	GAN	Generative Adversarial Network
GBM	Gradient Boosting Machine	GMM	Gaussian Mixture Model
GNB	Gaussian Nave Bayes	GNN	Graph Neural Network
GWO	Grey Wolf Optimization	HMM	Hidden Markov Model
KNN	K-Nearest Neighbors	LR	Linear Regression
LSM	Liquid State Machine	LSTM	Long Short-Term Memory
MC	Monte Carlo	ML	Machine learning
MLP	MultiLayer Perceptron	MNN	Modular Neural Network
NN	Neural Network	PBRS	Potential Based Reward Shaping
PNN	Probabilistic Neural Network	PPO	Proximal Policy Optimization
PSO	Particle Swarm Optimization	RBM	Restricted Boltzmann Machine
RL	Reinforcement Learning	RNN	Recurrent Neural Network
SVM	Support Vector Machine	SVR	Support Vector Regression
TD	Time Difference	TL	Transfer Learning
XGBoot	eXterme Gradient Boosting Trees	XRT	Extremely Randomized Trees

5.1. *Types of Learning*

5.1.1. *Supervised Learning*

Supervised learning refers to ML algorithms trained on labeled data (i.e., inputs for which the desired outputs are known) to learn a mapping [91]. Supervised learning can be used to solve two types of problems, namely:

- **Classification problems**, where the predicted output is a discrete categorical value representing the class to which the input sample belongs. Depending on the number of classes, the classification task can be either *binary* or *multiclass*.

- **Regression problems**, rely on ML algorithms that facilitate learning of a continuous mapping function, which can be used to predict the output, e.g, the prediction of the next user's position based on the previous residing locations.

5.1.2. Unsupervised Learning

Unlike supervised learning, unsupervised learning operates over unlabeled data by uncovering hidden patterns [92]. The key problems solved by unsupervised learning include:

- **Clustering**, which groups data into clusters based on their similarity. Samples within the same cluster exhibit high similarity, while samples belonging to different clusters have low similarity.

- **Dimensionality reduction**, which focuses on compressing the data while maintaining its structure and usefulness. Dimensionality reduction is the process of reducing the input's dimension (i.e., number of features) by retaining salient and informative features, which can decrease the computational complexity.

5.1.3. Semi-Supervised Learning

Semi-supervised learning exhibits the same usage as supervised learning, but with the key difference of leveraging partially labeled data for training. In real-world applications, labeled data may be scarce or expensive, and a fully labeled data set on large scale may not be feasible [93].

5.1.4. Reinforcement Learning

Reinforcement Learning (RL) allows an agent to learn through trial and error by interacting with its environment [94]. As illustrated in Figure 5, at each time step, the agent observes the environment's state and selects an action based on its policy. By executing the selected action, the environment transits from the old state to a new state and generates a feedback in the form of reward. This reward is used by the agent to determine the optimal policy that maximizes the expected cumulative rewards. An RL problem is typically modeled either as a Markov Decision Process (MDP) or as a Partially Observable Markov Decision Process (POMDP) and can be resolved by three main learning approaches, namely [95]:

- **Dynamic Programming (DP)** that computes optimal policies given a perfect knowledge of the environment, i.e., the state transition probabilities and rewards. The assumption of a perfect knowledge of the environment dynamics makes DP algorithms of limited practical use.

- **Monte Carlo (MC)**, assumes no prior knowledge of the environment and requires only experience in the form of sample sequences of states, actions and rewards from interaction with the environment. MC learns from an episode, i.e., a complete scenario of states-actions-rewards, that leads to a terminal state.

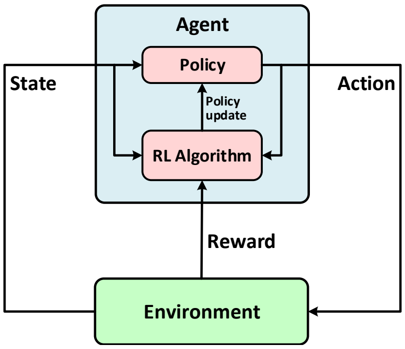


Figure 10: Reinforcement Learning.

- **Time-Difference (TD)** that allows the agent to learn the model from experiences without necessarily knowing the MDP modeling environment. Unlike MC, TD learns from an incomplete episode.

5.1.5. Transfer Learning

Transfer Learning (TL) leverages prior knowledge derived from a problem domain to solve new similar problems [96]. The capability of exploiting previous expertise enables faster learning for a new model.

5.1.6. Ensemble Learning

Ensemble learning considers multiple models combining their decisions. Common types of ensemble learning include: (i) *bagging*, which builds multiple models, each of them trained on a subset of the training data set; (ii) *boosting*, in which a set of models are built sequentially, where each subsequent model learns from the errors of the previous one; and (iii) *stacking*, which trains a supervisor model to aggregate the outcome of a set of models [97].

5.1.7. Online Learning

In online learning, ML models learn from continuous streams of data instead of the training data set at once, but it may take time before providing accurate solutions. Online learning is useful when the data is too large to fit into memory or when new data constantly arrives [98].

5.1.8. Federated Learning

Federated Learning (FL) is a distributed ML approach that aims to train a model based on local data preserving privacy. Figure 11 illustrates the masterslave paradigm that FL follows in which a number of clients or slaves, e.g., mobile devices or base stations, collaboratively train a centralized global model by aggregating their locally-computed model updates through a central server, i.e., a master, while keeping their training data localized [99]. Once updated, the new improved global model is shared with the clients and the procedure is repeated.

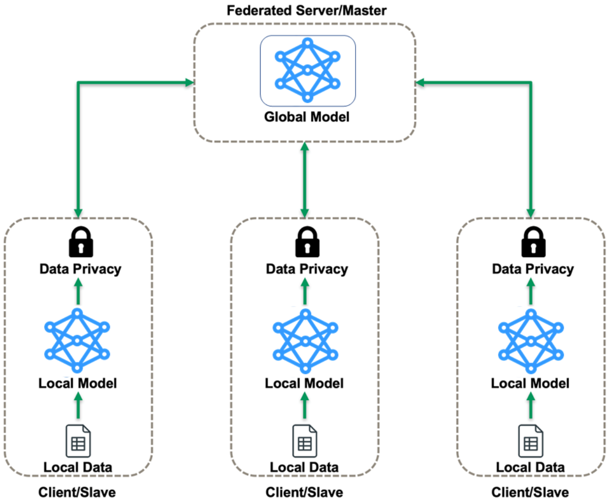


Figure 11: Federated Learning.

5.2. Depth of Learning

Considering the model architecture, ML algorithms can be divided into shallow learning algorithms and deep learning algorithms.

5.2.1. Shallow Learning

Shallow learning refers to ML methods that involve only one or two layers of input data transformation to learn the output. Shallow learning requires feature engineering to identify the relevant input features that improve the model's performance [100].

5.2.2. Deep Learning

Deep learning (DL) refers to ML techniques that rely on a multi-layered representation of the input data. The main advantages of DL over shallow learning are its ability to automatically learn useful features from multi-dimensional raw data and scale its performance with the volume of data. The popularity of DL in networking relies on recent computer technology developments (i.e., memory and processing power) [21].

5.3. Adopting Learning Types in 5G Advanced Mobile Networks

The adoption of the different types of learning in a 3GPP based 5G advanced mobile network depends on the expected architecture impact of analytics. This can be reflected on the goal of output results during the model inference and on the AI/ML model training techniques and data availability. In the 5G core, the analytics offered by NWDAF complement the decision making of other 5G NFs by offering statistics and predictions. The use of recommendations in the form of an embedded the decision logic on NWDAF would make obscure the services of other 5G NFs, e.g., if the NWDAF can recommend a UPF selection then the SMF selection service is no longer needed violating the basic consumer-producer principles of the SBA. ADAES follows similar principles to complement the consumer decisions on the application layer, but the MDA on the other hand offers additionally recommendations but for the purpose of root-cause analysis.

Table 4: Applicability of types of learning in mobile networks

Types of Learning	**Inference**	**Training**	**3GPP Architecture Element**
Supervised Learning	Statistics/Predictions	Labeled data (MNO/Verticals)	NWDAF, MDAF, ADAES
Unsupervised Learning	Pattern recognition	Unlabeled data	Data preparation
Semi-supervised Learning	Statistics/Predictions	Labeled/Unlabeled data (MNO/Vertical)	NWDAF, MDAF, ADAES
Reinforcement Learning	Recommendations	Network state/Reward	MDAF

An overview of the applicability of types of learning in mobile network considering the 3GPP architecture elements is summarized in table 4. 5G advanced is primarily considering supervised and semi-supervised learning as well as ensemble learning. AI/ML models are based on labeled data provided by the MNO or a vertical segment to facilitate statistics and predictions. Reinforcement learning is only applied for MDA recommendations related to trouble shooting, while unsupervised learning can be used in principle for the data preparation phase. Transfer learning is discussed for the scenario of UE applications, but more work is needed to identify the appropriate model transfer and the UE target. Currently, online learning is avoided due to the time it takes to reach a stable performance, but this technique may flourish with the advent of digital twins [101], which may train an AI/ML model before getting online.

The use of FL allows distributed training and data privacy in two scenarios including the 5G core network and UE AI/ML based application. In the 5G core, an NWDAF server controls the process by selecting FL clients and aggregating the updates providing the trained AI/ML model back to the requesting AnLF once the target performance requirements, i.e., in terms of accuracy or time limits, are met. For UE AI/ML based application a 3rd party AF acts as a server with the assistance of NEF, which acts as FL client discovery and selection based on the 3rd party service requirements as well as model exchange mediator. Using multiple layers of learning or running an AI/ML model can assist to split a model to accelerate the performance for both inference and training.

5.4. ML Algorithms

5.4.1. Shallow Algorithms

Several shallow ML algorithms currently exist in literature, including:

- **Linear Regression (LR)** is adopted by supervised learning to find the relation between variables to predict the next output [102]. The limitation of LR is its assumption of linearity between input and output data.

- **K-Nearest Neighbors (KNN)** is commonly used by supervised learning for both classification and regression. The core idea of KNN is that similar objects are close to each other. KNN selects K similar objects to a given item by calculating the degree of similarity, e.g., based on the Euclidean distance. KNN is easy to interpret, but its speed can be slow especially for large data sets [103].

- **Support Vector Machine (SVM)** is a supervised learning algorithm that can deal with both linear and non-linear classification and regression problems. SVM is defined by an optimal separating hyperplane that can accurately differentiate classes. For non-linear problems, kernel methods can be used to map the original input data into a higher-dimensional space, where it becomes linearly separable [104].

- **Decision Trees (DT)** is a supervised learning technique that uses a divide-and-conquer strategy, i.e., by selecting attributes of the input data, to construct a tree. The leaves of the tree represent the data labels or classes, while the non-leaf nodes represent the decision characteristics that lead to the classification [105]. While DT is robust to noisy data and easy to interpret, it is oversensitive to small changes in the input data.

- **Random Forest** is an ensemble learning technique that combines a set of DT, where each tree is constructed by randomly selected subset features and training data. The predictions of different DT are aggregated forming a decision by averaging the individual tree predictions for regression problems or taking a majority vote for classification problems [106]. Compared to DT, Random Forests provide improved accuracy, but at the price of decreased interpretation due to feature visibility.

- **Adaptive Boosting (AdaBoost)** is an ensemble learning algorithm, which can address both classification and regression problems. It combines multiple weak models to build a strong model. The weak models are used sequentially, where each subsequent model focuses on samples that are incorrectly classified by the previous model. To this end, the training data is weighted assigning higher weights to the incorrectly predicted samples [107].

- **Naïve Bayes** is a classification algorithm based on the Bayes theorem with a conditional "naïve" independence assumption between the features given the output class. A Naïve Bayes model can efficiently handle high-dimensional input data, thanks to the conditional independence assumption. However, its accuracy can significantly decrease if the features are not independent [108].

- **K-Means** is an unsupervised clustering algorithm that separates data into K groups. K-means tries to minimize the sum of the distance between each item and the center of the group, i.e., the centroid point [109]. Fuzzy C-means is a variant of K-means, where a data item can belong to more than one cluster. K-means is simple and can scale to large data sets forming clusters of different shapes and sizes. Nevertheless, it is sensitive to the initial selection of centroid points and may suffer from data outliers.

- **Expectation Maximization (EM)** is an unsupervised clustering algorithm that assumes that the data points follow a general probability distribution [110]. EM starts with a random guess related to the data distribution or clustering, and then proceeds to improve iteratively by alternating the following two steps. In the first step called expectation, it assigns each data point to a cluster probabilistically. Then in the consequent maximization step, it updates the hypothesis using the data generated in the expectation step. EM stops when the expectation and maximization steps converge.

- **Principal Component Analysis (PCA)** is an unsupervised learning technique used for reducing the dimensions of a large data set. The main idea of PCA is to identify the most valuables variables and reduce others to simplify the problem without compromising accuracy [111]. PCA relies on the use of eigenvalues to transform the original variables into new variables. Despite its benefits, PCA can lead to low performance if the original data set has a weak or no correlation.

- **Artificial Neural Networks (ANNs)** consists of multiple interconnected processing nodes, called artificial neurons, arranged into an input layer, hidden layer, and output layer. An ANN is a feed-forward neural network as the information moves only forward from the input layer, through the hidden layer towards the output layer. Inputs are real numbers forwarded via edges that typically have a weight towards an artificial neuron, which performs a computation mostly based on some non-linear function. ANN may adopt random weights initially, which can be optimized using the back-propagation algorithm [112].

5.4.2. Deep Learning Algorithms

DL algorithms are based on ANNs with multiple hidden layers between the input and output layers. The common DL approaches are presented below:

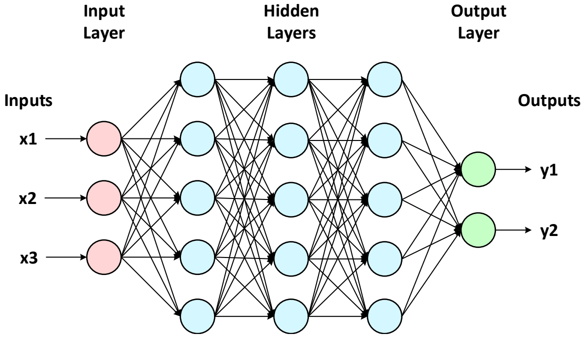


Figure 12: An MLP structure with three hidden layers.

- **MultiLayer Perceptron (MLP)**, also known as Feed-back Neural Networks (FNNs), is the quintessential deep form of ANNs. It consists of multiple fully connected layers, where every neuron is connected to all neurons in the subsequent layer [113]. Figure 12 illustrates an example of MLP with 3 hidden layers. The output propagated by a perceptron to the next layer is based on an activation function applied over the weighted sum of the received inputs plus a bias factor.

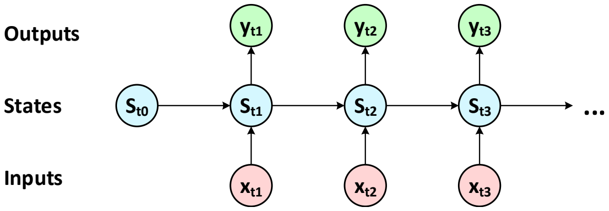


Figure 13: The basic RNN structure.

- **Recurrent Neural Network (RNN)** is designed to deal with sequential data endowed with an internal memory to keep track of past events. The internal memory is realized by feeding back the output of a hidden layer at time step t to the input of the same hidden layer at time $t+1$ as illustrated in Figure 13. RNNs are trained using Backpropagation Through Time algorithms and are ideal for time-series forecasting tasks [114]. RNN suffer from the vanishing and exploding gradient problems, introducing difficulties in training. Long Short-Term Memory Network (LSTM) and Gated Recurrent Units (GRU) are popular variants of RNN with the capability to address gradient issues [115, 116].

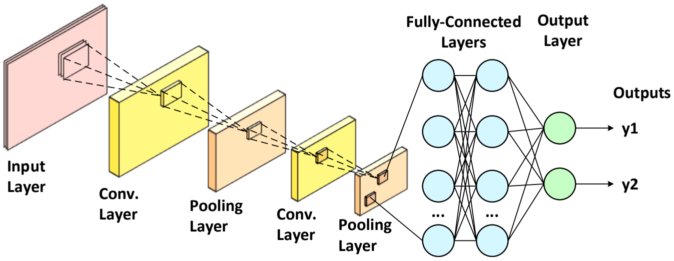


Figure 14: Typical Structure of a CNN.

- **Convolutional Neural Network (CNN)** is a feed-forward network that has neurons arranged in three dimensions: width, height and depth. The hidden layers in CNN are composed of a stack of convolutional and pooling layers, as depicted in Figure 14. A neuron inside a layer is connected to only a small set of neurons in the previous layer, called a receptive field. The purpose of the convolutional layer is to filter and extract the features. The pooling layer aims to reduce the spacial dimensions (i.e., width, height) of the input, making CNN less prone to overfitting, which allows better generalization [117]. Despite such benefits, CNNs incur high computational cost and a slow training speed.

- **Generative Adversarial Networks (GANs)** are composed of two competing

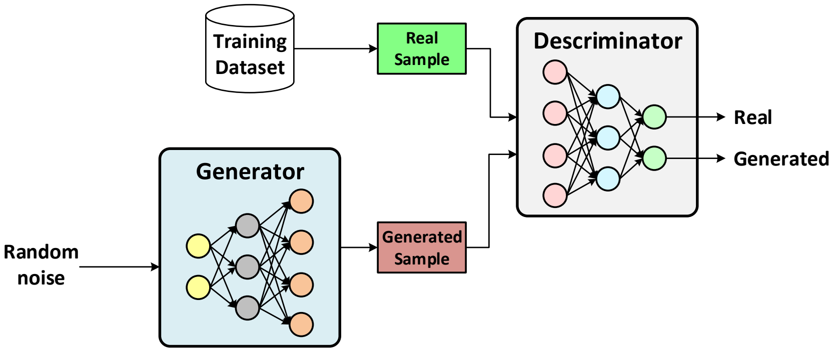


Figure 15: GAN Architecture.

NNs, a generator and a discriminator, that are trained by an adversarial process. The generator learns to generate plausible data, while the discriminator learns to distinguish the generated and real data. The training process reaches equilibrium when the discriminator can no longer differentiate the real data from the one produced by the generator [118]. Figure 15 shows the structure of a GAN. One prominent application of GANs is the introduction of training data for cases where training data may be scarce or expensive to acquire.

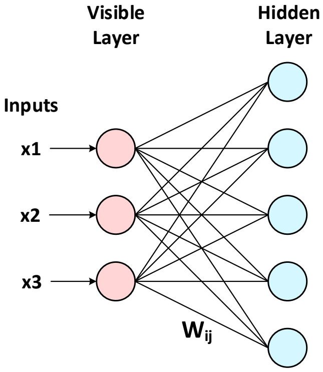


Figure 16: A typical structure of an RBM.

- **Deep Boltzmann Machines (DBMs)** are deep variant of Restricted Boltzmann Machines (RBMs) with multiple hidden layers. An RBM is a stochastic, undirected graphical model including a visible layer and a hidden layer forming a bipartite graph. All visible neurons are connected to all hidden neurons (using weights) and there are no connections between neurons of the same layer as shown in Figure 16. In DBMs, the bipartite connections are also established between adjacent hidden layers. The training of RBMs and DBMs consists in adjusting the parameters to learn the probability distribution that fits the input data [119]. Initially, used for unsupervised learning, RBMs and DBMs have also been successfully applied for automatic feature extraction.

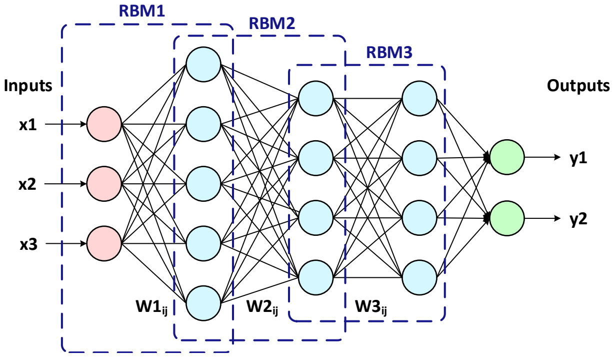


Figure 17: The structure of three-layer DBN.

- **Deep Belief Networks (DBNs)** are probabilistic generative models constructed by stacking multiple RBMs. A DBN is a hybrid graphical model including both directed and undirected connections [120]. Figure 17 illustrates a three-layer DBN. Like DBMs, DBNs aim to learn the probability distribution of the input data. The training of a DBN is performed in a greedy layer-wise manner, where each RBM is trained independently and the output of its hidden layer serves as input of the subsequent RBM. The DBNs have shown the potential to solve time-series forecasting tasks.

- **Deep Auto-Encoders (DAEs)** are unsupervised deep learning models trained to reproduce the input at the output layer. A DAE consists of two symmetric parts, an encoder and a decoder as depicted in Figure 18. The encoder converts the input into an abstraction, called code, which is then mapped back to the original input using the decoder. The training process of a DAE model aims to minimize the reconstruction error [121]. DAEs are suitable for non-linear dimensionality reduction, feature extraction, and anomaly detection.

5.4.3. Reinforcement Learning Algorithms

RL algorithms may vary depending on applicability and computation requirements. RL algorithms are mainly value-based or policy-based, but there is also

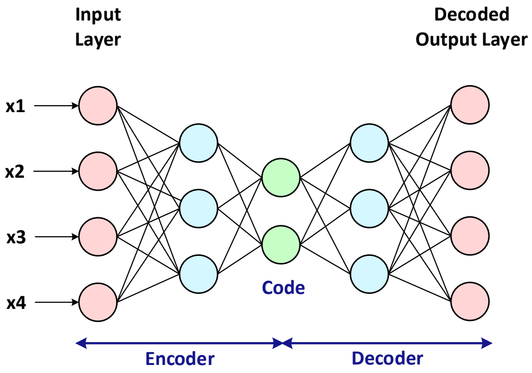


Figure 18: Typical DAE structure.

a hybrid approach that combines both methods [95]. Despite the proof of convergence, RL algorithms fail to find an optimal policy within a reasonable time in practice. Thus, the combination of RL and Deep Neural Networks (DNNs) is essential to effectively manage scalability issues.

- **Value-based RL algorithms** estimate the value or the state-action value of being in a given state. Q-learning is a prime example that uses a simple structure represented by a table, i.e., Q-table. However, the algorithm in practice is limited and inefficient [122]. Consequently, Deep Q-Network (DQN) replaces the static Q-table with a DNN. The DNN computes the quality of each action in a given state and maps states to actions. DQN relates to various algorithms associated with the value-based family, including SARSA, Double DQN [123], Dueling DQN [124], Noisy DQN [125], and DQN with Prioritized Experience Replay [126].

- **Policy-based RL algorithms** modify directly the policy instead of computing an action-state approximation for each state. Policy-based algorithms replace the value-based ones in high dimensional action-space. These algorithms address the exploration/exploitation trade-off problem leveraging stochastic probabilities for each action [127], but may converge on a local maximum rather than on the global optimum. Policy Gradients (PG) algorithms such as REINFORCE and its variants [128] are examples of policy-based RL algorithms.

- **Hybrid RL algorithms** combine both value-based and policy-based approaches [129] facilitating simplicity in selecting an algorithm considering the type of problem, learning time, and computational power. The respective agent measures the quality of actions through value-based methods, while it optimizes the policy function leveraging the policy-based methods. This category shelters many state-of-the-art algorithms such as Advantage Actor-Critic (A2C), Asynchronous Advantage Actor-Critic (A3C) [130], Deep Deterministic Policy Gradient (DDPG) [131], and Proximal Policy Optimization (PPO) [132].

6. AI/ML & Mobile Network Optimizations

This section provides a comprehensive overview related to the applicability of AI/ML algorithms described in Section 5 in mobile networks elaborating key procedures and operations.

6.1. Network & Service Optimization Assisting Analytics

Network and service assisting analytics can offer an insight to allow the MNO or a third party to optimize service and network usage. In mobile networks, the prediction of user mobility may assist other services and network operations allowing efficient and proactive resource management, service continuity and optimal location-based services [16]. Similarly, user grouping analytics can assist to improve network scalability and service experience, facilitating optimized allocation and efficient resource sharing among users.

6.1.1. Mobility Prediction

AI/ML techniques can handle large-scale data suiting the requirements for mobility prediction [133, 134]. Karimzadeh *et al.* [135] devised a hybrid Markov chain model to foresee the user's future locations. The use of ensemble learning and adaptive Markov Chain models for predicting users' position and trajectory is explored in [136]. Whilst Markov Chain-based models exhibit low computing complexity, they fail in inferring the long-term correlations between observations [137].

A dual connectivity mechanism for handover management based on trajectory prediction is detailed in [138] based on a three-layer LSTM that predicts UE's movement trends taking into account historical trajectories. Such predictions are used to determine whether a handover is required, in order to establish dual connectivity among the serving and target cells, which yields a significant improvement in service experience. Ozturk *et al.* [139] designed two novel DL-based mobility prediction models to enable proactive handover management. The proposed models leverage LSTM and MLP algorithms to predict the user's location considering historic data. The experimental results demonstrate the superiority of the LSTM-based model over MLP in terms of prediction accuracy, which yields significant benefits in reducing signaling overhead, latency and call dropping.

Mobility-awareness is also exploited to enable smart content caching strategies. Tang *et al.* [140] proposed a mobility-aware cache policy based on the RNN model for Information Centric Networks (ICN) with edge computing. The forecast locations based on historical trace are then leveraged to decide whether and where to cache proactively content, reducing significantly the access delay. Zhang *et al.* [137] use LSTM to predict target cells to be visited by a commuting mobile user. The user's preferred short video content is then pushed onto the predicted base station, resulting in enhanced user satisfaction. Hou *et al.* [141] designed an LSTM-based model to predict the subsequent moving direction of vehicles in order to empower effective proactive caching. Gebrie *et al.* [142] assessed the mobility prediction performance of four AI/ML models, namely: SVM, Semi-Markov, DNN, and eXtreme Gradient Boosting Trees (XGBoost). The models were evaluated with regard to prediction accuracy, training time and inference time, with the XGBoost model exhibiting the optimal speed-accuracy trade-off.

Wang *et al.* [143] explored ML methods for predicting both single-user and multi-user trajectory based on LSTM. The proposed LSTM suffers from poor generalization and error-accumulation effect for multi-step prediction. To overcome these issues, the authors proposed a region-oriented multi-user multi-step trajectory prediction scheme based on Sequence-to-Sequence (S2S) learning. S2S models use an encoder-decoder architecture to map an input sequence of arbitrary length into a variable-length output sequence [144]. The devised S2S model consists of an encoder LSTM network to encode the observation trajectory to a fixed-length vector and a decoder LSTM network that maps the vector to the predicted trajectory. Different mobility patterns lead to different resource re-allocation triggers, leading eventually to slice mobility when enough resources are to be migrated. A user mobility's impact on the optimal resource allocation within and between slices considering RL is analyzed in [145] investigating the applicability of two Deep RL based algorithms for allowing a fine-grained selection of mobility triggers that may instantiate slice and resource mobility actions.

Table 5 summarizes the investigated literature, highlighting the key advantages and limitations of the AI/ML techniques adopted for mobility prediction.

Table 5: Summary of AI/ML algorithms for mobility prediction.

Learning Algorithm	**Mobility Issue**	**Ref.**	**Benefits & Drawbacks**
Markov Chain	Users trajectory and locations	[135]	**- Pros:** Small space/time complexity **- Cons:** Fail to infer long-term correlations
Ensemble Learning	Users' trajectory and locations	[136]	**- Pros:** Enhanced prediction accuracy **- Cons:** High time complexity
RNN & LSTM	Proactive handover management	[138] [139]	**- Pros:** Effective in capturing long-term dependencies **- Cons:** Poor generalization and error accumulation for multi-step prediction
	Mobility-aware caching	[140] [137] [141]	
	Multi-user trajectory prediction	[143]	
XGBoost	Mobility-aware caching	[142]	**- Pros:** Optimal speed-accuracy trade-off **- Cons:** Fail to predict long-term patterns
S2S	Multi-user trajectory prediction	[144]	**- Pros:** Better generalization compared to LSTM **- Cons:** High computational and time costs
RL	Slice mobility	[145]	**- Pros:** Effective in capturing the user/ service mobility dynamics **- Cons:** Slow convergence time may hinder real-time prediction

6.1.2. User Grouping

Typical applications that leverage the benefits of user grouping include spectrum sharing in NOMA and massive MIMO environments as well as UAV grouping [146] and content caching [147]. Establishing the optimal user clustering is a combinatorial problem, where exhaustive search approaches are computationally costly due to large number of users [148]. The adoption of AI/ML techniques can determine near-optimal groupings in a reasonable time.

In [149], a K-means algorithm is elaborated to perform user grouping in a UAV communication system with MIMO antennas. Users with high channel correlation are clustered in the same group, allowing other users with low-correlated channels to be scheduled together. Trifan *et al.* [150] considered a clustering problem in a multi-user MIMO environment based on a modified K-means algorithm, where users are grouped according to the average angle between them. To tackle the local minima problem of K-means, [151] proposed a genetic algorithm based on K-means (GAK-means) to perform user grouping for obtaining the optimal UAV deployment. The aforementioned solutions consider only non-overlapping user grouping, which may lead to resource waste [152]. Neto *et al.* [153] exploited fuzzy C-means algorithm to construct overlapped user groups in mmWave systems, while [26] [154] investigated clustering techniques for dynamic user grouping. Cui *et al.* [155] proposed a K-means online user clustering to accommodate newly arriving users in a mmWave NOMA system. Ren *et al.* [154] developed an online user clustering based on the EM algorithm.

Table 6 summarizes the investigated literature, highlighting the key advantages and limitations of the AI/ML techniques adopted for user grouping.

Table 6: Summary of AI/ML algorithms for user grouping.

Learning Algorithm	**User Grouping Issue**	**Ref.**	**Benefits & Drawbacks**
K-means	Non-overlapping user grouping	[149] [150] [151]	- Pros: Simplicity and speed - Cons: Hard clustering, which leads to resource waste
	Dynamic user grouping	[155]	
Fuzzy C-means	Overlapping user grouping	[153]	- Pros: Soft clustering allowing cluster overlapping - Cons: Higher computational complexity compared to K-means
EM	Dynamic user grouping	[154]	- Pros: Probabilistic modeling and soft clustering - Cons: Higher computational complexity compared to K-means

6.2. Mobile Communications System Security

A timely detection and prediction of anomalous behaviors due to malicious actions is of utmost importance to meet the demanding reliability and sustainability requirements of 5G. 5G security focuses on both device specific aspects and network security risks.

6.2.1. Device Security

5G is expected to introduce various connected devices (e.g., IoT, vehicles) that may be prone to several threats, including spoofing and Sybil attacks, eavesdropping, jamming, and malware. Thus, a scalable real-time security risk identification and remediation is desired. An overview of ML-based IoT security focusing on access control, offloading and malware detection based on supervised, unsupervised and RL is presented in [156]. Authentication and authorization is essential in preventing attacks and controlling access privileges. The emerging authentication and authorization schemes are increasingly relying on multiple non-cryptographic attributes, related to users, resources and environment (e.g., time and location). ML/AI techniques are recognized as an appealing option to automatically combine these diverse and time-varying attributes to provide authentication and dynamically enforce fine-grained access policies [157].

Moreira *et al.* [158] propose a cross-layer authentication that considers physical layer information (i.e., RSS) and KNN to determine the authenticity of mobile terminals for network slices. In fact, KNN is used to build the authentication vector. A physical-layer authentication scheme based on Gaussian Mixture Model (GMM) is proposed in [159], which exploits the channel state information to detect identity spoofing attacks. Similarly, Liao *et al.* [38] leveraged channel state information to devise a DL-based multi-user authentication scheme to improve edge computing security. Hoang *et al.* [160] applied SVM models to detect active eavesdropping attacks, which aim at impersonating the legitimate users. To this end, the proposed SVM models use three features, namely mean, ratio and sum, extracted from wireless signals.

Fang *et al.* [161] introduced ML-based intelligent authentication by opportunistically leveraging physical layer attributes (e.g., carrier frequency offset, channel impulse response, and receiving signal strength) to achieve continuous and situation-aware authentication. In [162], a physical layer authentication approach is devised to deal with spoofing attacks in wireless networks based on adaptive CNN model, which can attune to time-varying channel attributes. A holistic authentication and authorization approach relying on online ML and trust management for achieving adaptive access control in a large-scale and dynamic IoT environment is proposed in [163]. The proposed access control scheme exploits time-varying features of the transmitter, hardware-related attributes and user behaviors, to refine access policies on run-time.

The high network performance and exposure capabilities of 5G in combination with the support for massive number of connected devices can speed up the proliferation of malware and botnets, putting user's privacy and network security at risk. In fact, mobile and IoT malware allow adversaries to steal personal data or carry out network attacks. The authors in [164] built three ML models, based on random forest, SVM and KNN algorithms, to detect ransomware. The features fed into the ML models are extracted from the storage access patterns. Sharmeen *et al.* [165] surveyed common classification models for detecting mobile malware in an Industrial IoT environment.

In self-driving vehicles, safety rely not only on high reliability and low-latency, but also on security and privacy against vulnerabilities. Self-driving vehicles require cryptographic-based security in order to be protected against an external threat and an intelligent Intrusion Detection System (IDS) to deal with threats caused by inside attackers. However, high mobility necessitate fast, reliable, and autonomous decision making with partial information collected from unknown vehicles, hence the use of RL, i.e., Q-learning, can stimulate such collaboratively report warning environment [166].

GPS spoofing is a serious threat hindering the integration of cellular-connected UAVs. To tackle this issue, Dang *et al.* [167] proposed an MLP-based model trained on the statistical properties of path loss between UAV and nearby 5G base stations to decide the authenticity of the UAV's GPS position. To further enhance the detection accuracy, the authors extended their solution by introducing a multi-MLP ensemble learning approach that integrates predictions from individual MLP models deployed at base stations [168]. In the same vein, authors in [169] leveraged the potentials of CNN and transfer learning to empower timely detection of GPS spoofing attack. CNN is used for extracting the deep features of path loss, while transfer learning is employed to transfer the CNN knowledge between edge servers during the UAV handovers.

Table 7 summarizes the investigated literature, highlighting the key advantages and limitations of the AI/ML techniques adopted for device security.

Table 7: Summary of AI/ML algorithms for device security.

Learning Algorithm	**Device Security Issue**	**Ref.**	**Benefits & Drawbacks**
KNN	Authentication and authorization	[161] [158]	**- Pros:** Low computational complexity on small datasets **- Cons:** Cannot handle large datasets with high dimensionality
	Malware detection	[164] [165]	
GMM	Authentication and authorization	[159]	**- Pros:** Ability to handle missing data **- Cons:** Unable to extract hidden features
SVM	Authentication and authorization	[160] [163]	**- Pros:** Effective even with small datasets
	Malware detection	[164] [165]	**- Cons:** Feature engineering required
MLP	GPS spoofing	[167] [168]	**- Pros:** Optimal speed-accuracy trade-off compared to other DL algorithms **- Cons:** Require labelled training datasets
CNN	Authentication and authorization	[162]	**- Pros:** Ability to extract deep features from data **- Cons:** Overfitting with small datasets and limited generalization ability
	GPS spoofing	[169]	
Ensemble Learning	GPS spoofing	[168]	**- Pros:** Enhanced detection accuracy **- Cons:** High time complexity
TL	GPS spoofing	[169]	**- Pros:** Knowledge sharing allowing fast and accurate detection **- Cons:** Fine-tuning required to improve model's generalization
Q-Learning	Trust-based intrusion detection	[166]	**- Pros:** Adapt to time-varying device behavior dynamics **- Cons:** Slow convergence

6.2.2. Network Security

AI/ML techniques can identify abnormal traffic patterns, which may result in service disruption or security risks. Indeed, the AI/ML power to unveil hidden patterns from a large-scale and time-varying data has promoted their adoption for network anomaly and intrusion detection [32, 33].

Herrara *et al.* [170] investigated ML techniques for network security in SDN environments, classifying the solutions broadly into two categories, namely: (i) ML models built to recognize general anomalies or specific network attacks, and (ii) IDS frameworks defining the whole cycle of detecting and mitigating attacks. The use of DNN for detecting intrusion in SDN networks is elaborated in [171, 172]. Mohammed *et al.* [173] devised a ML-based collaborative DDoS mitigation strategy in a multi-SDN controller environment. The detection is performed using Naive Bayes classifier based on flow features and upon detection of malicious behavior, the SDN controller is automatically notified to deny such IP based flow. Narayanadoss *et al.* [174] analyzed crossfire attacks [175], where an adversary coordinates a large number of bots to simultaneously generate low-intensity traffic in order to disconnect the target hosts or network links from the rest of the network. To counter this attack, three DL-based models were built using ANN, CNN and LSTM algorithms with an accuracy of a least 80%.

Typically, DDoS attacks concentrate on the network-layer [171, 172, 173] with a focus on saturating the network bandwidth by generating a large volume of traffic. However, the current trend in DDoS attacks concentrates on the application-layer according to Kaspersky's DDoS Q2 2019 report. An application-layer DDoS attack aims to exhaust the server's resources (e.g., CPU, memory, I/O) and disrupt the server from providing services to legitimate clients. The detection of application-layer DDoS attacks is challenging due to their stealthy nature as they seek to mimic legitimate behavior with low-bandwidth usage [176]. Siracusano *et al.* [177] investigated the capability of ML to identify low-rate application-layer DDoS using the characteristics of malicious Transport Control Protocol (TCP) flows. A detection accuracy of over 97% has been achieved using DT, KNN, and DNN techniques. The authors in [176] built a DL-based application-layer DDoS detection model that is robust to adversarial examples [157].

Mathas *et al.* [178] devised the Apache Spot ML framework, which was deployed on an SDN/NFV-enabled testbed evaluating its detection efficiency considering three different types of attacks, namely: (i) network-layer DDoS attack (UDP flooding), (ii) application-layer DDoS attack (Slowloris) and (iii) data exfiltration attack (DNS Tunneling). The Apache Spot adopts the Latent Dirichlet Allocation (LDA) algorithm [179]; an unsupervised generative probabilistic model for automatically uncovering a given number of topics (e.g., malicious/benign traffic) in a corpus of documents (e.g., network traffic within a time slot).

Detecting DDoS attacks solely through the analysis of collected network flow characteristics may not always be feasible, especially given the rise of stealthy application-layer DDoS attacks. To overcome this limitation, the work in [44] builds a LSTM-based AutoEncoder anomaly detection model that leverages resource usage (e.g., CPU usage, system load, memory usage, I/O network traffic) and performance (e.g., HTTP response time) metrics to detect application-layer DDoS attacks against network slices. The model is trained to reconstruct time-series for normal behavior; thus, an attack is detected if the reconstruction error of the observed metrics is above a given threshold.

Differently from the previous solutions that focus on the detection of DDoS attack, Javadpour *et al.* [180] devised a proactive mitigation approach that relies on slice isolation. The solution incorporates an actor-critic RL model that allows maximizing the number of satisfied slice instantiation requests on the shared infrastructure while minimizing the damage of DDoS attacks.

Table 8 summarizes the investigated literature, highlighting the key advantages and limitations of the AI/ML techniques adopted for network security.

6.3. Comforting QoE/QoS

5G systems aim to enhance the user experience and provide a plurality of network applications and services. The basis to achieve this, is to support mechanisms and operations that can comfort both QoS and QoE requirements, which are becoming more stringent.

6.3.1. QoE Assurance

QoE can be modeled and assessed depending on the type of service and its characteristics, the content, device and application, the context of use, and the user's expectations [181, 182]. AI/ML techniques can uncover complex nonlinear relationships, fostering their applicability for evaluating QoE [183, 184, 185]. Zheng *et al.* [186] designed an ANN model to estimate the MOS of video streams over Long Term Evolution (LTE) using a number of QoS metrics including jitter, delay, packet loss rate and mean loss burst size. To overcome the local minima problem in NNs and consequently enhance the achievable accuracy, Particle Swarm Optimization (PSO) [187] is applied to NN's weights aiming to reduce

Table 8: Summary of AI/ML algorithms for network security.

Learning Algorithm	**Network Security Issue**	**Ref.**	**Benefits & Drawbacks**
Naive Bayes	Network-layer DDoS attacks	[170] [173]	**- Pros:** Simplicity and support of incremental learning **- Cons:** Assume independence between traffic features
KNN	Application-layer DDoS attacks	[174] [177]	**- Pros:** Low computational complexity on small datasets **- Cons:** Cannot handle large datasets with high dimensionality
MLP	Network-layer DDoS attacks	[171] [172]	**- Pros:** Optimal speed-accuracy trade-off compared to other DL algorithms **- Cons:** Require labeled training dataset
	Application-layer DDoS attacks	[176] [177]	
RNN	Application-layer DDoS attacks	[174] [44]	**- Pros:** Effective in capturing long-term dependencies among traffic features **- Cons:** Unable to jointly capture temporal and spatial dependencies
CNN	Application-layer DDoS attacks	[174]	**- Pros:** Automated extraction of deep traffic features **- Cons:** Overfitting with small datasets and limited generalization ability
DAE	Application-layer DDoS attacks	[44]	**- Pros:** Automated features extraction and unsupervised learning **- Cons:** Need to be combined with other techniques for attack class identification
DRL	Network-layer DDoS attacks	[180]	**- Pros:** Handle large state space and continuous action spaces compared to RL **- Cons:** Computationally expensive
LDA	Application-layer DDoS attacks	[178]	**- Pros:** Generative learning **- Cons:** Slow convergence and may fail in capturing complex patterns

the model's mean square error between the desired and estimated MOS. Similarly, [188] integrates the PSO with a Probabilistic Neural Network (PNN) [189] to establish a mapping among IPTV viewing records and QoE. The type and popularity of TV channels along with network's QoS metrics are exploited to derive the QoE. In [190], the QoE prediction is performed by combining ANN with AdaBoost [191], an ensemble learning algorithm. Like PSO, AdaBoost is integrated to deal with ANN's overfitting and local optimum problems. The authors introduced a new subjective attribute, namely the viewing ratio, along with network-level QoS parameters (e.g., jitter, delay, media loss rate, average bitrate) to predict the QoE.

MLQoE [192] correlates network metrics with user feedback (i.e., MOS) to train a set of algorithms including ANN, DT, Support Vector Regression (SVR) machines and Gaussian Naive Bayes (GNB) classifiers and then select the optimal one. Lv *et al.* [193] considered the QoE prediction on imbalanced IPTV datasets; that is datasets exhibiting an unequal distribution between their classes. To this end, a multi-layer NN model is proposed, where the data's imbalanced character is solved by tuning the model's hyper-parameters (i.e., number of layers, number of neurons and activation function). The proposed model considers both network-level QoS and subjective parameters. Mao *et al.* [194] adopted the LSTM network to perform IPTV QoE forecast. The devised model considers both objective and subjective attributes. The objective factors include delay, LPR, media loss rate, jitter and average bit rate, while the subjective parameters encompass the type of service (e.g., live TV, video-on-demand) and the viewing time ratio. The conducted experiments showed that the proposed model outperforms KNN, SVM and CNN algorithms.

Unlike other contributions, [195] estimates the QoE of multimedia services based on network-level QoS parameters. The prediction is achieved using a Modular Neural Network (MNN) consisting of heterogeneous DBNs. Each DBN is in charge of QoE/QoS mapping for a specific multimedia service. The results of different DBNs are then integrated to produce the final QoE prediction. The authors in [196] focused on online QoE prediction model based on multiclass incremental SVM, while allowing to handle large-scale non-stationary data.

Table 9 summarizes the investigated literature, highlighting the key advantages and limitations of the AI/ML techniques adopted for QoE assurance.

6.3.2. *QoS Class Prediction*

QoS prediction is of utmost importance for improving the network performance and fulfill SLA requirements. In fact, QoS prediction is beneficial for

Table 9: Summary of AI/ML algorithms for QoE assurance.

Learning Algorithm	**QoE Assurance Issue**	**Ref.**	**Benefits & Drawbacks**
MLP	Specific multimedia services	[186] [190]	**- Pros:** Optimal speed-accuracy trade-off compared to other DL algorithms **- Cons:** Require labeled training datasets
PNN	Specific multimedia services	[188]	**- Pros:** Enable probabilistic predictions and memory-based learning **- Cons:** Usage limited to classification tasks
AdaBoost	Specific multimedia services	[190]	**- Pros:** Enhanced prediction accuracy of weak classifiers **- Cons:** Very sensitive to noisy data and outliers
SVR	Specific multimedia services	[192]	**- Pros:** Excellent generalization and high robustness to outliers **- Cons:** Computationally expensive on large datasets
GNB	Specific multimedia services	[192]	**- Pros:** Simplicity and speed **- Cons:** Assume normal distribution of data
LSTM	Specific multimedia services	[194]	**- Pros:** Effective in capturing long-term temporal dependencies **- Cons:** High computation costs and complexity
SVM	General multimedia services	[196]	**- Pros:** Effective even with small datasets **- Cons:** Feature engineering required
DBN	General multimedia services	[195]	**- Pros:** Probabilistic and generative modeling **- Cons:** High computational costs

MNOs to develop efficient network planning strategies, while supporting flexibly emerging new services. Moysen *et al.* [197] exploited network data and MDT reports to feed an ML planning tool for determining an appropriate network deployment layout based on the predicted QoS that should be offered to end-users. Similarly, [198] focuses on devising a QoS scheme to assist MNOs in future network planning based on MDT reports. To this end, ensemble learning based on different supervised algorithms, specifically KNN, ANN, SVM and DT, is leveraged to predict the physical resource block per offered megabit.

Torres *et al.* [199] proposed an ML-based approach for forecasting the average downlink throughput per user for a specific LTE cell. The average throughput is predicted using a variant of Autoregressive Integrated Moving Average (ARIMA) and supervised naive persistence model. A QoS forecast based on signal strength, while considering external information about the weather conditions is explored in [200] leveraging ARIMA and RNN (i.e., LSTM, CNN-LSTM) for signal strength prediction. The obtained results show the superiority of the LSTM model in predicting sudden signal strength changes, thanks to its ability of capturing non-linear relationships. A smart framework related to radio resource management, i.e., scheduling, offering high flexibility to reflect dynamic network conditions and cope with QoS provisioning for heterogeneous traffic is introduced in [201]. RL and ANN are jointly employed to determine suitable scheduling decisions based on current networking conditions. A DRL-based QoS-aware secure routing for IoT devices leveraging the benefits of SDN is detailed in [202], guaranteeing QoS by extracting knowledge from history traffic demands.

Table 10 summarizes the investigated literature, highlighting the key advantages and limitations of the AI/ML techniques adopted for QoS class prediction.

6.4. Mobile Edge Computing Intelligence

Mobile edge cloud computing in close proximity with the RAN can offer storage and computational capabilities, reducing latency for mobile service while allowing to utilize more efficiently the mobile core network. Knowing where, when and how to exploit edge computing can bring significant benefits for both end user and MNOs.

6.4.1. Computational Offloading

Given the high workload demands of emerging applications, UEs may frequently suffer from computation efficiency and energy consumption limitations. Computational offloading may resolve such performance inefficiencies. However,

Table 10: Summary of AI/ML algorithms for QoS class prediction.

Learning Algorithm	**QoS Class Prediction Issue**	**Ref.**	**Benefits & Drawbacks**
SVM	Network planning	[197] [198]	**- Pros:** Effective even with small datasets **- Cons:** Feature engineering required
KNN	Network planning	[198]	**- Pros:** Low computational complexity on small datasets **- Cons:** Cannot handle large datasets with high dimensionality
MLP	Network planning	[198]	**- Pros:** Optimal speed-accuracy trade-off compared to other DL algorithms **- Cons:** Require labeled training datasets
DT	Network planning	[198]	**- Pros:** Simplicity and interpretability **- Cons:** Fail to capture complex patterns from continuous data
ARIMA	UE QoS prediction	[199] [200]	**- Pros:** Handle seasonality and trends patterns **- Cons:** unable to capture complex non-linear patterns
LSTM	UE QoS prediction	[200]	**- Pros:** Effective in capturing long-term temporal dependencies **- Cons:** High computational costs and complexity
CNN	UE QoS prediction	[200]	**- Pros:** Automated extraction of deep features **- Cons:** Overfitting with small datasets and limited generalization ability

selecting the appropriate UEs and MEC platforms that can maximize these benefits is a complex decision. AI/ML can be used to select UEs eligible to offload, optimizing both the application quality and network resource usage.

Zhu *et al.* [203] focused on computation offloading in MEC environments based on an DRL self-adaptive algorithm. The computation problem was formulated in terms of energy and time optimization considering the user experience. The authors exploited the users' position, access points placement, and the workflow components to model the state space of the RL problem. Three actions were defined, i.e., process locally, offload, and change access point before offloading. A weighted sum of experience related to the workflow execution time and UE energy consumption is used as a reward function. The challenges of task dependent offloading towards MEC are addressed in [204] using a DRL-based agent to minimize the latency while discovering shared patterns behind various applications. The state space is represented as a Directed Acyclic Graph (DAG) holding the complete offloading plan with a vector containing the initial tasks. Tasks can be offloaded to a MEC server or executed in a local device with the reward function reflecting the total latency. By using DAG as an input feature, the authors were able to convert the offloading problem into an S2S prediction where the encoder and decoder are implemented by RNN and trained using the PPO algorithm.

To overcome computation efficiency and energy consumption limitations when offloading towards a MEC platform, the work in [205] proposes a DRL-based agent solution that optimally selects among local processing and task offloading. The authors use the SINR and computational capacity (e.g., CPU cycles per second) of a UE as the main input features and formulate a reward function as a weighted sum combining the total overhead related with local and offloaded computations. Yao *et al.* [206] provided a comprehensive procedure on detecting whether to execute locally or offload tasks to MEC. The authors designed and implemented an adaptive algorithm based on data forms with different data sizes and priorities. The proposed algorithm employs a Deep Q-Network-based learning method to remedy high dimensional space issues. Liu *et al.* [207] proposed a vehicle-assisted offloading scheme for UEs, which exploits both vehicles and MEC nodes to provide smart offloading. The authors initially formulated the problem as a semi-Markov process model and then designed two distinctive solutions based on RL and DRL considering the delay of the computation task and optimal resource allocation. Numerical results showed that the proposed approaches based on Q-Learning and Deep Q Network outperform both fixed and vehicular edge server with respect to CPU consumption and latency.

A binary offloading policy regarding wireless powered MEC networks is elab-

Table 11: Summary of AI/ML algorithms for computational offloading.

Learning Algorithm	**Computational offloading Issue**	**Ref.**	**Benefits & Drawbacks**
Q-learning	Energy and time constraints	[207]	**- Pros:** Adapt to time-varying parameters of the environment **- Cons:** Slow convergence
DRL	Energy and time constraints	[203] [207] [209] [210]	**- Pros:** Handle large state spaces and continuous action spaces compared to RL **- Cons:** High computational cost
	Offloading decisions	[206] [208]	
	Task dependency offloading	[204] [205]	

orated in [208] considering a DRL-based algorithm. The authors introduced an adaptive procedure to automatically adjust various parameters reducing the complexity and thus allowing to handle large scale networks in a reasonable time. Aiming to reduce the offloading latency in a joint NOMA and MEC environments, the authors of [209] launched a DRL-based solution considering the DQN algorithm. The proposed approach concurrently selects the set of end-users eligible to offload without losing the system's quality. The authors modeled the set of actions and states based on a matching scheme in which each user finds its associations (i.e., other users) susceptible to maintain the offloading latency reduced. An efficient and adaptive offloading mechanism for time-varying networks in MEC environments is proposed in [210]. The authors formulate the problem as a Markov Decision Process, then consider DRL (i.e., Double DQN) followed by a Q-function decomposition to simplify it and deliver an optimal offloading policy in reduced computational time. The decomposition method is used to reduce action-state spaces, constituted initially from task queue state, energy queue state, and the channel quality between users and base stations.

Table 11 summarizes the investigated literature, highlighting the key advantages and limitations of the AI/ML techniques adopted for computational offloading.

6.4.2. Edge Caching

Edge caching is recognized as a promising solution with the potential to alleviate back-haul traffic, boost network throughput, shorten service latency and improve user experience [211] [212]. However, enabling edge caching introduces three main challenges including: (i) where to cache (i.e., location); (ii) what to cache (i.e., content popularity); and (iii) how to cache (i.e., caching objectives, such as offloading and QoE) [213]. The success of AI/ML in addressing complex and dynamic problems has fostered its applicability for edge caching with increasing efforts focusing on RAN vehicular, UAV and D2D.

Bharath *et al* [214] leverages a transfer learning method to design a caching strategy in heterogeneous small cell networks. The caching policy is derived without prior knowledge of the time-varying popularity profile of cached contents. In fact, the popularity profile is estimated using transfer learning by exploiting the content demand history. A proactively content caching scheme relying on RL is introduced in [215], targeting an edge node environment where new content is released and user preferences change over time. Historic data is leveraged to predict the future content requests using a Grouped Linear Model (GLM), a variant of the multivariate linear model. A model-free RL approach is then applied to learn the cache replacement strategy considering both cache hits and replacement costs.

Zhong *et al.* [216] elaborated a DRL-based framework using the Wolpertinger architecture [217] for content caching at a single edge node (e.g., a base station). The Wolpertinger architecture is leveraged to narrow the action space size. The proposed framework makes appropriate cache replacement decisions based on users' requests, maximizing both long-term and short-term cache hit rates while reducing runtime. Yang *et al.* [218] proposed a content popularity based clustering, grouping users that share similar interests, which are served cooperatively by a set of base stations. The caching strategy is derived using an ε-greedy Q-learning model, yielding a near-optimal solution. Zhang *et al.* [219] explored proactive caching for multi-view 3D videos in 5G using a Markov decision process, which jointly considers views selection and local memory allocation. The proactive caching strategy is determined using a model-free DRL approach. The reward function is formulated as a combination of cache cost and quality of video streaming. The proposed solution is shown to be effective in maintaining the desired QoE for high-mobility users in small cell environments.

DeepCachNet [220] is a DL-based proactive caching framework for cellular networks that estimates the content popularity based on the users and content features, which are extracted using auto-encoder and stacked denoising autoencoders. The predicted popularity is used to cache strategic content to achieve higher backhaul offloading and user satisfaction. Similarly, [221] presents an online proactive caching strategy employing a Bidirectional Recurrent Neural Network (BRNN) model to forecast time-varying content requests and update edge caching accordingly. A deep multi-agent RL-based caching mechanism presented in [222] determines whether to cache a request as well as the cache replacement strategy taking into account the requests' priority. The authors in [223] investigate a joint communication, caching and computing design problem for edge-based vehicular networks. A deep Q-learning model is proposed to solve such problem by determining the subset of Road Side Units (RSUs)/vehicles including the corresponding caching and computing resources to serve a request. To cope with the high complexity caused by the large action space, a mobility-aware reward estimation is proposed.

Hou *at al.* [141] adopted an ε_n-greedy Q-learning model, assisted by vehicle movement prediction, for proactive caching at RSUs. The proposed model determines an optimal caching policy to minimize the expected long-term reward defined as the transmission latency of the requested data. Distributed caching in D2D-assisted mobile networks has been considered in [211], exploring similarity learning in caching for maximizing the user satisfaction. The similarity learning allows to identify the UEs with common content interests before selecting the ones that can act as edge caches. The accuracy of similarity learning depends on the availability of large amounts of training data. Li *et al.* [224] elaborated the D2D edge caching problem in hierarchical wireless networks leveraging on a double deep Q-learning network. A cache replacement strategy is proposed to maximize the hit rate of user-requested content.

Nikham *et al.* [225] dealt with content popularity prediction in a cache-enabled network for augmented reality applications. To address the privacy issue that stems from adopting users' search history, the authors recommended the use of federated learning based on AutoEncoders (AE) that improves the user experience, while preserving privacy. To detect passengers' preferences and select accurate caching decisions, i.e., on MEC nodes or central data-centers, [226] adopted DL based on a CNN model. An MLP framework is also used at data centers to predict the content in specific edge areas. The authors formulated an assembled optimization problem, whose goal is to minimize the content downloading delay.

Table 12 summarizes the investigated literature, highlighting the key advantages and limitations of the AI/ML techniques adopted for edge caching.

6.5. UAV Assisted Services

The combination of UAV capabilities with edge computing can create new services for high mobility users, e.g., connected cars, resolving issues related to connection drop, unavailability of content, or mediocre QoE [227]. Using UAVs for caching may also bring new challenges related to content prediction and privacy. Brik *et al.* [228] adopted FDL for privacy-preserving, which provides an efficient content caching considering mobility patterns. The proposed solution predicts content popularity and determines which content should be stored locally

Table 12: Summary of AI/ML algorithms for edge caching.

Learning Algorithm	**Edge Caching Issue**	**Ref.**	**Benefits & Drawbacks**
TL	Caching policies	[214]	**- Pros:** Knowledge sharing allowing fast and accurate popularity profile estimation **- Cons:** Fine-tuning required to improve model's generalization
Q-learning	Caching policies	[215] [218] [141]	**- Pros:** Adapt to time-varying popularity profile dynamics **- Cons:** Slow convergence
DRL	Caching policies	[219] [222] [223] [224]	**- Pros:** Handle large state spaces and continuous action spaces compared to RL **- Cons:** High computational cost
	Caching locations	[216]	
DAE	Caching policies	[220]	**- Pros:** Unsupervised extraction of relevant users/contents' features **- Cons:** Need to be combined with other techniques to predict content popularity
BRNN	Caching policies	[221]	**- Pros:** Ability to capture context from both past and future data **- Cons:** Increased computational complexity compared to RNNs
FDL	Caching policies	[225]	**- Pros:** Knowledge transfer while preserving data privacy **- Cons:** Prone to privacy leakage
CNN	Caching policies	[226]	**- Pros:** Automated extraction of deep users' features **- Cons:** Overfitting with small datasets and limited generalization ability

in UAV-caches. Practically, this can be seen as a classification problem in which ANN algorithms are used in a federated way to select whether to store content locally or not, identifying also the content itself. The authors in [229] studied the problem of proactive deployment of cache-enabled UAVs for optimized QoE in centralized RANs. The user context information, e.g., visited locations, requested contents, gender, job, device type, is leveraged to foresee the content distribution using an Echo State Network (ESN) model. An ESN is an RNN variant devised for performing non-linear systems prediction [230].

UAVs can also assist rural and hardly accessible regions with coverage limitation providing cost-effective radio solutions. However, user behavior may differ from one zone to another inducing complexity in UAV-to-ground channel selection. To efficiently resolve UAVs deployment in a distributed way, the authors of [228] leveraged on FDL to assure an improved network coverage considering ground users as FDL clients and edge or core clouds as aggregators. The proposed solution attained its objective by observing ground users' behavior and mobility patterns, i.e., positions, directions and speed, to enable an optimal placement of UAVs. The authors used FDL as a hybrid deep CNN algorithm with LSTM dealing with Spatio-Temporal characteristics. Chen *et al.* [231] considered the problem of joint caching and resource allocation for cached-enabled UAVs serving UE in an LTE-Unlicensed (LTE-U) network. The distribution of users' content requests is predicted using a Liquid State Machine (LSM) algorithm, which can effectively deal with time-varying data while reducing the training complexity compared to e.g., CNN, LSTM, ESN. Indeed, the performance results showed that LSM outperforms ESN and Q-learning in terms of prediction accuracy and convergence time, respectively.

Table 13 summarizes the investigated literature, highlighting the key advantages and limitations of the AI/ML techniques adopted for UAV assisted services.

6.6. Network Resource Management

Planning and managing dynamically network resource allocation is a complex problem that involves various parameters especially in a vitalized 5G environments, which consider both networking and cloud resources.

6.6.1. Resource Allocation

Optimal resource allocation is essential to improve 5G network utilization efficiency, while fulfilling the diverse service requirements. Nevertheless, the variety of resources (e.g., spectrum, bandwidth, computing), the service diversity,

Table 13: Summary of AI/ML algorithms for UAV assisted services

Learning Algorithm	**UAV Assisted Services Issue**	**Ref.**	**Benefits & Drawbacks**
FDL	UAVs as caches	[228]	- **Pros:** Knowledge transfer while preserving data privacy - **Cons:** Prone to privacy leakage
	UAVs as base stations	[228]	
ESN	UAVs as caches	[229]	- **Pros:** Simplify training of RNNs; only output layer is trained - **Cons:** Difficulty to interpret the internal state
LSM	UAVs as caches	[231]	- **Pros:** Reduced training complexity and fast convergence compared to other RNNs - **Cons:** Not suitable for capturing long-term dependencies

the massive number of devices, the network dynamics, and the multiple conflicting objectives (e.g., latency, reliability, fairness, spectrum efficiency) make the optimization of resource allocation a combinatorial and non-convex problem [232, 233]. Traditional solutions (e.g., [234, 235]) rely on approximation techniques to simplify this problem into computationally tractable ones, allowing a resolution within a reasonable time, but at the price of sacrificing optimality. AI/ML can optimize the resource allocation owing to the ability of solving complex problems, while achieving the desired performance/complexity balance [34].

A DL based scheme considering DBN to jointly optimize transmission scheduling and time allocation, aiming to maximize the wireless network capacity is elaborated in [236]. Lei *et al.* [237] formulated two time-based resource allocation problems with the goal of minimizing, respectively, the total transmission time and the total energy consumption for content delivery at the network edge. Both CNN and fully-connected DNN are investigated in solving such optimization problems. The devised models aim at determining the best strategy for grouping mobile terminals and share the time resources among the selected groups. Cui *et al.* [238] developed a multi-agent RL-based resource allocation framework for UAV networks allowing each UAV, as a learning agent running a Q-learning algorithm, to independently select the communicating user, power level and subchannel. Zhang *et al.* [239] applied random forest-based ensemble learning to enable a self-adaptive scheduling of transmission time intervals among coexisting slices considering the enhanced Mobile BroadBand (eMBB) and Ultra-Reliable Low-Latency Communication (URLLC).

A radio resource sharing strategy that can reflect momentary network conditions and QoS requirements based on DRL is detailed in [240]. Huang *et al.* [241] proposed a CNN-based cooperative resource allocation scheme to dynamically allocate spectrum and antennas in 5G ultra-dense wireless networks. The use of DRL for radio resource allocation is also considered in [242], but with the aim of guaranteeing fairness between users. Zia *et al.* [243] developed a distributed multi-agent RL-based autonomous spectrum allocation mechanism for multi-tier heterogeneous networks, allowing D2D users to maximize their throughput and spectral efficiency with minimal cellular interference. To deal with the Q-learning scalability issues, the authors in [244] introduced a multi-agent DRL-based spectrum allocation for D2D underlay networks. A deep transfer RL model based on RNNs is devised in [245] for wireless resource allocation focusing on virtual reality applications. The proposed model aims to determine the optimal uplink and downlink resource block allocation strategy that maximizes the users' successful transmission probability. The use of transfer learning leads to an increased learning speed, thanks to its ability of transferring the knowledge from one allocation policy to another. Chen *et al.* [246] explored a joint caching and spectrum allocation scheme for UAV-assisted LTE-U networks using a variant of spiking NNs, namely LSM [247]. The authors in [248] proposed a DRL-based radio resource allocation strategy to enable high QoS provisioning for live ultra high definition video streaming in a highly dynamic UAV-based 5G network.

Li *et al.* [249] highlighted the opportunities of AI to enable intelligent management and orchestration in 5G considering radio resource management and service provisioning. An overview of the recent advances in AI-based network traffic control is provided in [250], embracing traffic classification, network performance prediction and resource management. The notion of feasibility check, i.e., assuring resource availability within a future time window, is significant for deciding whether to allow establishing a network slice [251] [252]. Feasibility check is a process that can be carried out by a slice broker [253], which can assist resource planning, charging and dymanic resource adaptation. A comprehensive overview of ML, i.e., (un-)supervised learning and RL, embracing traffic and network performance prediction as well as network adaptation and resource management is presented in [254]. Bega *et al.* [255] elaborated an AI framework for slice management, detailing both feasibility check and dynamic resource allocation for RAN and 5GC considering dedicated resources and customized NFs. A network slice framework, named DeepCog, that accommodates future time-varying service demands based on DNN is explored in [256]. Unlike conventional traffic forecasting, DeepCog returns a cost-aware capacity forecast, which can be used

Table 14: Summary of AI/ML algorithms for resource allocation.

Learning Algorithm	**Resource Allocation Issue**	**Ref.**	**Benefits & Drawbacks**
DBN	Transmission link and time scheduling	[236]	**- Pros:** Probabilistic and generative modeling **- Cons:** High computational costs
CNN	Transmission link and time scheduling	[237]	**- Pros:** Automated extraction of deep communication channel features **- Cons:** Overfitting with small datasets and limited generalization ability
	Dynamic resource allocation	[241]	
MLP	Transmission link and time scheduling	[237]	**- Pros:** Optimal speed-accuracy trade-off compared to other DL algorithms **- Cons:** Require labeled training datasets
	Dynamic resource allocation	[256] [255]	
Multi-agent Q-learning	Dynamic resource allocation	[238] [243]	**- Pros:** Experience sharing, resulting in fast learning **- Cons:** Curse of dimensionality more sever than in RL
Ensemble learning	Transmission link and time scheduling	[239]	**- Pros:** Enhanced accuracy of transmission time selection **- Cons:** High time complexity
DRL	Dynamic resource allocation	[240] [242] [244] [245] [248] [257]	**- Pros:** Handle dynamic environments with large state spaces and continuous action spaces **- Cons:** Computationally expensive
TL	Dynamic resource allocation	[245]	**- Pros:** Fast learning of allocation policy **- Cons:** Fine tuning required to improve model's generalization
LSM	Dynamic resource allocation	[246]	**- Pros:** Reduced training complexity and fast convergence compared to other RNNs **- Cons:** Not suitable for capturing long-term dependencies

to fine tune the resource allocation maximizing revenues. An intelligent resource allocation framework leveraging DRL for obtaining the optimal computation and communication resources for multiple users in a collaborative MEC environment is proposed in [257].

Table 14 summarizes the investigated literature, highlighting the key advantages and limitations of the AI/ML techniques adopted for resource allocation.

6.6.2. VNF Service Consumption Prediction

Built on softwarization principles, 5G enables flexible virtual NF deployments supporting VNF service provision assurance. Service assurance is a complex process that relies on prediction mechanisms, which can be facilitated by AI/ML based on historic data collection or via learning experience paradigms. Kim *et* *al.* [258] proposed a prediction method for VNFs resource demands exploiting both the flexibility offered by virtualization paradigms and the immense data volume available from monitoring tools to enable accurate prediction patterns. The model is built on an ML technique dubbed Context and Aspect Embedded Attentive Target Dependent LSTM (CAT-LSTM), a type of RNNs. Unlike approaches that enable an individual model for each VNF prediction, CAT-LSTM allows prediction of resource demands for a group of VNFs simultaneously while keeping the accuracy high.

Mijumbi *et al.* [259] proposed a supervised based ML technique, named Graph Neural Network (GNN), to predict and manage VNF resource requirements. Such technique exploits the VNF Forwarding Graph (VNF-FG) topology information to achieve a dynamic allocation of resources for each VNF Component (VNFC) in any given Service Function Chaining (SFC) [260]. In fact, historical local VNFC resource utilization information is used considering also the data collected from the neighbors through two different FNNs. Jmila *et al.* [261] introduced a resource management and prediction scheme for VNFs in virtualized environments emphasizing the inputs features, i.e., processed traffic. The proposed solution leverages SVR, a supervised ML technique, to estimate VNFs' requirements in terms of CPU. A set of experiments using SDN-enabled VNFs and security appliances showed significant improvements in terms of CPU prediction. A joint caching and computing service placement for sensing-data-driven IoT applications is investigated in [262] based on DRL, which can adapt to a heterogeneous system with limited prior knowledge. A policy network based on the encoderdecoder model is constructed to address the issue of varying sizes of joint caching and computing service placement states and actions caused by different numbers of caching functions related to applications.

Table 15 summarizes the investigated literature, highlighting the key advantages and limitations of the AI/ML techniques adopted for VNF service consumption prediction.

6.6.3. NF and Service Relocation Prediction

5G introduces significant improvements in network flexibility with the adoption of micro-services and has enriched the service capabilities by employing a virtualized network architecture. However, the variation of UE types, e.g., UAVs, vehicles, IoT devices, and the corresponding services in conjunction with mobility patterns and device concentration per unit areas have led to various challenges in QoS provision and service continuity. In fact, there is a need to extend the notion of mobility prediction, which is traditionally focusing on user devices towards NF

Table 15: Summary of AI/ML algorithms for VNF service consumption prediction.

Learning Algorithm	**VNF Service Consumption Prediction Issue**	**Ref.**	**Benefits & Drawbacks**
TD-LSTM	VNF resource demands	[258]	**- Pros:** Embedding target-specif context, resulting in high performance compared to LSTM **- Cons:** ineffective for handling multi-step prediction
GNN	VNF resource demands	[259]	**- Pros:** Capture spatial dependencies in graph-structured data **- Cons:** Higher time and space complexity
SVR	VNF resource demands	[261]	**- Pros:** Excellent generalization and high robustness to outliers **- Cons:** Computationally expensive on large datasets
DRL	VNF resource demands	[262]	**- Pros:** Handle dynamic environments with large state spaces and continuous action spaces **- Cons:** Computationally expensive

and service relocation. AI/ML is a pillar technology which can help devising NF and service relocation prediction solutions.

Lange *et al.* [263] designed a mechanism for managing highly dynamic SFC deployments using various ML-based classification methods, i.e., eXtreme Gradient Boosting (XGBoost), Gradient Boosting Machine (GBM), Distributed Random Forest (DRF), Extremely Randomized Trees (XRT), and ANN, to deliver proactive decisions. The proposed approach relies on classification algorithms to create accurate deployment, placement, and relocation decisions for VNFs. Indeed, the authors presented a workflow for generating training data as well as selecting the best monitoring features, utilizing various metrics during the training and development phases. Shaw *et al.* [264] developed a scheme for handling energy losses within clouds and data-centers leveraging RL to create a consolidation agent responsible for relocating Virtual Machines (VMs) across data-centers. A reward shaping technique known as Potential Based Reward Shaping (PBRS) is also employed to cope with the RL limitations. The respective agent uses PBRS to assist and refine the RL decisions by including experts' advice.

To settle the resource consumption issue in cloud computing environments, a Predictive Anti-Correlated VM Placement Algorithm (PACPA) is described in [265]. A three-stage based architecture is introduced where hosts are monitored to predict which VMs should be migrated towards more suitable hosts. In fact, the monitoring phase collects and stores the current and future CPU consumption and then an algorithm named Local Regression Minimum Migration Time (LR-MMT) is used to select potentially overloaded hosts identifying the respective VMs to be migrated. An MLP model forecasts the CPU consumption of VMs in the migration list and then the PACPA algorithm selects VMs that should be co-located together based on the aggregated CPU requirements. The authors of [266] tackled the virtual networks relocation problem in which one or multiple network resources are scaled in/out to create or extend a certain network deployment. An RL agent ensures the desired QoS by dynamically selecting virtual resources, among edge, core and data center networks, and migrating them to assure the performance requirements of critical services [267]. The authors define the link selection between two adjacent nodes as the action space while the reward is based on users' QoS satisfaction levels.

Cao *et al.* [268] proposed an algorithm dubbed "MigRL" for managing cloud environments leveraging live migration operations. The authors used Q-learning to reduce the network overhead of inter-cloud systems while minimizing service costs. Historical access information is used to determine the time and place for migrating services across different available nodes, thus, meeting the rigid requirements begotten by the upcoming network architectures. A CPU load variation prediction method to allow efficient live migration in the data-centers scope is introduced in [269], leveraging the ML shallow algorithm dubbed LR. The proposed approach allows detecting both over-utilized and under-utilized hosts by approximating the short-time future CPU utilization based on the usage history within each host. The proposed algorithm allows to obtain forecasting of next CPU usage, based on the actual CPU load, thus allowing to anticipate migration decisions. Slice mobility, i.e., a slice that moves between service areas, whereby the inter-dependent service and resources shall be migrated to reduce system overhead and minimize communication latency following end-user mobility patterns is elaborated in [270] considering DRL to optimize bandwidth allocations and to adjust the network usage to minimize slice migration overhead.

Table 16 summarizes the investigated literature, highlighting the key advantages and limitations of the AI/ML techniques adopted for NF and service relocation prediction.

6.7. *Fault Management*

Fault detection and resolution is a complex process especially in 5G. In fact, a single event may generate a large amount of alarms, since a physical resource may affect numerous logical NFs. In other words, a lower layer alarm may cause

Table 16: Summary of AI/ML algorithms for NF and service relocation prediction.

Learning Algorithm	**NF/Service Relocation Prediction Issue**	**Ref.**	**Benefits & Drawbacks**
Ensemble learning	Dynamic deployment	[263]	**- Pros:** Enhanced prediction accuracy **- Cons:** High time complexity
MLP	Dynamic deployment	[263]	**- Pros:** Optimal speed-accuracy trade-off compared to other DL algorithms **- Cons:** Require labeled training datasets
	Relocation and consolidation	[265]	
RL	Relocation and consolidation	[264] [266] [268]	**- Pros:** Effective in capturing the environment's dynamics **- Cons:** Cannot handle large state spaces
LR	Relocation and consolidation	[269]	**- Pros:** Simplicity and speed **- Cons:** Unable to capture non-linear patterns
DRL	Dynamic deployment	[270]	**- Pros:** Handle dynamic environments with large state spaces and continuous action spaces **- Cons:** Computationally expensive

multiple alarms in higher layers or across an end-to-end communication service. AI/ML can assist the root cause analysis and alarm correlation to identify the source of a fault.

Kawasaki *et al.* [271] applied ML to analyze the root cause of failures in NFV environments. The authors evaluated the ML-based fault classifier considering: (i) three algorithms, namely MLP, random forest, SVM, (ii) the volume and balance of training data, and (iii) the number of features. An experimental study showed that random forest provides the highest performance even with small amounts of data. Chigurupati *et al.* [272] built a Bayesian Network to model the cause-effect relationship between the degradation parameters, i.e., cause, and failure modes providing an automate root cause analysis. In [273], a method is proposed for root cause analysis by discovering the alarms relation among network nodes, i.e., treating clustered groups of alarms instead of specific events, based on data mining. Shehu *et al.* [274] explored transfer learning and language modeling for fault localization, avoiding the need to build knowledge bases from scratch to account for new services.

Sampaio *et al.* [275] fostered resilience in SDN by using RL to derive the appropriate policies for dealing with network anomalies. The adopted RL approach, namely Q-learning, suffers from scalability because of the state-action space related with the network size, i.e., number of switches. A hybrid model for anomaly detection in cloud environments is presented in [276] leveraging the Grey Wolf

Optimization (GWO) and CNN for relevant feature extraction and anomaly classification. Wang *et al.* [277] introduced a method of alarm pre-processing and correlation analysis. The proposed method combines a time series segmentation and time sliding window to extract the alarm transactions, followed by K-means and back propagation NNs to evaluate the alarm importance quantitatively.

Table 17 summarizes the investigated literature, highlighting the key advantages and limitations of the AI/ML techniques adopted for fault management.

7. AI/ML Business Drivers

This section provides an overview of the emerging AI/ML use cases considering the requirements of mobile network operators, new applications with critical needs and the potential of big data. An overview of a representative set of AI/ML business drivers summarizing the value creation for MNOs is shown in Table 18.

7.1. Network Operations

The value creation of AI/ML for MNOs is evident in cloud-native environments by reducing Capital and Operational Expenditures (CAPEX/OPEX), while enabling efficiently agile operations and assuring service quality. The main use cases related to AI/ML in networking, as highlighted by MNOs in [278] [279], include the following:

- *Network planning* is a complex process that relies on a wide variety of parameters that influence site location, antenna directivity, links and nodes capacity as well as service capabilities in terms of QoS and reliability. AI/ML can assist identifying network infrastructure needs to meet the expected targets, while minimizing investment costs and evaluating the network effectiveness for the expected user services.

- *Network fault diagnosis* focuses on troubleshooting complex network issues based on root cause analysis. AI/ML can provide the means of analyzing efficiently a wide variety of network and user data in order to identify the main cause of a fault considering alarm correlation across different network domains or even to perform a prognosis analysis to prevent it.

- *Network optimization* predicts the traffic demands and service Key Performance Indicators (KPIs) to assist dynamic operations, such as resource allocation or energy-saving, enhancing network agility and utilization.

Table 17: Summary of AI/ML algorithms for fault management.

Learning Algorithm	**Fault Management Issue**	**Ref.**	**Benefits & Drawbacks**
MLP	Root cause analysis	[271]	**- Pros:** Optimal speed-accuracy trade-off compared to other DL algorithms **- Cons:** Require labeled training datasets
Random Forest	Root cause analysis	[271]	**- Pros:** Simplicity and enhanced detection accuracy **- Cons:** ineffective for real-time predictions
SVM	Root cause analysis	[271]	**- Pros:** Effective with small datasets **- Cons:** Feature engineering required
Bayesian Networks	Root cause analysis	[272]	**- Pros:** Probabilistic correlation and causality modling **- Cons:** Exponential increase in inference complexity for large-scale systems
TL	Alarms correlation	[274]	**- Pros:** Fast creation of fault knowledge bases for new services **- Cons:** Fine-tuning required to improve model's generalization
Q-learning	Root cause analysis	[275]	**- Pros:** Adapt to time-varying system's dynamics **- Cons:** Slow convergence time
CNN	Alarms correlation	[276]	**- Pros:** Automated extraction of the traffic's deep features **- Cons:** Overfitting issues and limited generalization ability
GWO	Alarms correlation	[276]	**- Pros:** Simplicity and better convergence compared to other optimization algorithms **- Cons:** Limited performances on highly complex optimization problems
K-means	Alarms correlation	[277]	**- Pros:** Simplicity and speed **- Cons:** Hard clustering which may lead to inaccurate alarms classification

Table 18: Summary of AI/ML business drivers.

Business Driver	**Business Sector**	**Value Creation**
Network Operations	Network planning	Enable effective network infrastructures
	Network fault diagnosis	Troubleshooting complex network issues
	Network optimization	Predict traffic demands and service KPIs
	Security	Identify effectively abnormal behaviours
Service Assurance	Service feasibility check	Assurance resource availability for service duration
	Tactile communications	Assurance of real-time interaction
	Virtual and Mixed Reality	Merging physical and virtual data seamlessly
Vehicular QoS	Switching the LoA	Assist in predicting LoA
	Platooning	Advent automated highway systems
	Warning signals	Enhanced road safety
	Remote data collection	Enhanced driving experience
	Improving traffic control	Managing efficiently city transport
UAV Services	Navigation planning	Allocate flying path conforming safety and privacy
	Machine vision	Discover and identify objects
	Airborne RAN	Identify flying locations for optional coverage
	Airborne services	Optimal proximity services
Customer Data Services	Customer services	Customer analysis for service development
	Fraud detection	Prevent theft, fake profiles and identity cloning
	Monetizing data	Sell data to 3rd parties for knowledge creation

- *Security* where the AI/ML can identify abnormal user/device or network equipment behavior considering the expected user/device communication and mobility patterns or network equipment load in relation with control signaling and user plane traffic patterns.

7.2. Service Quality Assurance

Service assurance leverages the benefits of AI/ML to preserve service performance by analyzing the expected KPIs, users' behavior and traffic demands. Service assurance is critical for delay sensitive applications that rely on immediate response or absolute coordination.

7.2.1. Service Feasibility Check

In the context of network slicing, service requirements are translated into network resources with the AI/ML identifying the quantity and location that fulfils the Service Level Agreement (SLA) with a minimum cost. To this end AI/ML is used to perform a feasibility check, i.e., check if the network resource available is sufficient to assure the desired service quality for the duration of the slice request.

7.2.2. Tactile Communications

Tactile communications [280] involve immediate and highly robust control, touch, and sensing/actuation services to enable a real-time interaction among participating parties in distinct locations. The use of AI/ML can assist towards a predictable networking environment that can assure the desired delay, jitter and degree of reliability. Some representative use case may include:

- *Teleoperation* enables humans to interact with real or virtual objects and perform tasks such as remote touch and control. It allows a wide variety of applications, including remote robotic operations, e.g., in Industry 4.0.

- *Internet of skills* allows transferable human skills to be taught and executed via the Internet, e.g., remote surgery. It facilitates visual or audio sensory experiences equivalent to local and leverages the benefits of virtual/mixed reality for rendering 3D visual representations.

- *Autonomous worksites* enabling automation and remote operations for e.g., harbors, in terms of loading, logistics and traffic control.

7.2.3. Interconnected Virtual and Mixed Reality

Interconnected virtual and mixed reality enable a holographic teleportation and sound in real time from different locations towards a common physical or virtual social existence [281, 282, 283, 284]. Merging physical and virtual data seamlessly requires a coordinated latency to assure synchronization among diverse sites. Common virtual and mixed reality use cases may include:

- *Enhanced communications* allowing conversations, business meetings, and social opportunities for people with special needs.

- *Public events* with virtual presence, allowing, e.g., musicians residing in different physical locations to deliver in real-time a concert together.

- *Gaming* experience that enhances audiovisual based on virtual and augmented reality with a sense of touch through the correlation of computer-generated and real-world sensory information.

7.3. QoS Sustainability in Vehicular Communications

Autonomous cars are expected to form 25% of the vehicles on the road by 2035 [285]. This rapid growth and development is mainly supported by the advances in AI/ML, edge computing and hardware. AI/ML can help maintaining the desired performance and safety considering the expected vehicle mobility. Various AI/ML based use cases related to vehicular communications are documented in [286], including the following:

- *Switching the Level-of-Automation (LoA)*, according to the Society of Automotive Engineers (SAE), there are six LoA with gradually different features starting from manual to full automation [287]. Each LoA requires a different network performance, with the higher LoA (i.e., full automation) needing more stringent performance. To assure safety while driving, AI/ML can assist in predicting the expected network performance and trigger the vehicular application to switch towards a lower LoA once the expected network requirements cannot meet the desired demands and vice versa.

- *Platooning* is a method for driving a group of closely distant vehicles, which are linked electronically, allowing a synchronous behavior; i.e., accelerate or brake simultaneously. Typically, platooning relies on automated highway systems; i.e., with marks that act as sensors to assist vehicles to measure the speed and direction, as well as on a computer control either on-board the last vehicle or within the network. Such computer control collects traffic and road data (e.g., speed control) and leverages the benefits of AI/ML.

- *Warning signals* may vary from roadworks, traffic jam, vehicle approaching, emergency brake and collision risks. Some of these warnings are based on analysis of local vehicular data via on-board sensors and AI/ML, while others rely on additional data provided via the network edge or road infrastructure.

- *Remote data collection* from other vehicles or road infrastructure including sensor data sharing, e.g., speed or direction, or real-time video, e.g., to see-through for pass maneuver. An AI/ML can coordinate such data acquisition considering the driving behavior and vehicle position.

- *Improving traffic control* may assist the city transport system by using AI/ML to predict and avoid congestion.

7.4. Assurance Control in UAV

UAVs support a wide range of use cases including public safety, inspection, emergency situations, healthcare, goods delivery and agriculture. Leveraging the benefits of mobility, adaptive altitude and flexible network connectivity, UAVs can offer on-demand communication and computing capabilities, real-time monitoring and control. Although UAVs are navigated autonomously, a real-time control at any time, which relies on accurate location tracking and low latency high reliable communications is a prerequisite. The use of AI/ML should consider such needs alongside other limitations related to computational and energy resources to enable automation in:

- *Navigation planning* allocates the UAV path, speed and altitude by correlating data from various sources: (i) privacy and law that indicate forbidden flying regions, (ii) weather forecasts, terrain and obstacle data that provides performance insights for UAV connectivity, and (iii) air operator data regarding airway routes and other UAV navigation plans in the region. AI/ML can correlate such data considering also the expected network and cloud resource availability for assuring real-time control and computing capabilities along the allocated airway route.

- *Machine vision and image recognition* enables UAVs to identify and label objects. Thanks to AI/ML, UAVs are not only displaying camera captures but can perceive and understand the surrounding environment assisting in terrain recognition, e.g., for emergency situations.

- *Airborne RAN* on-board UAVs can accommodate connectivity for special events, or temporary coverage faults in a cost efficient manner. AI/ML should assist UAVs to determine the optimal flying locations or parking positions, considering the expected traffic demands, user mobility and the connectivity with mobile network.

- *Airborne services* on-board UAVs, including content caching, computing and other proximity services can assist highly mobile users, e.g., vehicles. AI/ML can help predicting the content and service popularity with respect to particular locations.

7.5. Customer Data Services

With the rise of AI/ML, which offers the capability to analyze usage insights in communication networks, customer data has become a significant asset for MNOs driving a number of use cases as detailed in [288], including the following:

- *Business decisions* may rely on AI/ML to extract insights from customers in order to scale and develop further services, understand customer segmentation and customer lifetime value. These insights can help forming customers profiles for pricing plans and creating new business services.

- *Fraud detection* including illegal access, theft, fake profiles and identity cloning. AI/ML can detect such frauds and provide alerts.
- *Monetizing data* allows a MNO to share collected data in order to enable third parties to perform AI/ML model training or transfer learning, but requires to impose restrictions, e.g., per partner or geographical area.

8. Lessons Learned & Open Challenges

AI/ML is revolutionary technology that introduces intelligence and automation in every networking layer and within the entire life-cycle of a communication service enabling an AI native environment. It creates new opportunities for service provisioning and brings new technical and business challenges. This section discusses such AI/ML and network architecture challenges and highlights some research directions.

8.1. Native AI

Typically, AI/ML algorithms devise the analytics model, which is packaged together with other capabilities including storage, consumer registration, service creation, data monitoring/formatting and analytics distribution, comprising a single analytics function. Multiple analytics functions should be established in a mobile network architecture for serving different geographical areas. This raises several questions on how analytics functions shall be arranged and inter-work within the network architecture in order to assure scalability on collecting and exchanging data. To address this issue the monolithic AI/ML block should be split into multiple pieces with a smaller functional scope, which can be flexibly arranged to reflect particular needs [289] using an AI/ML orchestrator that manages, creates, composes and relocates analytics services.

Analytics functions follow the consumer-produce paradigm, which is inefficient for conveying high volumes of data, towards multiple consumers. The main issue is that the underlying service-based architecture is not designed to carry high volumes of data and cannot operate to support real-time data collection and distribution. Alternative mechanisms are needed such as streaming data, e.g., Amazon Kinesis [3], which can handle dynamic data generated on a continual basis. The emerging mobile networks shall consider to adopt a separate communication medium to support efficient real-time data collection and distribution. Besides the

[3] https://docs.aws.amazon.com/streams/latest/dev/introduction.html

need for a new communication paradigm, the concept of native AI [290], i.e., embedding AI/ML in network elements and equipment, or integrating it in communication protocols, can further advance the AI/ML operations integrating AI/ML with the emerging mobile networks.

8.2. Life-cycle of AI/ML Model

The life-cycle of an AI/ML model consists of four phases including the planning, preparation, operation and decommissioning. The planning phase involves the model selection and configuration, which can be performed by the mobile network operator or a 3rd party provided that the appropriate exposure interfaces are in place to support: (i) model selection and parameter configuration or (ii) 3rd party model injection either via the means of software uploading or meta-language description. The support of the aforementioned processes require extensions on the corresponding exposure interfaces.

The operational phase should then ensure a consistent performance of the AI/ML model, which requires regular checks that may trigger potential model updates, modifications or even the launch of a new model. Such a process requires the consumer or another entity to provide feedback regarding the performance of the AI/ML model, indicating the inaccuracy degree, problematic output range or inefficiencies of the input data. In addition, there is a need to indicate, which type of a model is needed by defining model profiles that can be used to serve certain goals. Once an AI/ML model is characterized as problematic it should be disabled or certain output should be filtered out depending on the type and the degree of the problem. An AI/ML orchestration entity should be defined to take care of AI/ML model health by analyzing the feedback and propose a resolution, e.g., re-training and validation, including also potential alternations of the data sources and/or enriching the type or modifying the time schedule details of the collected data.

8.3. AI/ML & Architectural Enhancements

Network intelligence stretches AI/ML across the 5GC, RAN, management and application plane, with the need to exchange knowledge, i.e., analytics, as well as raw data and KPIs. To achieve this cross system AI [291] in an efficient way, there is a need to introduce unified AI/ML mechanisms and hosting environments that merge various types of analytics providing interoperability, while minimizing data exchange and the response time towards the consumer. Such hosting environment can be offered in the form of a unified platform across the core, management and

RAN, allowing a customized assembly of various analytics to assist particular network services or applications. The RAN requires a new data exposure perspective, which is currently handled via the management plane. Architecture enhancements should be investigated to reshape the RAN into a service-based architecture where entities like RIC can share intelligence and knowledge directly with 5GC and application plane, enabling innovation in service provision and QoE. A step further would be the introduction of a new AI plane [292], with common data collection and distribution mechanisms across 5GC, RAN and OAM, and a logical data lake to assist the need for holding historical data and providing a unified model training and transfer learning across different AI/ML entities in an effective manner.

In addition, there is a need to introduce mechanisms enabling a prompt interaction between the application layer and analytics, allowing the network layer to benefit from application data. In fact, application data can provide significant information for the network, e.g., a closed-circuit television can offer user mobility information or a factory camera inspection system can alter the service provision and hence the required network resources for completing a product. Data translators or new KPIs to capture the application intelligence should be investigated, which map application data into networking "language". Obtaining user context information from the application related to the user profile, location, usage, time, related events, etc., can further enhance the network performance and service experience. Finally, there is a need for further architecture enhancements, i.e., APIs, that would enable interaction with digital twins [101]. Digital twins support network emulation, i.e., sandbox, and can serve for real-time testing of AI/ML decision that impact the network configuration and can provide training of AI/ML models before being applied in the network in a real situation or can assist training functions with missing data.

8.4. Analytics & Charging

Similarly to other network services, analytics shall introduce charging events depending on the type of requested service and the number of provided reports as specified in [293]. Such charging capabilities needs to take into account more complex scenarios, where an analytics report requires raw data or other analytics, e.g., service experience requires RAN analytics and QoE measurements from 5GC and application. Charging can alternatively be coupled with service assurance or primal user contracts or can be a part of specific applications, e.g., autonomous driving. Charging records can also provide useful information related to user context for deriving analytics. In fact, charging records can provide rich information related to service usage for a specific user or in particular locations and under certain circumstances. Mobile network operators can use such analytics for figuring out the usage and popularity of certain services in order to optimize new service development, edge computing and network resources. Such information is also useful for evolving charging models taking into account emerging service usage. Further work is required towards this direction for anlyzing charging records and correlating service usage data.

8.5. Privacy-preserving in AI/ML

User data forms the foundation for realizing the paradigm shift from network-centric to user-centric operations in 5G. However, the reliance on AI/ML to automatically process and derive insights for customizing the user experience raises privacy concerns. In fact, the use of AI/ML increases the risks of revealing user's sensitive information, such as identity, position, personal interests and activities. Thus, privacy-preserving AI/ML techniques are paramount to reap the benefits of automation in empowering user-centric networks without infringing the user's privacy, with popular approaches including differential privacy, homomorphic encryption and decentralized learning.

Differential privacy withholds data about individuals by adding a controlled amount of noise during the model training, which prevents an adversary to infer whether a specific individual input belongs to the training dataset or not. Homomorphic encryption guarantees privacy by enabling training over encrypted data. The decentralized learning (e.g., federated learning) maintains the privacy by enabling training on the user's private data without requiring direct access to such data. Despite the merit of the aforementioned approaches, their application exhibits accuracy, performance and privacy breaching challenges [157]. Indeed, the use of differential privacy may negatively impact the accuracy due to the introduced noise. Homomorphic encryption induces significant computational complexity. Although federated learning protects the privacy by sharing only the model parameters instead of data, the disclosure of private data is still possible using; for instance, gradient leakage [294] or membership inference [295] attacks. Therefore, how to preserve privacy without trading-off performance and accuracy is still an open question.

8.6. Cost Efficiency and AI/ML

The promise of AI/ML in 5G and beyond is not without an increased cost. In fact, the noticeable performances exhibited by emerging AI/ML techniques, such as DL, come at the cost of high computation and operational complexity. The computation complexity of AI/ML techniques is proportional to the time required to process a single sample, the data set size, the model size (i.e., model's parameters), and the number of experiments needed to tune the model's hyperparameters (e.g., number of layers, the number of neurons in each layer, the batch size, the learning rate) [6]. To empower cost efficient AI/ML solutions, new strategies are required to develop techniques that rely on reduced amount of data and fewer parameters to reach the desired accuracy level. For instance, the authors in [296] discussed different compression approaches that can be leveraged to reduce the size of a DRL model. Furthermore, it is essential to improve the computation efficiency of hyperparameter tuning; an expensive process requiring several training and testing trials in order to find the optimal set of hyperparameters yielding the higher accuracy. Novel hyperparameter optimization methods that can support parrallelization are desirable [297].

8.7. AI/ML in Roaming Scenarios

The use of AI/ML is considered within a single MNO. However, there are several cases where a user may move out of such an MNO coverage area. In this case the user may roam to a mobile network that belongs to a different MNO, provided that there is a roaming agreement in place. In such a case there is a need to consider how ongoing AI/ML operations shall be handled or even to share user analytics. To accomplish this there are several challenges related to data sharing, since operators need to respect user data privacy and at the same time shall not reveal network internal information, e.g., related to network load and utilization. Certainly, data anonymity is needed related to user data, but also mechanisms to be able to decide when to provide this data and towards which AI/ML entity in the other MNO network. In terms of sharing network data between MNO, instead of sharing sensitive data it may be preferable to used FL in where different MNO can use their local data to train an AI/ML instead of sharing, but still there are challenges on exchanging the AI/ML model. One solution can be to share a model only among AI/ML entities that reside on different MNOs but belong to the same vendor, in order to avoid revealing model information to competitors, but further work is needed on the security process.

9. Conclusion

This survey provides a comprehensive insight of AI/ML in emerging mobile communications considering the business perspective, the main concepts and fundamental algorithms as well as their applicability into the control and management plane. It sheds light on how key technologies related to the network evolution towards service-based architectures can assist the adoption of AI/ML across different domains, including the RAN, 5GC, and OAM, emphasizing the notion of AI/ML service request and reporting as well as data collection and distribution. The adoption of AI/ML in 3GPP networks is elaborated considering the management plane with SON and MDA as well as the control plane focusing on NWDAF. Furthermore, it overviews how the main AI/ML algorithms are used in networking, considering a user-centric and a network-centric insight while pointing out their adoption in the control and management plane, i.e., NWDAF and MDA, respectively. Finally, other standardization and open source efforts are reviewed before documenting the lessons learned and identifying the further challenges that would shape AI/ML applicability for mobile systems beyond 5G.

Acknowledgments

This research work is partially supported by the European Union's Horizon 2020 Research and Innovation Program through the aerOS project under Grant No. 101069732; the Business Finland 6Bridge 6Core project under Grant No. 8410/31/2022; the Research Council of Finland (former Academy of Finland) under IDEA-MILL project (Grant No. 352428) and 6G Flagship program (Grant No. 346208); and the European Union's HE research and innovation programe HORIZON-JUSNS-2022 under the RIGOUROUS project (Grant No. 101095933).

[1] T. Taleb *et al.*, “6G System Architecture A Service of Services Vision,” *ITU Journal on Future and Evolving Technologies*, vol. 3, no. 3, pp. 710 – 743, Dec. 2022.

[2] K. Samdanis and T. Taleb, “The Road beyond 5G: A Vision and Insight of the Key Technologies,” *IEEE Network*, vol. 34, no. 2, pp. 135–141, Mar/Apr. 2020.

[3] ITU-T, “Network 2030: A Blueprint of Technology, Applications and Market Drivers Towards the Year 2030 and Beyond,” *Study Group 13, FG-NET-2030*, 2019.

[4] O. Hireche, C. Benzaid, and T. Taleb, “Deep Data Plane Programming and AI for Zero Trust Self-Driven Networking in Beyond 5G,” *Computer Networks*, vol. 203, Feb. 2022.

[5] K. Letaief *et al.*, “The Roadmap to 6G: AI Empowered Wireless Networks,” *IEEE Communications Magazine*, vol. 57, no. 9, pp. 84 – 90, Aug. 2019.

[6] C. Benzaid and T. Taleb, “AI-driven Zero Touch Network and Service Management in 5G and Beyond: Challenges and Research Directions,” *IEEE Network Magazine*, vol. 34, no. 2, pp. 186 – 194, Mar/Apr. 2020.

[7] GSMA, “The Mobile Economy,” 2023.

[8] G. Frisiani, J. Jubas, T. Lajous, and P. Nattermann, “A future for Mobile Operators: The Keys to Successful Reinvention,” *McKinsey & Company, Telecommunications*, Feb. 2017.

[9] R. Addad *et al.*, “Network Slice Mobility in Next Generation Mobile Systems: Challenges and Potential Solutions,” *IEEE Network*, vol. 34, no. 1, pp. 84–93, Jan. 2020.

[10] J. Wang *et al.*, “Thirty Years of Machine Learning: The Road to Pareto-Optimal Next-Generation Wireless Networks,” *IEEE COMST*, vol. 22, no. 3, pp. 1472–1514, 2019.

[11] Q. Mao, F. Hu, and Q. Hao, “Deep Learning for Intelligent Wireless Networks: A Comprehensive Survey,” *IEEE COMST*, vol. 20, no. 4, pp. 2595 – 2621, 2018.

[12] J. Kaur, M. Khan, M. Iftikhar, M. Imran, and Q. U. Haq, “Machine Learning Techniques for 5G and Beyond,” in *IEEE Access*, vol. 9, Jun. 2021, pp. 23 472 – 23 488.

[13] T. Nguyen and G. Armitage, “A Survey of Techniques for Internet Traffic Classification using Machine Learning,” *IEEE COMST*, vol. 10, no. 4, pp. 56 – 76, 2008.

[14] S. Rezaei and X. Liu, “Deep Learning for Encrypted Traffic Classification: An Overview,” *IEEE Communications Magazine*, vol. 57, no. 5, pp. 76 – 81, May 2019.

[15] F. Pacheco *et al.*, “Towards the Deployment of Machine Learning Solutions in Network Traffic Classofocation: A Systematic Survey,” *IEEE COMST*, vol. 21, no. 2, pp. 1988 – 2014, 2019.

[16] H. Zhang and L. Dai, “Mobility Prediction: A Survey on State-of-the-Art Schemes and Future Applications,” *IEEE Access*, vol. 7, pp. 802 – 822, Dec. 2018.

[17] Z. M. Fadlullah *et al.*, “State-of-the-Art Deep Learning: Evolving Machine Intelligence Toward Tomorrow’s Intelligent Network Traffic Control Systems,” *IEEE COMST*, vol. 19, no. 4, pp. 2432 – 2455, 2017.

[18] M. Usama *et al.*, “Unsupervised Machine Learning for Networking: Techniques, Applications and Research Challenges,” *IEEE Access*, vol. 7, pp. 65 579 – 65 615, May 2019.

[19] R. Boutaba *et al.*, “A Comprehensive Survey on Machine Learning for Networking: Evolution, Applications and Research Opportunities,” *Journal of Internet Services and Applications*, vol. 9, no. 16, pp. 1 – 99, June 2018.

[20] Y. Sun *et al.*, “Applications of Machine learning in Wireless Networks: Key Techniques and Open Issues,” *IEEE COMST*, vol. 21, no. 4, pp. 3072–3108, 2019.

[21] C. Zhang, P. Patras, and H. Haddadi, “Deep Learning in Mobile and Wireless Networking: A Survey,” *IEEE COMST*, vol. 21, no. 3, pp. 2224–2287, 2019.

[22] P. Klaine, M. Imran, O. Onireti, and R. Souza, "A Survey of Machine Learning Techniques applied to Self-Organizing Cellular Networks," *IEEE COMST*, vol. 19, no. 4, pp. 2392 – 2431, 2017.

[23] X. Wang, X. Li, and V. Leung, "Artificial Intelligence-based Techniques for Emerging Heterogeneous Network: State of the Arts, Opportunities and Challenges," *IEEE Access*, vol. 3, pp. 1379 – 1391, Aug. 2015.

[24] O. Nassef, W. Sun, H. Purmehdi, M. Tatipamul, and T. Mahmoodi, "A survey: Distributed Machine Learning for 5G and beyond," in *Computer Networks*, vol. 207, Apr. 2022.

[25] E. Hodo *et al.*, "Shallow and Deep Networks Intrusion Detection System: A Taxonomy and Survey," *CoRR*, vol. abs/1701.02145, 2017.

[26] Y. Xin *et al.*, "Machine Learning and Deep Learning Methods for Cybersecurity," *IEEE Access*, vol. 6, pp. 35 365 – 35 381, May 2018.

[27] N. Moustafa, J. Hu, and J. Slay, "A Holistic Review of Network Anomaly Detection Systems: A Comprehensive Survey," *Journal of Network and Computer Applications*, vol. 128, pp. 33 – 55, Feb 2019.

[28] K. da Costa *et al.*, "Internet of Things: A Survey on Machine Learning-based Intrusion Detection Approaches," *Computer Networks*, vol. 151, pp. 147 – 157, Mar. 2019.

[29] P. Mishra, E. Pilli, V. Varadharajan, and U. Tupakula, "Intrusion Detection Techniques in Cloud Environment: A Survey," *Journal of Network and Computer Applications*, vol. 77, pp. 18 – 47, Jan. 2017.

[30] N. Sultaba, N. Chilamkurti, W. Peng, and R. Alhadad, "Survey on SDN-based Network Intrusion Detection System using Machine Learning Approaches," *Peer-to-Peer Networking and Applications*, vol. 12, no. 2, pp. 493 – 501, Mar. 2019.

[31] P. Mishra, V. Varadharajan, U. Tupakula, and E. Pilli, "A Detailed Investigation and Analysis of Using Machine Learning Techniques for Intrusion Detection," *IEEE COMST*, vol. 21, no. 1, pp. 686 – 728, 2019.

[32] T. Nguyen and V. J. Reddi, "Deep Reinforcement Learning for Cyber Security," *CoRR*, vol. abs/1906.05799, June 2019.

[33] M. Husák, J. Komárková, E. Bou-Harb, and P. Čeleda, "Survey of Attack Projection, Prediction, and Forecasting in Cyber Security," *IEEE COMST*, vol. 21, no. 1, pp. 640 – 660, 2019.

[34] M. Chen *et al.*, "Artificial Neural Networks-based Machine Learning for Wireless Networks: A Tutorial," *IEEE COMST*, vol. 21, no. 4, pp. 3039 – 3071, 2019.

[35] C. Jiang *et al.*, "Machine Learning Paradigms for Next-Generation Wireless Networks," *IEEE Wireless Communications*, vol. 24, no. 2, pp. 98 – 105, Apr. 2017.

[36] J. Jagannath *et al.*, "Machine Learning for Wireless Communications in Internet of Things: A Comprehensive Survey," *Ad Hoc Networks*, vol. 93, p. 101913, June 2019.

[37] N. Luong *et al.*, "Applications of Deep Reinforcement Learning in Communications and Networking: A Survey," *IEEE COMST*, vol. 21, no. 4, pp. 3133–3174, 2019.

[38] N. Ahad, J. Qadir, and N. Ahsan, "Neural Networks in Wireless Networks: Techniques, Applications and Guidelines," *Journal of Network and Computer Applications*, vol. 68, pp. 1 – 27, June 2016.

[39] A. Zappone, M. D. Renzo, and M. Debbah, "Wireless Networks Design in the Era of Deep Learning: Model-Based, AI-Based, or Both?" *IEEE TCOM*, vol. 67, no. 10, pp. 7331–7376, Oct. 2019.

[40] G. P. T. Board, "AI and ML Enablers for Beyond 5G Networks," in *Version 1.0*, May 2021.

[41] 3GPP TS 32.500, "Telecommunication management; Self-Organizing Networks (SON); Concepts and requirements," *Rel.8*, Apr. 2022.

[42] 3GPP TS 28.535, "Management and orchestration; Management services for communication service assurance; Requirements," *Rel.16*, Jul. 2020.

[43] ETSI GS ZSM 002, "Zero-touch network and Service Management (ZSM); Reference Architecture," *V1.1.1*, Aug. 2019.

[44] C. Benzaid, T. Taleb, and J. Song, "AI-based Autonomic & Scalable Security Management Architecture for Secure Network Slicing in B5G," *IEEE Network Magazine*, vol. 36, no. 6, pp. 165 – 174, Nov./Dec. 2022.

[45] 3GPP TR 28.810, "Study on Concepts, Requirements and Solutions for Levels of Autonomous Network," *Rel.16*, Sep. 2020.

[46] ETSI GS ZSM-009-1, "Zero-touch network and Service Management (ZSM); Closed-loop automation; Enablers," *V0.9.1*, Sep. 2020.

[47] ETSI GS ZSM-009-2, "Zero-touch network and Service Management (ZSM); Closed-loop automation; Solutions," *V0.3.1*, Jul. 2020.

[48] ETSI GS ZSM-009-3, "Zero-touch network and Service Management (ZSM); Closed-loop automation; Advanced topics," *V0.0.1*, Sep. 2019.

[49] ITU-T FG-ML5G, "Unified Architecture for Machine Learning in 5G and Future Networks," Mar. 2019.

[50] 3GPP TS 23.501, "System Architecture for the 5G System (5GS)," *Rel.15*, Apr. 2023.

[51] R. Fielding and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content," *IETF RFC 7231*, Jun. 2014.

[52] 3GPP TS 28.533, "Management and orchestration; Architecture framework," *Rel.17*, Mar. 2022.

[53] 3GPP TS 23.502, "Procedures for the 5G System (5GS)," *Rel.15*, Apr. 2023.

[54] 3GPP TR 23.791, "Study of enablers for Network Automation for 5G," *Rel.16*, Jun. 2019.

[55] 3GPP TS 23.288, "Architecture enhancements for 5G System (5GS) to support Network Data Analytics Services," *Rel.16*, Jul. 2020.

[56] 3GPP TS 28.552, "Management and Orchestration; 5G Performance Measurements," *Rel.15*, Mar. 2023.

[57] 3GPP TS 28.554, "Management and orchestration; 5G end to end Key Performance Indicators (KPI)," *Rel.15*, Mar. 2023.

[58] 3GPP TR 23.700-91, "Study on enablers for network automation for the 5G System (5GS); Phase 2," *Rel.17*, Jul. 2020.

[59] 3GPP TR 23.700-80, "Study on 5G System Support for AI/ML-based Services," *Rel.18*, Dec. 2022.

[60] K. Samdanis, A. N. Abbou, J. Song, and T. Taleb, "AI/ML Service Enablers Model Maintenance for Beyond 5G Networks," *IEEE Network Magazine*, Jan. 2023.

[61] 3GPP TR 28.809, "Study on enhancement of Management Data Analytics," *Rel.16*, Jun. 2020.

[62] 3GPP TS 28.104, "Management and orchestration; Management Data Analytics," *Rel.17*, Apr. 2022.

[63] 3GPP TS 32.423, "Telecommunication management; Subscriber and equipment trace; Trace data definition and management," *Rel.6*, Dec. 2021.

[64] 3GPP TS 28.532, "Management and orchestration; Generic management services," *Rel.17*, Mar. 2022.

[65] 3GPP TS 23.436, "Functional architecture and information flows for Application Data Analytics Enablement Service," *Rel.18*, Apr. 2023.

[66] E. Pateromichelakis, D. Dimopoulos, and A. Salkintzis, "NetAPPs Enabling Application-Layer Analytics for Vertical IOT Industry," *IEEE Internet of Things Magazine*, vol. 5, no. 4, pp. 130 – 135, Dec. 2022.

[67] T. Taleb *et al.*, "On Multi-access Edge Computing: A Survey of the emerging 5G Network Edge Cloud Architecture and Orchestration," *IEEE COMST*, vol. 19, no. 3, pp. 1657 – 1681, 2017.

[68] Open Network Automation Platform, "ONAP Architecture Overview," Jun. 2018.

[69] Open Network Automation Platform, online documentation, "https://docs.onap.org/en/latest/," 2020.

[70] Open Platform for NFV, online documentation, "https://www.opnfv.org," 2020.

[71] ORAN, “O-RAN Architecture Description,” *v01.00.00*, Feb. 2020.

[72] ——, “Non-RT RIC & A1 Interface: Use Cases and Requirements,” *v02.00*, Apr. 2020.

[73] ——, “AI/ML Workflow Description and Requirements,” *v01.01*, Apr. 2020.

[74] ——, “Near-Real-time RAN Intelligent Controller Near-RT RIC Architecture,” *v01.00*, Apr. 2020.

[75] ETSI GS ENI 002, “Experiential Networked Intelligence (ENI); ENI Requirements,” Sep. 2019.

[76] W. Hapsari *et al.*, “Minimization of Drive Tests Solution in 3GPP,” *IEEE Communications Magazine*, vol. 50, no. 6, pp. 28 – 36, Jun. 2012.

[77] A. Xiaoguang and L. Xiaofan, “Packet Capture and Protocol Analysis Based on Winpcap,” in *IEEE ICRIS*, Zhangjiajie, Aug 2016.

[78] S. Alias, S. Manickam, and M. Kadhum, “A Study on Packet Capture Mechanisms in Real Time Network Traffic,” in *IEEE Int. Conf. on Advanced Computer Science Applications and Technologies*, Kuching, Dec. 2013.

[79] R. Zou, T. Xu, and H. Hou, “An Enhanced Netflow Data Collection System,” in *IEEE Int. Conf. on Instrumentation, Measurement, Computer, Communication and Control*, Harbin, Dec. 2012.

[80] W. Li and X. Yu, “An Online Flow-Level Packet Classification Method on Multi-core Network Processor,” in *IEEE CIS*, Shenzhen, Dec 2015.

[81] A. Bhole, B. Adinarayana, and S. Shenoy, “Log analytics on Cloud Using Pattern Recognition a Practical Perspective to Cloud Based Approach,” in *IEEE ICGCIoT*, Noida, Oct 2015.

[82] 3GPP TR 28.842, “Management and orchestration; Study on data management phase 2,” *Rel.18*, Apr. 2023.

[83] H. Lee, S. Moon, and T. Cho, “Adaptive False Data Filtering Method for Sensor Networks Based on Fuzzy Logic and Commutative Cipher,” in *IEEE Int. Conf. on Computer and Electrical Engineering*, Phuket, Dec 2008.

[84] J. Zhang, D. Fang, and L. LIiu, "Intelligent Content Filtering Model for Network Security Audit System," in *Int. Workshop on Knowledge Discovery and Data Mining*, Moscow, Jan 2009.

[85] J. Pfender and W. Seah, "Leveraging Localisation Techniques for In-Network Duplicate Event Data Detection and Filtering," in *IEEE LCN*, Singapore, Oct 2017.

[86] ETSI, "Improved operator experience through Experiential Networked Intelligence (ENI)," *White Paper No. 22*, Oct. 2017.

[87] ETSI GS ENI 005, "Experiential Networked Intelligence (ENI); System Architecture," *V1.1.1*, Sep. 2019.

[88] ETSI GS ENI 001, "Experiential Networked Intelligence (ENI); ENI use cases," *V2.1.1*, Sep. 2019.

[89] A. Clemm, L. Ciavaglia, L. Granville, and J. Tantsura, "Intent-Based Networking - Concepts and Definitions," *IETF Draft*, Mar. 2020.

[90] ETSI GR ZSM 005, "Zero-touch network and Service Management (ZSM); Means of Automation," *V1.1.1*, May 2020.

[91] A. Singh, N. Thakur, and A. Sharma, "A review of supervised machine learning algorithms," in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, New Delhi, India, Mar. 2016.

[92] Z. Ghahramani, *Unsupervised Learning*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004.

[93] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning (Adaptive Computation and Machine Learning)*. MIT Press, 2006.

[94] K. Arulkumaran, M. Deisenroth, M. Brundage, and A. Bharath, "Deep Reinforcement Learning: A Brief Survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, Nov. 2017.

[95] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. MIT Press, Cambridge, Massachusetts., 2018.

[96] S. Pan and Q. Yang, “A Survey on Transfer Learning,” *IEEE TKDE*, vol. 22, no. 10, pp. 1345 – 1359, Oct. 2010.

[97] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, “A survey on ensemble learning,” *Frontiers of Computer Science*, vol. 14, pp. 241–258, 2020.

[98] S. C. Hoi, D. Sahoo, J. Lu, and P. Zhao, “Online learning: A comprehensive survey,” *Neurocomputing*, vol. 459, pp. 249–289, 2021.

[99] O. Wahab, A. Mourad, H. Otrok, and T. Taleb, “Federated Machine Learning: Survey, Multi-Level Classification, Desirable Criteria and Future Directions in Communication and Networking Systems,” *IEEE COMST*, vol. 23, no. 2, pp. 1342 – 1397, Feb. 2021.

[100] Y. Xu, Y. Zhou, P. Sekula, and L. Ding, “Machine learning in construction: From shallow to deep learning,” *Developments in the built environment*, vol. 6, p. 100045, 2021.

[101] Ericsson, “Digital twins: what are they and how are they enabling future networks?” in *White Paper*, Mar. 2022.

[102] D. Maulud and A. M. Abdulazeez, “A review on linear regression comprehensive in machine learning,” *Journal of Applied Science and Technology Trends*, vol. 1, no. 4, pp. 140–147, 2020.

[103] P. Cunningham and S. J. Delany, “k-nearest neighbour classifiers-a tutorial,” *ACM computing surveys (CSUR)*, vol. 54, no. 6, pp. 1–25, 2021.

[104] C. Cortes and V. Vapnik, “Support-Vector Networks,” *Machine Learning*, vol. 20, no. 3, pp. 273 – 297, Sep. 1995.

[105] S. B. Kotsiantis, “Decision trees: a recent overview,” *Artificial Intelligence Review*, vol. 39, pp. 261–283, 2013.

[106] Y. Liu, Y. Wang, and J. Zhang, “New machine learning algorithm: Random forest,” in *Information Computing and Applications: Third International Conference, ICICA 2012*, Chengde, China, Sep. 2012.

[107] D.-C. Feng, Z.-T. Liu, X.-D. Wang, Y. Chen, J.-Q. Chang, D.-F. Wei, and Z.-M. Jiang, “Machine learning-based compressive strength prediction for concrete: An adaptive boosting approach,” *Construction and Building Materials*, vol. 230, 2020.

[108] D. D. Lewis, “Naive (bayes) at forty: The independence assumption in information retrieval,” in *European conference on machine learning*, Berlin, Heidelberg, 1998.

[109] F. Yi and I. Moon, “Extended K-Means Algorithm,” in *IEEE IHMSC*, Hangzhou, Aug 2013.

[110] K. Greff, S. Van Steenkiste, and J. Schmidhuber, “Neural expectation maximization,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[111] F. Kherif and A. Latypova, “Principal component analysis,” in *Machine Learning*. Elsevier, 2020, pp. 209–225.

[112] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986. [Online]. Available: https://doi.org/10.1038/323533a0

[113] M. W. Gardner and S. Dorling, “Artificial neural networks (the multilayer perceptron)a review of applications in the atmospheric sciences,” *Atmospheric environment*, vol. 32, no. 14-15, pp. 2627–2636, 1998.

[114] H. Salehinejad, S. Sankar, J. Barfett, E. Colak, and S. Valaee, “Recent advances in recurrent neural networks,” *arXiv preprint arXiv:1801.01078*, 2017.

[115] H. Palangi, L. Deng, Y. Shen, J. Gao, X. He, J. Chen, X. Song, and R. Ward, “Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 4, pp. 694–707, 2016.

[116] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *arXiv preprint arXiv:1412.3555*, 2014.

[117] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in *2017 International Conference on Engineering and Technology (ICET)*, 2017.

[118] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE signal processing magazine*, vol. 35, no. 1, pp. 53–65, 2018.

[119] G. E. Hinton and R. R. Salakhutdinov, "A better way to pretrain deep boltzmann machines," *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[120] A. Mohamed, G. Dahl, G. Hinton *et al.*, "Deep belief networks for phone recognition," in *Nips workshop on deep learning for speech recognition and related applications*, vol. 1, no. 9. Vancouver, Canada, 2009.

[121] H. Shao, H. Jiang, Y. Lin, and X. Li, "A novel method for intelligent fault diagnosis of rolling bearings using ensemble deep auto-encoders," *Mechanical Systems and Signal Processing*, vol. 102, pp. 278–297, 2018.

[122] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[123] H. v. Hasselt, A. Guez, and D. Silver, "Deep Reinforcement Learning with Double Q-Learning," in *ACM AAAI*, Phoenix, 2016.

[124] Z. Wang *et al.*, "Dueling Network Architectures for Deep Reinforcement Learning," New York, Jun. 2016.

[125] M. Fortunato *et al.*, "Noisy networks for exploration," 2017.

[126] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized Experience Replay," 2015.

[127] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy Gradient Methods for Reinforcement Learning with Function Approximation," in *Advances in neural information processing systems*, Denver, Jun. 2000.

[128] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3-4, pp. 229–256, May 1992.

[129] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," in *Advances in neural information processing systems*, Denver, Jun. 2000.

[130] V. Mnih *et al.*, “Asynchronous Methods for Deep Reinforcement Learning,” in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, New York, Jun. 2016.

[131] T. P. Lillicrap *et al.*, “Continuous Control with Deep Reinforcement Learning,” *CoRR*, vol. abs/1509.02971, 2016.

[132] J. Schulman *et al.*, “Proximal Policy Optimization Algorithms,” 2017.

[133] E. Toch, B. Lerner, E. Ben-Zion, and I. Ben-Gal, “Analyzing Large-Scale Human Mobility Data: A Survey of Machine Learning Methods and Applications,” *Knowledge and Information Systems*, vol. 58, no. 3, pp. 501 – 523, Mar. 2019.

[134] N. Bui *et al.*, “. A Survey of Anticipatory Mobile Networking: Context-Based Classification, Prediction Methodologies, and Optimization Techniques,” *IEEE COMST*, vol. 19, no. 3, pp. 1790 – 1821, 2017.

[135] M. Karimzadeh, Z. Zhao, F. Geber, and T. Braun, “Mobile Users Location Prediction with Complex Behavior Understanding,” in *IEEE LCN*, Chicago, Oct. 2018.

[136] Z. Zhao, M. Karimzadeh, F. Gerber, and T. Braun, “Mobile Crowd Location Prediction with Hybrid Features using Ensemble Learning,” *Future Generation Computer Systems*, vol. 110, pp. 556 – 571, June 2018.

[137] Q. Li, Y. Zhang, H. Huang, and J. Yan, “Deep Learning-based Short Video Recommendation and Prefetching for Mobile Commuting Users,” in *ACM SIGCOMM Workshops*, Beijing, Aug. 2019.

[138] C. Wang, Z. Zhao, Q. Sun, and H. Zhang, “Deep Learning-based Intelligent Dual Connectivity for Mobility Management in Dense Network,” in *IEEE VTC-Fall*, Aug. 2018.

[139] M. Ozturk *et al.*, “A Novel Deep Learning Driven, Low-Cost Mobility Prediction Approach for 5G Cellular Networks: The Case of the Control/Data Separation Architecture (CDSA),” *Neurocomputing*, vol. 358, pp. 479 – 489, Sept. 2019.

[140] Y. Tang *et al.*, “A Smart Caching Mechanism for Mobile Multimedia in Information Centric Networking with Edge Computing,” *Future Generation Computer Systems*, vol. 91, pp. 590 – 600, Feb. 2019.

[141] L. Hou, L. Lei, K. Zheng, and X. Wang, “A Q-Learning-based Proactive Caching Strategy for Non-Safety Related Services in Vehicular Networks,” *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4512 – 4520, Jun. 2019.

[142] H. Gebrie, H. Farroq, and A. Imran, “What Machine Learning Performs Best for Mobility Prediction in Cellular Networks?” in *IEEE ICC Workshops*, Shanghai, July 2019.

[143] C. Wang *et al.*, “Exploring Trajectory Prediction Through Machine Learning Methods,” *IEEE Access*, vol. 7, pp. 101 441 – 101 452, July 2019.

[144] I. Sutskever, O. Vinyals, and Q. Le, “Sequence to Sequence Learning with Neural Networks,” in *ACM NIPS*, Dec. 2014, pp. 3104 – 3112.

[145] R. A. Addad, D. Dutra, T. Taleb, and H. Flinck, “Toward using Reinforcement Learning for trigger selection in Network Slice Mobility,” *IEEE JSAC*, vol. 39, no. 7, pp. 2241 – 2253., Jul. 2021.

[146] S. Hajri and M. Assaad, “Energy Efficiency in Cache-Enabled Small Cell Networks With Adaptive User Clustering,” *IEEE TWC*, vol. 17, no. 2, pp. 955 – 968, Feb. 2018.

[147] Q.-V. Pham *et al.*, “A Survey of Multi-Access Edge Computing in 5G and Beyond: Fundamentals, Technology Integration, and State-of-the-Art,” *IEEE Access*, vol. 8, pp. 116 974–117 017, Jun. 2020.

[148] S. Sigdel and W.A. Krzymien, “Efficient User Selection and Ordering Algorithms for Successive Zero-Forcing Precoding for Multiuser MIMO Downlink,” in *IEEE VTC-Spring*, Barcelona, Apr. 2009.

[149] Z. Cheng, J. Yang, Z. Wei, and H. Yang, “User Clustering and Scheduling in UAV Systems Exploiting Channel Correlation,” in *IEEE PIMRC*, Istanbul, Nov. 2019.

[150] R. Trifan, R. Lerbour, G. Donnard, and Y. L. Helloco, “K-Means MU-MIMO User Clustering for Optimized Precoding Performance,” in *IEEE VTC-Spring*, Kuala Lumpur, May 2019.

[151] X. Liu, Y. Liu, and Y. Chen, “Reinforcement Learning in Multiple-UAV Networks: Deployment and Movement Design,” *IEEE TVT*, vol. 68, no. 8, pp. 8036 – 8049, Aug. 2019.

[152] Y. Xiong, Y. Chang, M. Hu, and J. Li, “Packet-Size Based Overlapping User Grouping in MU-MIMO Systems,” in *IEEE WCNC*, Marrakesh, Oct. 2019.

[153] F. Neto, D. Araujo, and T. Maciel, “Hybrid Beamforming Design based on Unsupervised Machine Learning for Millimeter Wave Systems,” *International Journal of Communication Systems*, vol. 33, no. e476, Jan. 2020.

[154] J. Ren *et al.*, “An EM-Based User Clustering Method in Non-Orthogonal Multiple Access,” *IEEE TCOM*, vol. 67, no. 12, pp. 8422 – 8434, Dec. 2019.

[155] J. Cui, Z. Ding, P. Fan, and N. Al-Dhahir, “Unsupervised Machine Learning-Based User Clustering in Millimeter-Wave-NOMA Systems,” *IEEE TWC*, vol. 17, no. 11, pp. 7425 – 7440, Nov. 2018.

[156] L. Xiao *et al.*, “IoT Security Techniques based on Machine Learning: How do IoT Devices Use AI to Enhance Security?” *IEEE Signal Processing Magazine*, vol. 35, no. 5, pp. 41 – 49, Sep. 2018.

[157] C. Benzaid and T. Taleb, “AI for Beyond 5G Networks: A Cyber-Security Defense or Offense Enabler?” *IEEE Network Magazine*, Sept. 2020.

[158] C. Moreira, G. Kaddoum, and E. Bou-Harb, “Cross-Layer Authentication Protocol Design for Ultra-Dense 5G HetNets,” in *IEEE ICC*, Kansas City, Jul. 2018.

[159] X. Qiu, T. Jiang, S. Wu, and M. Hayes, “Physical Layer Authentication Enhancement Using a Gaussian Mixture Model,” *IEEE Access*, vol. 6, pp. 53 583 – 53 592, Sep. 2018.

[160] T. Hoang, T. Duong, and S. Lambotharan, “Secure Wireless Communication Using Support Vector Machines,” in *IEEE CNS*, Washington DC, Jun. 2019.

[161] H. Fang, X. Wang, and S. Tomasin, “Machine Learning for Intelligent Authentication in 5G and Beyond Wireless Networks,” *IEEE Wireless Communications*, vol. 26, no. 5, pp. 55 – 61, Oct. 2019.

[162] X. Qiu, J. Dai, and M. Hayes, “A Learning Approach for Physical Layer Authentication Using Adaptive Neural Network,” *IEEE Access*, vol. 8, pp. 26 139 – 26 149, Feb. 2020.

[163] H. Fang, A. Qi, and X. Wang, “Fast Authentication and Progressive Authorization in Large-Scale IoT: How to Leverage AI for Security Enhancement?” *IEEE Network*, vol. 34, no. 3, pp. 24–29, Jun. 2019.

[164] M. Hirano and R. Kobayashi, “Machine Learning Based Ransomware Detection Using Storage Access Patterns Obtained From Live-forensic Hypervisor,” in *IEEE IOTSMS*, Granada, Oct. 2019.

[165] S. Sharmeen *et al.*, “Malware Threats and Detection for Industrial Mobile-IoT Networks,” *IEEE Access*, vol. 6, pp. 15 941 – 15 957, Mar. 2018.

[166] R. Xing *et al.*, “Trust-Evaluation-Based Intrusion Detection and Reinforcement Learning in Autonomous Driving,” *IEEE Network*, vol. 33, no. 5, pp. 54–60, Oct. 2019.

[167] Y. Dang, C. Benzaid, B. Yang, and T. Taleb, “Deep Learning for GPS Spoong Detection in Cellular-Enabled UAV Systems,” in *Proc. Intl Conf. NaNA*, Nov. 2021, pp. 501 – 506.

[168] Y. Dang *et al.*, “Deep Ensemble Learning based GPS Spoofing Detection for Cellular-Connected UAVs,” *IEEE Internet of Things Journal*, vol. 9, no. 24, pp. 25 068 – 25 085, Dec. 2022.

[169] ——, “Transfer Learning based GPS Spoofing Detection for Cellular-Connected UAVs,” in *IEEE IWCMC*, May 2022, pp. 629 – 634.

[170] J. Herrera and J. Camargo, “A Survey on Machine Learning Applications for Software Defined Network Security,” *Springer LNCS*, vol. 11605, pp. 70 – 93, June 2019.

[171] T. Tang *et al.*, “Deep Learning Approach for Network Intrusion Detection in Software Defined Networking,” in *IEEE WINCOM*, Fez, Oct. 2016.

[172] ——, “Deep Recurrent Neural Network for Intrusion Detection in SDN-based Networks,” in *IEEE NetSoft*, Montreal, June 2018.

[173] S. Mohammed *et al.*, “A New Machine Learning-based Collaborative DDoS Mitigation Mechanism in Software-Defined Network,” in *IEEE WiMob*, Limassol, Oct. 2018.

[174] A. Narayanadoss, T. Truong-Huu, P. Mohan, and M. Gurusamy, "Crossfire Attack Detection using Deep Learning in Software Defined ITS Networks," in *IEEE VTC2019-Spring*, Kuala Lumpur, Apr./May 2019.

[175] M. Kang, S. Lee, and V. D. Gligor, "The Crossfire Attack," in *IEEE Symposium on Security and Privacy*, Berkeley, May 2013, pp. 127 – 141.

[176] C. Benzaid, M. Boukhalfa, and T. Taleb, "Robust Self-Protection Against Application-Layer (D)DoS Attacks in SDN Environment," in *IEEE WCNC*, Seoul, May 2020.

[177] M. Siracusano, S. Shiaeles, and B. Ghita, "Detection of LDDoS Attacks based on TCP Connection Parameters," in *IEEE GIIS*, Thessaloniki, Oct. 2018.

[178] C. Mathas *et al.*, "Evaluation of Apache Spot's Machine Learning Capabilities in an SDN/NFV enabled Environment," in *ACM ARES*, Hamburg, Aug. 2018.

[179] D. Blei, A. Ng, and M. Jordan, "Latent Dirichlet Allocation," *Journal Machine Learning Research*, vol. 3, pp. 993 – 1022, Mar. 2003.

[180] A. Javadpour, F. Ja'fari, T. Taleb, and C. Benzaid, "DReinforcement Learning-based Slice Isolation against DDoS Attacks in Beyond 5G Networks," *IEEE TNSM*, Mar. 2023.

[181] ITU-T P.10/G.100, "Vocabulary for Permance, Quality of Service and Quality of Experience," Nov. 2017.

[182] K. Brunnstrom *et al.*, "Qualinet White Paper on Definitions of Quality of Experience," in *Output from the 5th Qualinet Meeting*, Mar. 2013.

[183] M. Alreshoodi and J. Woods, "Survey on QoE/QoS Correlation Models for Multimedia Services," *International Journal of Distributed and Parallel Systems*, vol. 4, no. 3, May 2013.

[184] S. Barakovic and L. Skorin-Kapov, "Survey and Challenges of QoE Management Issues in Wireless Networks," *Journal of Computer Networks and Communications*, p. 28, Mar. 2013.

[185] R. Huang, X. Wei, and L. Zhou, “A Survey of Data-driven Approach on Multimedia QoE Evaluation,” *Frontiers of Computer Science*, vol. 12, no. 6, pp. 1060 – 1075, Dec. 2018.

[186] K. Zheng *et al.*, “Quality-of-Experience Assessment and its Application to Video Services in LTE Networks,” *IEEE Wireless Communications*, vol. 22, no. 1, pp. 70 – 78, Feb. 2015.

[187] R. E. J. Kennedy, “Particle Swarm Optimization,” in *IEEE ICNN*, Perth, Dec. 1995.

[188] X. Wei *et al.*, “QoE Prediction for IPTV based on Imbalanced Dataset by the PNN-PSO Algorithm,” in *IEEE IWCMC*, Limassol, June 2018.

[189] D.F. Specht, “Probabilistic Neural Networks for Classification, Mapping, or Associative Memory,” in *IEEE ICNN*, San Diego, Jul. 1988.

[190] Y. Gao *et al.*, “QoE Prediction for IPTV based on BP_adaboost Neural Networks,” in *IEEE IWCMC*, Valencia, July 2017.

[191] Y. Freund and R. Schapire, “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting,” *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119 – 139, Aug. 1997.

[192] P. Charonyktakis, M. Plakia, I. Tsamardinos, and M. Papadopouli, “On User-Centric Modular QoE Prediction for VoIP based on Machine-Learning Algorithms,” *IEEE TMC*, vol. 15, no. 6, pp. 1443 – 1456, June 2016.

[193] C. Lv *et al.*, “QoE Prediction on Imbalanced IPTV Data based on Multi-Layer Neural Network,” in *IEEE IWCMC*, Valencia, July 2017.

[194] J. Mao *et al.*, “IPTV User QoE Prediction based on the LSTM Network,” in *IEEE WCSP*, Nanjing, Oct. 2017.

[195] S. Xu, X. Wang, and M. Huang, “Modular and Deep QoE/QoS Mapping for Multimedia Services over Satellite Networks,” in *International Journal of Communication Systems*, Aug. 2018.

[196] Y. B. Youssef, M. Afif, R. Ksantini, and S. Tabbane, “A Novel Online QoE Prediction Model based on Multiclass Incremental Support Vector Machine,” in *IEEE AINA*, Krakow, May 2018.

[197] J. Moysen, L. Giupponi, and J. Mangues-Bafalluy, “A Machine Learning enabled Network Planning Tool,” in *IEEE PIMRC*, Valencia, Sep. 2016.

[198] ——, “On the Potential of Ensemble Regression Techniques for Future Mobile Network Planning,” in *IEEE ISCC*, Messina, Jun. 2016.

[199] P. Torres et al., “Data Analytics for Forecasting Cell Congestion on LTE Networks,” in *IEEE TMA*, Dublin, Jun. 2017.

[200] D. Madariaga, M. Panza, and J. Bustos-Jimenéz, “I’m Only Unhappy when it Rains: Forecasting Mobile QoS with Weather Conditions,” in *IEEE TMA*, Vienna, Jun. 2018.

[201] I. Coma, R. Trestian, G. Muntean, and G. Ghinea, “5MART: A 5G sMART Scheduling Framework for Optimizing QoS Through Reinforcement Learning,” *IEEE TNSM*, vol. 17, no. 2, pp. 1110–1124, Dec. 2020.

[202] X. Guo, H. Lin, Z. Li, and M. Peng, “Deep Reinforcement Learning based QoS-aware Secure Routing for SDN-IoT,” *IEEE Internet of Things Journal*, Dec. 2019.

[203] A. Zhu *et al.*, “Computation Offloading for Workflow in Mobile Edge Computing Based on Deep Q-Learning,” in *IEEE WOCC*, Beijing, July 2019.

[204] J. Wang *et al.*, “Computation Offloading in Multi-Access Edge Computing Using a Deep Sequential Model Based on Reinforcement Learning,” *IEEE Communications Magazine*, vol. 57, no. 5, pp. 64–69, May 2019.

[205] Y. Wei, Z. Wang, D. Guo, and F. R. Yu, “Deep Q-Learning Based Computation Offloading Strategy for Mobile Edge Computing,” *Computers, Materials and Continua*, vol. 59, no. 1, pp. 89–104, 2019.

[206] P. Yao, X. Chen, Y. Chen, and Z. Li, “Deep Reinforcement Learning Based Offloading Scheme for Mobile Edge Computing,” in *IEEE SmartIoT*, Tianjin, Aug 2019.

[207] Y. Liu, H. Yu, S. Xie, and Y. Zhang, “Deep Reinforcement Learning for Offloading and Resource Allocation in Vehicle Edge Computing and Networks,” *IEEE TVT*, vol. 68, no. 11, pp. 11 158–11 168, Nov 2019.

[208] L. Huang, S. Bi, and Y. J. Zhang, "Deep Reinforcement Learning for Online Computation Offloading in Wireless Powered Mobile-Edge Computing Networks," *IEEE TMC*, vol. 19, no. 11, pp. 2581–2593, July 2020.

[209] P. Yang *et al.*, "Latency Optimization for Multi-user NOMA-MEC Offloading Using Reinforcement Learning," in *IEEE WOCC*, Beijing, May 2019.

[210] X. Chen *et al.*, "Optimized Computation Offloading Performance in Virtual Edge Computing Systems Via Deep Reinforcement Learning," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4005–4018, Jun. 2019.

[211] Z. Chang *et al.*, "Learn to Cache: Machine Learning for Network Edge Caching in the Big Data Era," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 28 – 35, June 2018.

[212] A. Boudi *et al.*, "AI-based Resource Management in Beyond 5G Cloud Native Environment," in *IEEE Network Magazine*, vol. 35, no. 2, Sep. 2021, pp. 128 – 135.

[213] H. Zhu *et al.*, "Deep Reinforcement Learning for Mobile Edge Caching: Review, New Features, and Open Issues," *IEEE Network*, vol. 32, no. 6, pp. 50 – 57, Nov./Dec. 2018.

[214] B. Bharath, K. Nagananda, and H. Poor, "A Learning-based Approach to Caching in Heterogeneous Small Cell Networks," *IEEE TCOM*, vol. 64, no. 4, pp. 1674 – 1686, Apr. 2016.

[215] N. Zhang, K. Zheng, and M. Tao, "Using Grouped Linear Prediction and Accelerated Reinforcement Learning for Online Content Caching," in *IEEE ICC Workshops*, Kansas City, May 2018.

[216] C. Zhong, M. Gursoy, and S. Velipasalar, "A Deep Reinforcement Learning-based Framework for Content Caching," in *IEEE CISS*, Princeton, Mar. 2018.

[217] G. Dulac-Arnold *et al.*, "Deep Reinforcement Learning in Large Discrete Action Spaces," *CoRR*, vol. abs/1512.07679, 2015.

[218] Z. Yang, Y. Liu, and Y. Chen, "Q-Learning for Content Placement in Wireless Cooperative Caching," in *IEEE GLOBECOM*, Abu Dhabi, Dec. 2018.

[219] Z. Zhang *et al.*, “Proactive Caching for Vehicular Multi-View 3D Video Streaming via Deep Reinforcement Learning,” *IEEE TWC*, vol. 18, no. 5, pp. 2693 – 2706, May 2019.

[220] S. Rathore, J. Ryu, P. Sharm, and J. Park, “DeepCachNet: A Proactive Caching Framework Based on Deep Learning in Cellular Networks,” *IEEE Network*, vol. 33, no. 3, pp. 130 – 138, May/June 2019.

[221] L. Ale *et al.*, “Online Proactive Caching in Mobile Edge Computing using Bidirectional Deep Recurrent Neural Network,” *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5520 – 5530, June 2019.

[222] C. Wang *et al.*, “Content-centric Caching using Deep Reinforcement Learning in Mobile Computing,” in *IEEE HPBD&IS*, Shenzhen, May 2019.

[223] L. Tan and R. Hu, “Mobility-Aware Edge Caching and Computing in Vehicle Networks: A Deep Reinforcement Learning,” *IEEE TVT*, vol. 67, no. 11, pp. 10 190 – 10 203, Nov. 2018.

[224] W. Li *et al.*, “Edge Caching for D2D Enabled Hierarchical Wireless Networks with Deep Reinforcement Learning,” *Wireless Communications and Mobile Computing*, p. 12, Feb. 2019.

[225] S. Niknam, H. Dhillon, and J. Reed, “Federated learning for wireless communications: Motivation, opportunities and challenges,” *IEEE Communications Magazine*, vol. 58, no. 6, pp. 46–51, Jun. 2020.

[226] A. Ndikumana *et al.*, “Deep Learning Based Caching for Self-Driving Cars in Multi-Access Edge Computing,” *IEEE Trans. on Intell. Transp. Syst.*, pp. 1–16, Mar. 2020.

[227] O. Bekkouche, K. Samdanis, M. Bagaa, and T. Taleb, “A Service-Based Architecture for enabling UAV enhanced Network Services,” *IEEE Network*, vol. 34, no. 4, pp. 328–335, Jul/Aug. 2020.

[228] B. Brik, A. Ksentini, and M. Bouaziz, “Federated Learning for UAVs-Enabled Wireless Networks: Use Cases, Challenges, and Open Problems,” *IEEE Access*, vol. 8, pp. 53 841–53 849, Mar. 2020.

[229] M. Chen *et al.*, “Caching in the Sky: Proactive Deployment of Cache-enabled Unmanned Aerial Vehicles for Optimized Quality-of-Experience,” *IEEE JSAC*, vol. 35, no. 5, pp. 1046 – 1061, May 2017.

[230] H. Jaeger and H. Haas, “Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication,” *Science*, vol. 304, no. 5667, pp. 78 – 80, Apr. 2004.

[231] M. Chen, W. Saad, and C. Yin, “Liquid State Machine Learning for Resource and Cache Management in LTE-U Unmanned Aerial Vehicle (UAV) Networks,” *IEEE TWC*, vol. 18, no. 3, pp. 1504 – 1517, Mar. 2019.

[232] N. Xia, H. Chen, and C. Yang, “Radio Resource Management in Machine-to-Machine Communications – A Survey,” *IEEE COMST*, vol. 20, no. 1, pp. 791 – 828, 2018.

[233] Y. Teng *et al.*, “Resource Allocation for Ultra-Dense Networks: A Survey, Some Research Issues and Challenges,” *IEEE COMST*, vol. 21, no. 3, pp. 2134 – 2168, 2019.

[234] Y. Wu and D.H.K. Tsang, “Distributed Power Allocation Algorithm for Spectrum Sharing Cognitive Radio Networks with QoS Guarantee,” in *IEEE INFOCOM*, Rio de Janeiro, Apr. 2009.

[235] L. Qian, Y. Zhang, and J. Huang, “MAPEL: Achieving Global Optimality for a Non-Convex Wireless Power Control Problem,” *IEEE TWC*, vol. 8, no. 3, pp. 1553 – 1563, Mar. 2009.

[236] R. Ma *et al.*, “A DBN-Based Independent Set Learning Algorithm for Capacity Optimization in Wireless Networks,” in *IEEE GLOBECOM*, Abu Dhabi, Dec. 2018.

[237] L. Lei *et al.*, “Learning-based Resource Allocation: Efficient Content Delivery Enabled by Convolutional Neural Network,” in *IEEE SPAWC*, Cannes, Jul. 2019.

[238] J. Cui, Y. Liu, and A. Nallanathan, “Multi-Agent Reinforcement Learning-Based Resource Allocation for UAV Networks,” *IEEE TWC*, vol. 19, no. 2, pp. 729 – 743, Feb. 2020.

[239] J. Zhang *et al.*, “Machine Learning Based Flexible Transmission Time Interval Scheduling for eMBB and uRLLC Coexistence Scenario,” *IEEE Access*, vol. 7, pp. 65 811 – 65 820, May 2019.

[240] I. C. et al., "Towards 5G: A Reinforcement Learning-Based Scheduling Solution for Data Traffic Management," *IEEE TNSM*, vol. 15, no. 4, pp. 1661 – 1675, Dec. 2018.

[241] D. Huang *et al.*, "Deep Learning based Cooperative Resource Allocation in 5G Wireless Networks," *Mobile Networks and Applications*, Dec. 2018.

[242] I. Comşa *et al.*, "Enhancing User Fairness in OFDMA Radio Access Networks Through Machine Learning," in *IEEE Wireless Days*, Manchester, Apr. 2019.

[243] K. Zia *et al.*, "A Distributed Multi-Agent RL-Based Autonomous Spectrum Allocation Scheme in D2D Enabled Multi-Tier HetNets," *IEEE Access*, vol. 7, pp. 6733 – 6745, Jan. 2019.

[244] Z. Li and C. Guo, "Multi-Agent Deep Reinforcement Learning Based Spectrum Allocation for D2D Underlay Communications," *IEEE TVT*, vol. 69, no. 2, pp. 1828 – 1840, Feb. 2020.

[245] M. Chen, W. Saad, C. Yin, and M. Debbah, "Data Correlation-Aware Resource Management in Wireless Virtual Reality (VR): An Echo State Transfer Learning Approach," *IEEE TCOM*, vol. 67, no. 6, pp. 4267 – 4280, June 2019.

[246] M. Chen, W. Saad, and C. Yin, "Liquid State Machine Learning for Resource and Cache Management in LTE-U Unmanned Aerial Vehicle (UAV) Networks," *IEEE TWC*, vol. 18, no. 3, pp. 1504 – 1517, Mar. 2019.

[247] W. Maass, "Liquid State Machines: Motivation Theory and Applications," *Computability in Context: Computation and Logic in the Real World*, pp. 275 – 296, 2011.

[248] I. S. Coma, G. M. Muntean, and R. Trestian, "An Innovative Machine-Learning-Based Scheduling Solution for Improving Live UHD Video Streaming Quality in Highly Dynamic Network Environments," *IEEE ToB*, vol. 67, no. 1, pp. 212 – 224, Mar. 2021.

[249] R. Li *et al.*, "Intelligent 5G: When Cellular Networks Meet Artificial Intelligence," *IEEE Wireless Communications*, vol. 24, no. 5, pp. 175 – 183, Oct. 2017.

[250] J. Xu and K. Wu, “Living with Artificial Intelligence: A Paradigm Shift toward Future Network Traffic Control,” *IEEE Network*, vol. 32, no. 6, pp. 92 – 99, Nov./Dec. 2018.

[251] I. Afolabi *et al.*, “Network Slicing & Softwarization: A Survey on Principles, Enabling Technologies & Solutions,” *IEEE COMST*, vol. 20, no. 3, pp. 72 429 – 2453, Mar. 2018.

[252] T. Taleb, I. Afolabi, K. Samdanis, and F. Z. Yousaf, “On Multi-domain Network Slicing Orchestration Architecture & Federated Resource Control,” in *IEEE Network Magazine*, vol. 33, no. 5, Sep. 2019, pp. 242 – 252.

[253] K. Samdanis, X. Costa-Perez, and V. Sciancalepore, “From Network Sharing to Multi-tenancy: The 5G Network Slice Broker,” *IEEE Communications Magazine*, vol. 54, no. 5, pp. 32–39, Jul. 2016.

[254] M. Wang *et al.*, “Machine Learning for Networking: Workflow, Advances and Opportunities,” *IEEE Network*, vol. 32, no. 2, pp. 92 – 99, Mar/Apr. 2018.

[255] D. Bega *et al.*, “Network Slicing Meets Artificial Intelligence: an AI-based Framework for Slice Management,” *IEEE Communications Magazine*, vol. 58, no. 6, pp. 32–38, Jun. 2020.

[256] ——, “DeepCog: Optimizing Resource Provisioning in Network Slicing with AI-based Capacity Forecasting,” *IEEE JSAC*, vol. 38, no. 2, pp. 361–376, Feb. 2020.

[257] J. Chen *et al.*, “iRAF: A Deep Reinforcement Learning Approach for Collaborative Mobile Edge Computing IoT Networks,” *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 7011 – 7024, Aug. 2019.

[258] H. Kim *et al.*, “Machine Learning-Based Method for Prediction of Virtual Network Function Resource Demands,” in *IEEE NetSoft*, Paris, Jun. 2019.

[259] R. Mijumbi *et al.*, “A Connectionist Approach to Dynamic Resource Management for Virtualised Network Functions,” in *IEEE CNSM*, Montreal, Oct. 2016.

[260] R. Addad *et al.*, “Towards studying Service Function Chain Migration Patterns in 5G Networks and beyond,” in *IEEE GLOBECOM*, Waikoloa, Dec. 2019.

[261] H. Jmila, M. I. Khedher, and M. A. El Yacoubi, “Estimating VNF Resource Requirements Using Machine Learning Techniques,” pp. 883–892, Oct. 2017.

[262] Y. Chen, Y. Sun, C. Wang, and T. Taleb, “Dynamic Task Allocation and Service Migration in Edge-Cloud IoT System based on Deep Reinforcement Learning,” *IEEE IoT Journal*, vol. 9, no. 18, pp. 19 501 – 19 514, Sep. 2022.

[263] S. Lange *et al.*, “Machine Learning-based Prediction of VNF Deployment Decisions in Dynamic Networks,” in *IEEE APNOMS*, Matsue, Sep. 2019.

[264] R. Shaw, E. Howley, and E. Barrett, “An Advanced Reinforcement Learning Approach for Energy-aware Virtual Machine Consolidation in Cloud Data Centers,” in *IEEE ICITST*, Cambridge, Dec. 2017.

[265] ——, “A Predictive Anti-Correlated Virtual Machine Placement Algorithm for Green Cloud Computing,” in *IEEE/ACM UCC*, Zurich, Dec. 2018.

[266] T. Miyazawa, V. Kafle, and H. Harai, “Reinforcement Learning Based Dynamic Resource Migration for Virtual Networks,” in *IFIP/IEEE IM*, Lisbon, May 2017.

[267] R. Addad *et al.*, “Fast Service Migration in 5G Trends and Scenarios,” *IEEE Network*, vol. 34, no. 2, pp. 92–98, Apr. 2020.

[268] S. Cao, Y. Wang, and C. Xu, “Service Migrations in the Cloud for Mobile Accesses: A Reinforcement Learning Approach,” in *IEEE NAS*, Shenzhen, Aug 2017.

[269] F. Farahnakian, P. Liljeberg, and J. Plosila, “LiRCUP: Linear Regression Based CPU Usage Prediction Algorithm for Live Migration of Virtual Machines in Data Centers,” in *IEEE SEAA*, Santander, Sep. 2013.

[270] R. A. Addad, D. Dutra, T. Taleb, and H. Flinck, “AI-based Network-aware Service Function Chain migration in 5G & Beyond Networks,” *IEEE TNSM*, vol. 19, no. 1, pp. 472 – 484., Mar. 2022.

[271] J. Kawasaki, G. Mouri, and Y. Suzuki, “Comparative Analysis of Network Fault Classification Using Machine Learning,” in *IEEE/IFIP NOMS*, Budapest, Apr. 2020.

[272] A. Chigurupati and N. Lassar, “Root cause analysis using artificial intelligence,” in *RAMS*, Orlando, Jan. 2017.

[273] M. Lozonavu, M. Vlachou-Konchylaki, and V. Huang, “Relation discovery of mobile network alarms with sequential pattern mining,” in *IEEE ICNC*, Santa Clara, Jan. 2017.

[274] Y. Shehu and R. Harper, “Improved Fault Localization using Transfer Learning and Language Modeling,” in *IEEE/IFIP NOMS*, Budapest, Apr. 2020.

[275] L. Sampio *et al.*, “Using NFV and Reinforcement Learning for Anomalies Detection and Mitigation in SDN,” in *IEEE ISCC*, Natal, June 2018.

[276] S. Garg *et al.*, “A Hybrid Deep Learning-based Model for Anomaly Detection in Cloud Datacenter Networks,” *IEEE TNSM*, vol. 16, no. 3, pp. 924 – 935, Sept. 2019.

[277] D. Wang *et al.*, “Dealing With Alarms in Optical Networks Using an Intelligent System ,” *IEEE Access*, vol. 7, pp. 97 760–97 770, July 2019.

[278] GSMA, “AI in Network: Use Cases in China,” Oct. 2019.

[279] J. Crawshaw, “AI in Telecom Operations: Opportunities & Obstacles,” *Heavy Reading*, Sep. 2018.

[280] A. Aijaz and M. Sooriyabandara, “The Tactile Internet for Industries: A Review,” *Proc. of the IEEE*, vol. 107, no. 2, Feb. 2019.

[281] E. Bastug, M. Bennis, M. Medard, and M. Debbah, “Towards Interconnected Virtual Reality: Opportunities, Challenges, and Enablers,” *IEEE Communication Magazine*, vol. 55, no. 6, pp. 110 – 117, June 2017.

[282] H. Yu, T. Taleb, K. Samdanis, and J. Song, “Towards Supporting Holographic Services over Deterministic 6G Integrated Terrestrial & Non-Terrestrial Networks,” *IEEE Network Magazine*, Mar. 2023.

[283] T. Taleb *et al.*, “Toward Supporting XR Services: Architecture and Enablers,” *IEEE IoT Journal*, vol. 10, no. 4, pp. 3567 – 3586, Feb. 2023.

[284] H. Mazandarani, M. Shokrnezhad, T. Taleb, and R. Li, “Self-Sustaining Multiple Access with Continual Deep Reinforcement Learning for Dynamic Metaverse Applications,” in *IEEE MetaCom*, Kyoto, Jun. 2023.

[285] Z. Su, Y. Hui, and T. H. Luan, “Distributed Task Allocation to Enable Collaborative Autonomous Driving With Network Softwarization,” *IEEE JSAC*, vol. 36, no. 10, pp. 2175–2189, Oct. 2018.

[286] 5GAA, “C-V2X Use Cases: Methodology, Examples and Service Level Requirements,” *White Paper*, June 2019.

[287] A. Taha and N. AbuAli, “Route Planning Considerations for Autonomous Vehicles,” *IEEE Communications Magazine*, vol. 56, no. 10, pp. 78–84, Oct. 2018.

[288] I. Bobriakov, “Top 10 Data Science Use cases in Telecom,” *ActiveWizards AI & ML for Startups*, Jan. 2019.

[289] T. Taleb *et al.*, “6G System Architecture A Service of Services Vision,” *ITU Journal on Future and Evolving Technologies*, vol. 3, no. 3, pp. 710 – 743, Dec. 2022.

[290] Ericsson, “Defining AI native: A key enabler for advanced intelligent telecom networks,” in *White Paper*, Feb. 2023.

[291] M. Bagaa, T. Taleb, J. Riekki, and J. Song, “Collaborative Cross System AI: Toward 5G System and Beyond,” in *IEEE Network Magazine*, vol. 35, no. 4, Jul. 2021, pp. 286 – 294.

[292] M. Camelo *et al.*, “DAEMON: A Network Intelligence Plane for 6G Networks,” in *IEEE GC Wkshps*, Rio de Janeiro, Dec. 2022.

[293] 3GPP TS 28.201, “Charging management; Network slice performance and analytics charging in the 5G System (5GS); Stage 2,” *Rel.16*, Mar. 2022.

[294] L. Zhu, Z. Liu, and S. Han, “Deep Leakage from Gradients,” in *Proc. of Advances in Neural Information Processing Systems 32*, Vancouver, Dec. 2019.

[295] Z. Wang *et al.*, “Beyond Inferring Class Representatives: User-Level Privacy Leakage From Federated Learning,” in *IEEE INFOCOM*, Paris, Apr./May 2019.

[296] Z. Du *et al.*, “Green Deep Reinforcement Learning for Radio Resource Management: Architecture, Algorithm Compression and Challenge,” *IEEE Vehicular Technology Magazine*, Sept. 2020.

[297] J. Dodge, K. Jamieson, and N. Smith, “Open Loop Hyperparameter Optimization and Determinantal Point Processes,” in *AutoML*, Aug. 2017.