Machine Learning Techniques in Advanced Network and Services Management

Eugen Borcoci
University Politehnica Bucharest
Electronics, Telecommunications and Information Technology Faculty (ETTI)
Eugen.Borcoci@elcom.pub.ro

SoftNet 2019 Conference, November 24-28, Valencia

Machine Learning Techniques in Advanced Network and Services Management

Acknowledgement

- This overview and analysis is compiled and structured, based on several **public documents** like: conference proceedings, studies (overviews, tutorials, research papers), standards, projects, etc. (see specific references in the text and the Reference list).

- **The selection and structures of this material belongs to the author.**
- **Notes:**
  - Given the extension of the topics, this **presentation is limited to a high level overview only**, mainly on architectural aspects.
  - The **presentation is not an in-depth overview of the machine learning** topics (math background is not detailed) but it basically try to **show how ML are useful in advanced network and services management.**
  - Some examples taken from the literature are selected to illustrate the application of the ML to the management of 5G networks

Machine Learning Techniques in Advanced Network and Services Management

- **Motivation of this talk**

  - **Current state in networks and services**
    - **Increased complexity** (challenges: integration of cloud/edge computing and networking technologies, big data, …)

    - **Driving forces for new IT&C technologies :** IoT, smart cities, industry governance, IoV/automotive needs, safety/emergency oriented systems, entertainment, environment, etc.

- **Why Artificial Intelligence (AI)/ Machine Learning (ML) – in networking and services?**
  - **AI/ML:**
    - are nowadays widely used in numerous areas, including networking domain
    - enable a system to explore (big)data and deduce knowledge
    - ML is considered as part of AI

CONTENTS

CONTENTS

1. Introduction

- **Current state in networks and services**

  - **Increased complexity** (challenges: integration of cloud/edge computing and networking technologies, big data, …)
  - **Driving forces for new IT&C technologies :** IoT, smart cities, industry, governance, IoV/automotive needs, safety/emergency-oriented systems, energy saving, entertainment, environment preservation, etc.

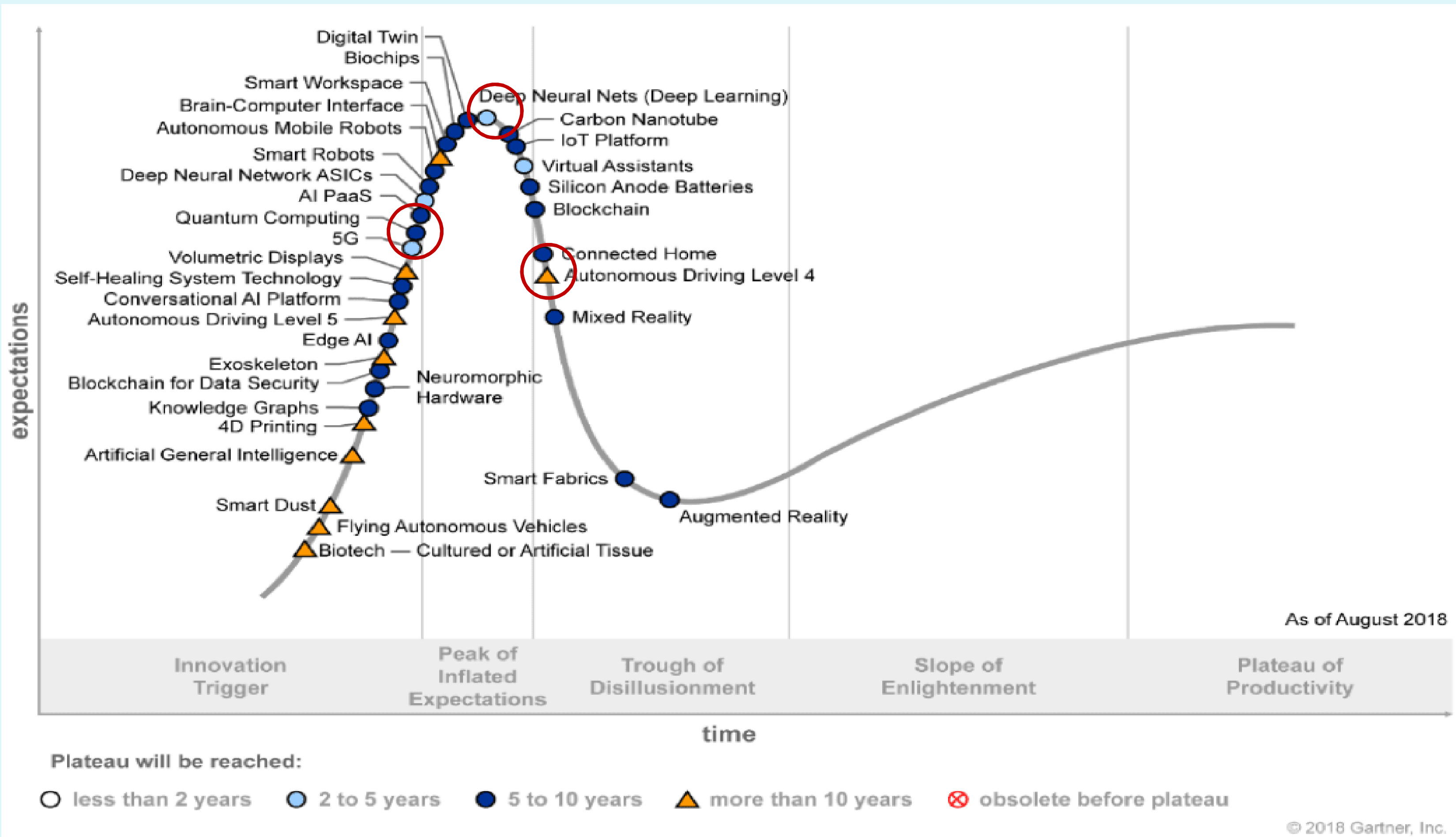
  - **Example: 5G new generation of mobility-capable networks** offering a large range of services to satisfy various customer demands
    - 5G aims to support:
      - **Large communities** of users/terminals (e.g., IoT)
      - Dedicated - **logical separated slices,** customized for various business demands, having different requirements
      - **Programmability** through softwarization, open sources and open interfaces that allow access for third parties
  - **Management and control (M&C) for 5G Multi-x** (**x= tenant, operator, provider, domain**) E2E → many **open research issues and challenges**

1. Introduction

- **AI:-** how to **train the computers** so that computers can do things which at present human can do better
- **ML**: machine can learn by its own without being explicitly programmed
  - It is an application of AI that provide system the ability to automatically learn and improve from experience
- **ML – strong methods:**
  - **widely used** nowadays in numerous areas, including networking domain
  - enable a system to **explore data** and **deduce knowledge**
  - **identify and exploit hidden patterns** in “training” data
  - go further than simply learning or extracting knowledge, towards **improving knowledge** over time and with experience
  - The **data-driven nature** facilitates **automatic learn** of the complexity of the communications and networking environment and to dynamically adjust protocols and actions without human interactions
  - ML techniques offer **additional support to network/services M&C**, operations & automation, including adaptive features

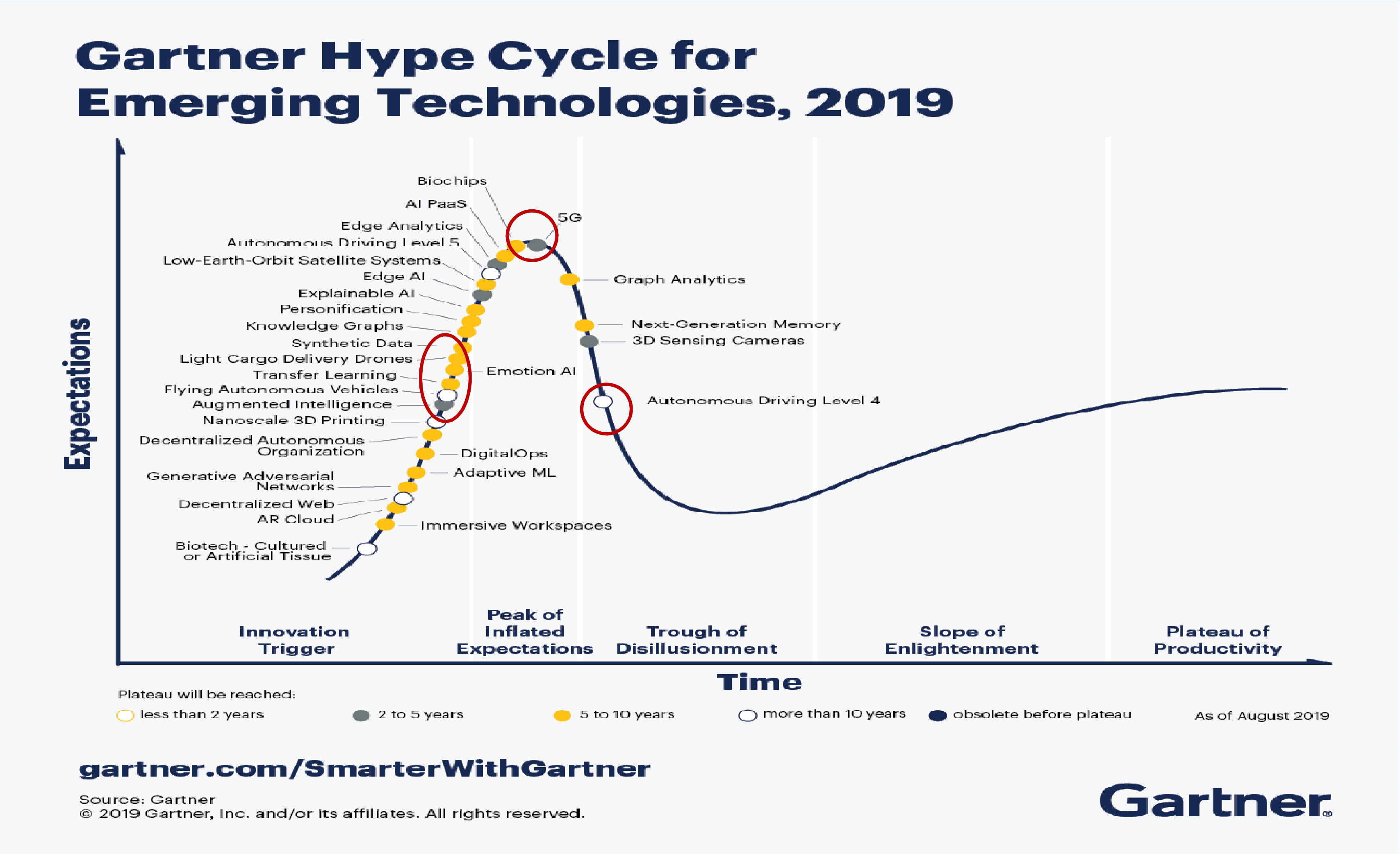
1. Introduction

- **Emerging technologies**
  - **Gartner Hype Cycle for emerging technologies 2018**



1. Introduction

- **Emerging technologies (cont'd)**



1. Introduction

- **Machine Learning – in networking and services – can contribute to**
- Adaptive and effective pattern mining
  - Learning as the data or patterns change (traffic, users/tenants requests, network conditions, etc.)
  - Scaling with network and services data

- Feature-extraction capabilities
  - Network and services management knowledge
  - Security support
  - Statistical-based procedures

- **Wide variety of architectures, methods and algorithms based on**
  - **Unsupervised (UML), Supervised (SML), Semi-supervised (SSML), Reinforcement (RL) machine learning**
  - **Deep learning (DL), Deep reinforcement learning (DRL),** etc.
  - Adaptive and automation capabilities
    - Autonomic Network Management
    - Cognitive management

CONTENTS

2. Network and services management and control – supported by machine learning

2.1 Use cases in networking supported by ML - examples

- **Traffic management and control**
  - Traffic **prediction**
    - Synthetic and real traffic traces with flow statistics
    - Link load and traffic volume prediction in ISP networks
    - Early flow-size prediction and elephant flow detection
  - Traffic **classification:** labeled and unlabeled traffic traces
    - Payload-based
    - Host-behaviour based
    - Flow feature based
- **Spectrum management in 5G**

- **Congestion control**
  - Collect experience from network simulator
  - Packet loss clasification
  - Queue management
  - Congestion window fot TCP

2. Network and services management and control – supported by machine learning

2.1 Use cases in networking supported by ML – examples (cont'd)

- **Resource management**
  - Synthetic workload with different patterns is used for training
  - Admission control
  - Resource reservation and allocation (e.g., for slices (RAN, Core network, Cloud), VNFs, etc.)- in multi-domain contexts

- **Fault magement**
  - Prediction
  - Detection
  - Localizing the faults
  - Automated mitigation

- **Network adaptation**
  - Routing stategy:
    - Traffic patterns labeling with routing paths computed by routing protocols
    - Route measurement
    - End-to-end path bandwidth availability prediction
    - Decentralized/partially centralized/ centralized routing

2. Network and services management and control – supported by machine learning

2.1 Use cases in networking supported by ML – examples (cont'd)

- **QoS and QoE management**
  - QoE optimization: Session quality information with features in large time scale
  - QoS/QoE correlation with supervised ML
  - QoS prediction under QoS impairment
  - QoS prediction for HAS and DASH
- **Performance prediction**
  - Datasets with quality measurements – e.g. from public CDNs
  - Throughput prediction: Datasets of HTTP throughput measurement
- **Network security**
  - Intrusion detection: Misuse-based, Anomaly-based
  - Anomaly detection
  - Hybrid intrusion detection
- **Mobile networks**
  - Network- Level Mobile Data Analysis
  - Mobility analysis, User localization, Mobile networks applications
- **IoT**
- **Wireless sensor networks**

2. Network and services management and control – supported by machine learning

2.2 FCAPS challenges solvable by ML techniques – examples

- **Reminder: classical management systems have the functions**
  - **F- Failure** detection (based on monitoring) and repairing
  - **C- Configuration** of the entities (physical, logical)
  - **A - Accounting** of resource usage (who, what, when, how much)
  - **P- Performance** evaluation (in order to check Service Level Agreements fulfillment)
  - **S - Security** – protection of the system

2. Network and services management and control – supported by machine learning

2.2 FCAPS - challenges solvable by ML techniques – examples (cont'd)

- **Failure Prevention:**
  - **Proactive mitigation** combined with **fault prediction** can prevent upcoming failures
    - To select the mitigation step, the root cause of the predicted fault has to be identified
    - But, existing ML-based localization approaches: still poor scalability for the high-dimensional device-log-attributes , even in moderate-size networks → dimensionality reduction is needed
- **Fault Management in Cloud and Virtualized Environments**
  - The multi-tenancy in cloud/NFV environment raises the complexity and dimensions of the fault space in a network
  - ML (in particular DeepNNs) can **model complex multi-dimensional state spaces-** -- > used to predict and locate faults in such networks
  - Any **automated mitigation** within a Virtual Network (VN)/slice should not affect other coexisting VNs
    - ML (in particular- **RL combined with DNNs**) can learn to optimize mitigation steps

Adapted from : Sara Ayoubi, et.al., Machine Learning for Cognitive Network Management, IEEE Comm.Magazine , January 2018, pp.158-165

2. Network and services management and control – supported by machine learning

- **2.2 FCAPS - challenges solvable by ML techniques – examples** (cont'd)
- **Performance Management**
  - **Adaptive Probing**
    - Large number of devices, parameters, small time intervals to log data → high overhead for measuring traffic
    - **Regression,** mostly based on time series data, can predict the value of the measured parameters → can optimize probing
    - Objective: to set probing rates that keep the measuring traffic overhead enough low, while minimizing performance degradation and providing high prediction accuracy
  - **Detecting Patterns of Degradation**
    - Need to detect the characteristic patterns of degradation before the quality drops below an acceptable level
    - Elastic resource allocation can dynamically accommodate user demands for achieving optimum performance while maximizing resource utilization
    - ML (in particular SML) can **predict the value of network perf**.
    - However, employing performance prediction for autonomic tuning of the network behavior is still a challenge

2. Network and services management and control – supported by machine learning

- **2.2 FCAPS - challenges solvable by ML techniques – examples** (cont'd)
- **Configuration Management:**
  - **Mapping** *High-Level Requirements* to *Low-Level Configurations*:
    - There is a gap between high-level slice/services requirements and low-level configurations (e.g., resources to be provisioned)
    - RL techniques can be applied
    - The reward for selecting a configuration setting of a given network element can be seen as the utility indication of that particular setting in delivering the high-level requirements under a given network condition
  - **Configuration and Verification**
    - Configuration changes (e.g., access control lists, routing tables) should comply with high-level requirements and not adversely affect the expected network behavior
    - Interest exists in applying DL-aided verification, code correction, and theorem proving

Source: Sara Ayoubi, et.al., Machine Learning for Cognitive Network Management, IEEE Comm.Magazine , January 2018, pp.158-165

2. Network and services management and control – supported by machine learning

2.3 Cognitive Management concepts

- Current networks have complex management requirements related to multi – tenant/domain/operator context and softwarization of network resources

- Need of real-time mgmt. based on a hierarchy of complex decision making techniques that analyze *historical, temporal and frequency network data*

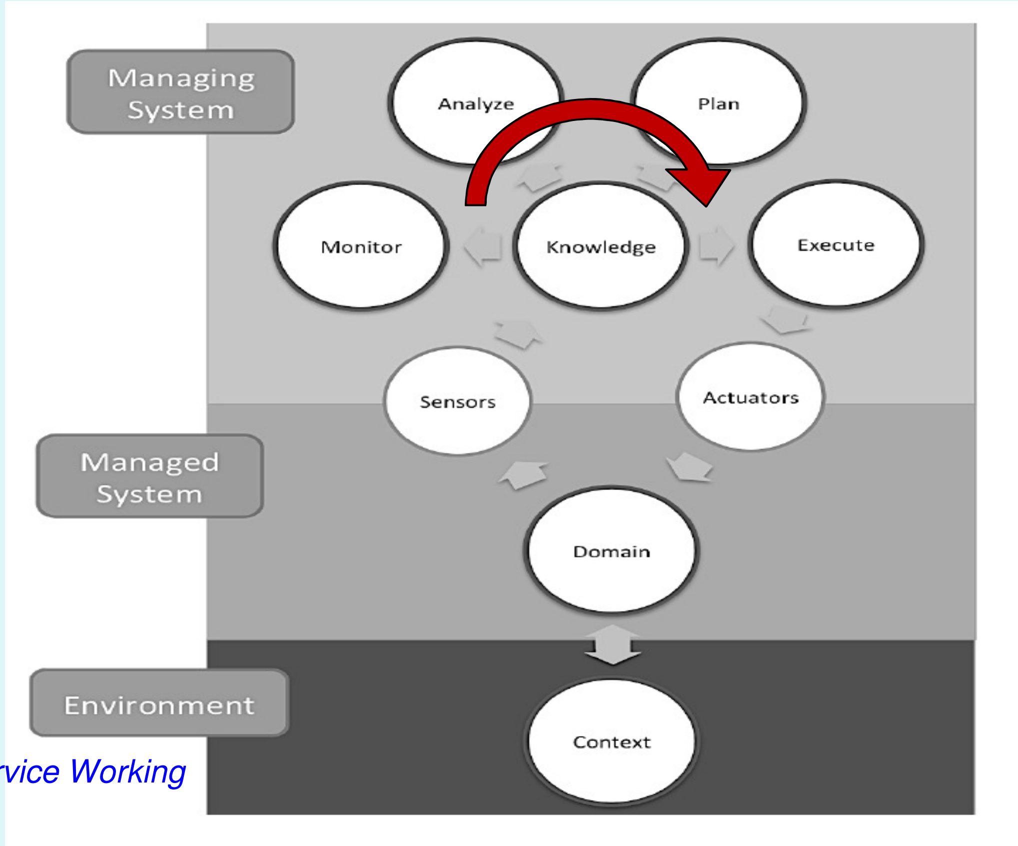
- **Cognitive network management** – recent trend using **AI**/**ML** and in particular to develop **self-x capabilities** (-x= -aware, -configuring, -optimization, -healing and -protecting systems)
- Cognitive management– extension of **Autonomic Management (AM)** (coined by IBM ~ 2001)- later extended in networking domain → ANM
  - **AM + Machine learning = Cognitive Management (CogM)**

- **Challenge:** to deploy the CogM and its orchestration across multiple heterogeneous networks: Radio & Other Access Networks, Edge/ Aggregation/ Core Networks, Edge and centralized Computing Clouds, Satellite Networks

2. Network and services management and control – supported by machine learning

- **2.3 Cognitive Management concepts** (cont'd)
- **Autonomous Network Management (ANM) :** introduce self-governed networks for pursuing business and network goals while maintaining performanc

  - **Loop:** The **Monitor-Analyse-Plan-Execute** over a shared Knowledge
  - (**MAPE-K**) is a control theory-based feedback model for self-adaptive systems
  - Full-duplex communication between ***managing systems managed system*** and the ***environment***
  - **AM –** hierarchical and recursive approach



Source: 5GPPP Network Management & Quality of Service Working Group, "Cognitive Network Management for 5G", 2017

2. Network and services management and control – supported by machine learning

2.3 Cognitive Management concepts (cont'd)

- **Autonomic Network Management functions**
  - **Monitoring**: active/passive, centralized/distributed, granularity-based, timing-based and programmable
  - **Analysis:** many approaches exist –relying, e.g., on probability and Bayesian models for anticipation on knowledge, timing, mechanisms, network-level, user applications
    - **Challenge: to define a concentrated data set** that comprehensively captures information across all anticipation points.
    - Recent solutions – **use learning and reasoning** to achieve specific goals
  - **Planning and Execution**
    - The network **adaptation plan** – several aspects: knowledge, strategy, purposefulness, degree of adaptation autonomy, stimuli, adaptation rate, temporal/spatial scope, open/closed adaptation and security
    - **Current status**: the adaptation solutions differ broadly and there is no unanimity in defining proper planning and execution guideline

2. Network and services management and control – supported by machine learning

2.3 Cognitive Management concepts (cont'd)

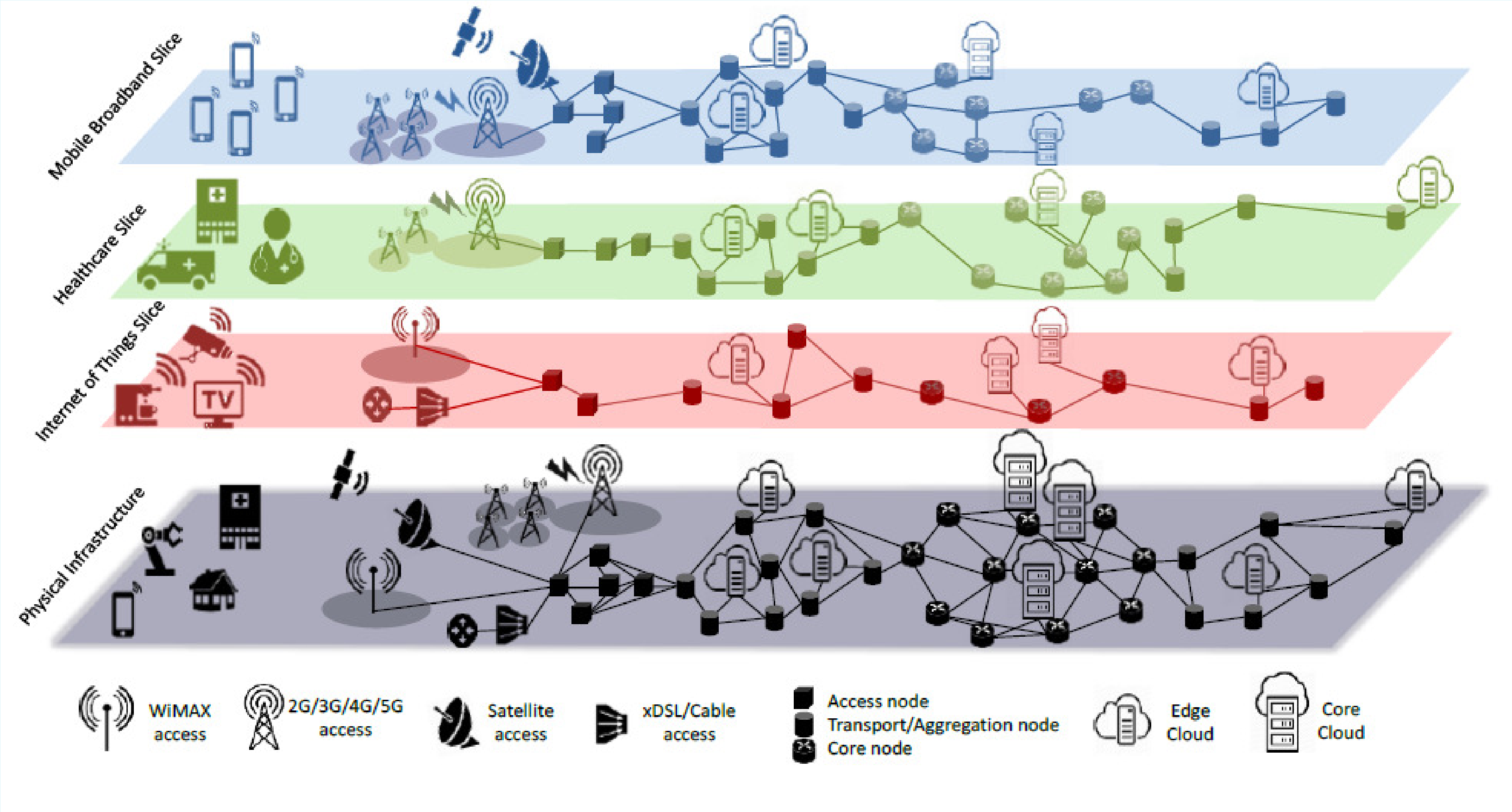
- **Autonomic Network Management functions** (cont'd)
  - **Knowledge base**
  - The network information is shared across the MAPE-K architecture
  - Many approaches exist - to build knowledge on network/topology, including models from learning and reasoning, ontology and DEN-ng models.

  - Integrated solution- able to capture knowledge on: ***structure***, ***control*** and ***behaviour***

  - **Typically:**
    - In order to drive the decisions of **Self Organizing Network (SON)-type** (e.g., self-planning, self-optimization and self-healing), the knowledge-based framework should :
      - process the input data from multiple sources
      - extract relevant knowledge, **through learning-based classification, prediction and clustering models**

2. Network and services management and control – supported by machine learning

2.4 Example: 5G slicing management and control (M&C) aspects



Source: J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J.J. Ramos-Munoz, J. Lorca, J. Folgueira, Network "Slicing for 5G with SDN/NFV: Concepts, Architectures and Challenges",
IEEE Communications Magazine, 2017, Citation information: DOI 10.1109/MCOM.2017.1600935

2. Network and services management and control – supported by machine learning

- **2.4 Example: 5G slicing management and control (M&C) aspects** (cont'd)
  - Problems to be solved
    - Service/data model & mapping on slices
    - Customized slice design and preparation, stitching / composition in a single domain and cross-domain
    - Network slice life cycle management, monitoring and updating
  - **M&C system: should react in real - time, based on complex decision making techniques that analyse historical, temporal and frequency network data**
    - **Cognitive network management technologies are added to traditional M&C**
      - Using AI/ML one can achieve: **self-x** characteristics – of the network, where [x= awareness, configuration, optimization, healing, protecting]
    - Generally, **self – organizing – networks (SON)** capabilities can be achieved

2. Network and services management and control – supported by machine learning

2.4 Example: 5G slicing management and control (M&C) aspects (cont'd)

- **Network functions requiring automation**
  - **Planning and design**: Requirements and environment analysis, topology determination; it provide inputs to
  - **Construction and deployment**: Static resource allocation, VNF placement, orchestration actions; it provide inputs to
  - **Operation, control and management**: Dynamic resource (re)allocation, adjustment; policy adaptation; it interact bi-directionally with
    - **Fault detection**: Syslog analysis, behavior analysis, fault localization
    - **Monitoring**: Workload, performance, resource utilization
    - **Security:** Traffic analysis, DPI, threat identification, infection isolation

Adapted from source: V. P. Kafle, et. al., "Consideration on Automation of 5G Network slicing with Machine Learning" , ITU Caleidoscope Santafe 2018

CONTENTS

3. Machine learning summary

- **3.1 General characteristics**
- **(ML) (subset of AI)**
- **Traditional programming**
  - <u>Input Data, **Rules** (function)</u> → **Computing Machine** →<u>Output data</u>

- **<u>ML: The rules are not known in advance, but discovered by a machine</u>**
- **ML idea: "Optimizing a performance criterion** using **example data and past experience"**
  - "A computer program is said **to learn** from **experience E** with, respect to some **class of tasks T** and **performance measure P**, if its performance at tasks in T, as measured by P, **improves with experience E** "
    - *Tom Mitchell. Machine Learning 1997*
  - The **experience E** comes usually in the form of data
    - A **learning algorithm** is used to discover and learn knowledge or properties from the data
  - In some cases, algorithms learn by rewards and/or punishments
  - The **dataset quality or quantity** affect the learning and prediction perf.
  - After first learning, the ML can provide results, for new input unknown data

See also other sources, like: Wei-Lun Chao, Machine Learning Tutorial, 2011, http://disp.ee.ntu.edu.tw/~pujols/Machine%20Learning%20Tutorial.pdf

3. Machine learning summary

3.2 Machine Learning (ML) methods (I)

- **Supervised learning (SML)**-predicting one or more dependent variables
  - based on (initially) labeled data
  - *use cases examples*: **classification** and **regression**
  - **semi-supervised learning (SSML)**: not all data is labeled
  - active learning: the algorithm has to ask for some labels with a limited budget

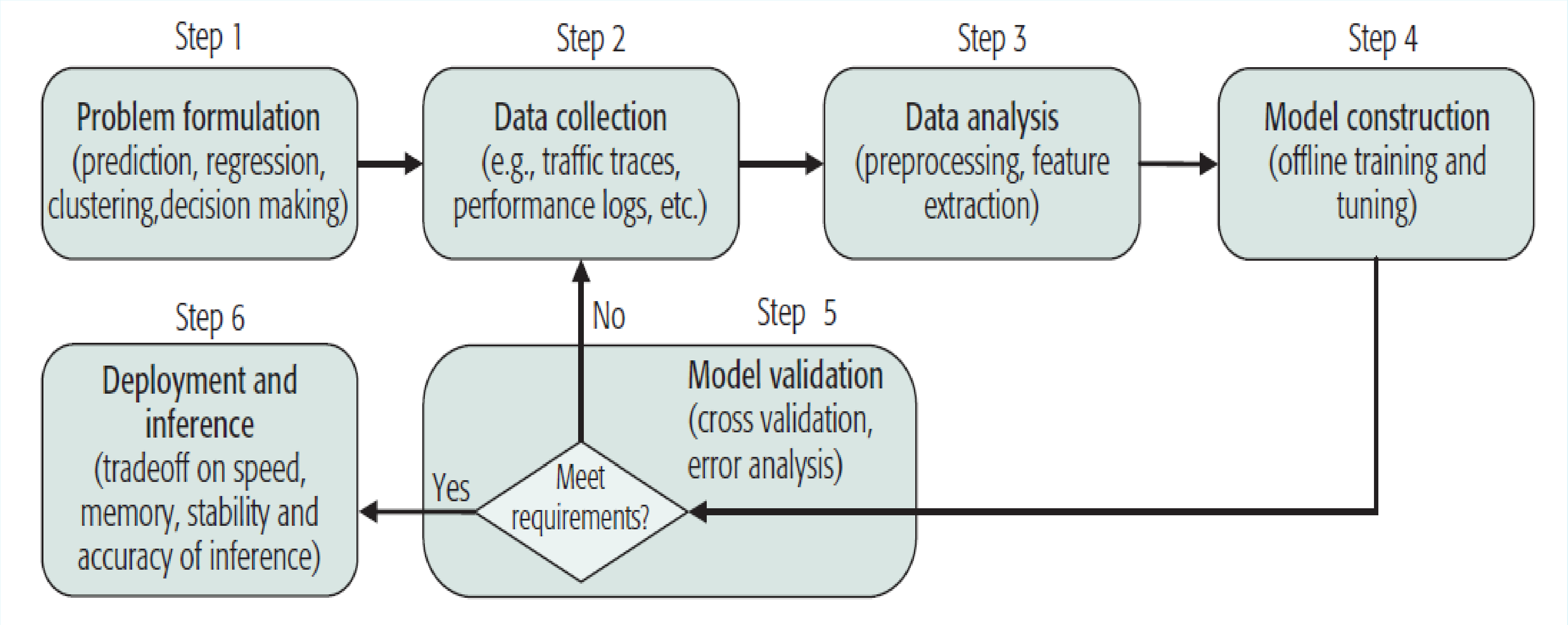
- **Unsupervised learning (UML)**-look for structure in (unlabelled) data sets
  - *use cases examples*: clustering or pattern mining

- **Reinforcement learning (RL)** -using **feedback** to an **agent actions** in a dynamic environment
  - *use cases examples*: self driving cars, learning games, ...
  - no feedback exists on individual actions, just win or lose information

- **Deep learning (DL) -** use a cascade of *multiple layers* of nonlinear processing units for *feature extraction* and *transformation*
- **Deep Reinforcement Learning**: **DRL = RL + DL (**general intelligence)
  - RL defines the objective; DL gives the mechanism

3. Machine learning summary

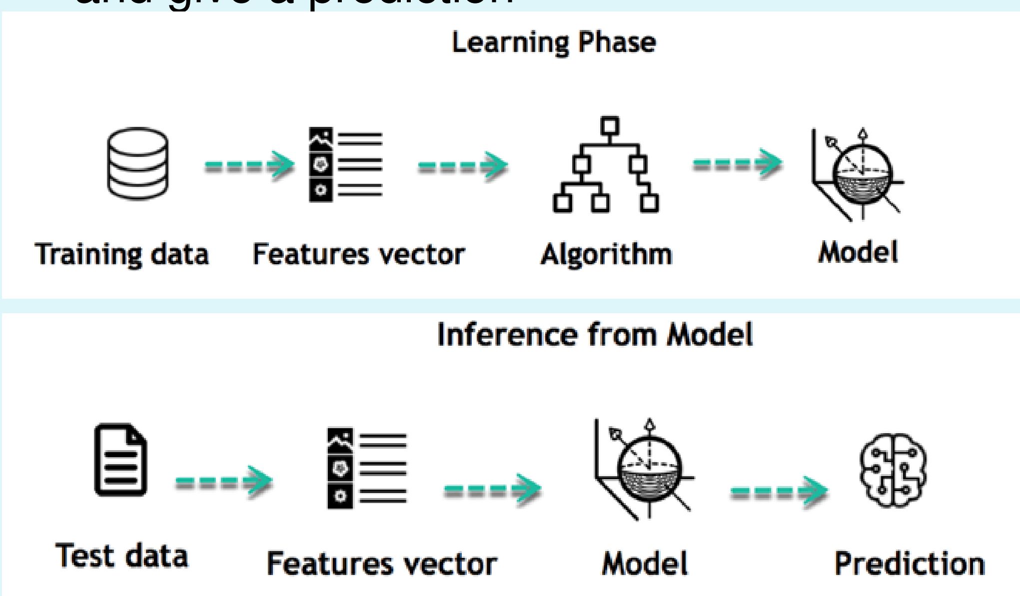
- **3.3 The typical workflow of ML (e.g., for networking)**



Source: M.Wang, et al., "Machine Learning for Networking: Workflow, Advances and Opportunities", IEEE Network Vol.32, Issue 2 , March-April 2018

3. Machine learning summary

- **3.4 ML general working phases**
- **Phase (1) learning**: through the discovery of patterns in input data
  - The list of attributes used to solve **a problem** is called a ***features vector*** (this is a subset of data that are used to tackle a problem)
  - Phase (1) is used to describe the data and summarize it into a model
- **Phase (2) Inference: using the model, test it on never-seen-before data**
  - The new data are transformed into a features vector, go through the model and give a prediction



General steps in ML life cycle

1. Define a question
2. Collect data
3. Visualize data
4. ***Train algorithm***
5. ***Test the algorithm***
6. ***Collect feedback***
7. ***Refine the algorithm***
8. Loop 4-7 until the results are satisfying
9. Use the model to make a prediction

Source: "Machine Learning Tutorial for Beginners", https://www.guru99.com/machine-learning-tutorial.html

3. Machine learning summary

- **3.4 ML general working phases** (cont'd)
- In order **to construct a ML** system – a **human designer expert should**
    - define the questions/tasks, terminology, evaluation metrics
    - define the annotation of the training and testing data
    - have a good intuition on useful feature definition
        - take care: defining the features, could be more important than the choice of learning algorithm
    - define the procedure for error analysis
    - define the constraints to guide the learning process

- **Evaluation guidelines in ML**
  - 1. **Phase 1 (training)** : selection of an evaluation procedure (a "metric"), e.g.:
    - for classification : accuracy- proportion of correct classifications?
    - for regression: mean squared error can be used
  - 2. **Phase 2 (testing on new data):** applying the model to a test set and evaluate
    - The test set must be different from the training set

See also: R.Johansson, Applied Machine Learning Lecture 1: Introduction, Univ. of Gothenburg, 2019, http://www.cse.chalmers.se/~richajo/dit866/lectures/l1/l1.pdf

3. Machine learning summary

3.5 ML – examples of generic problems solved

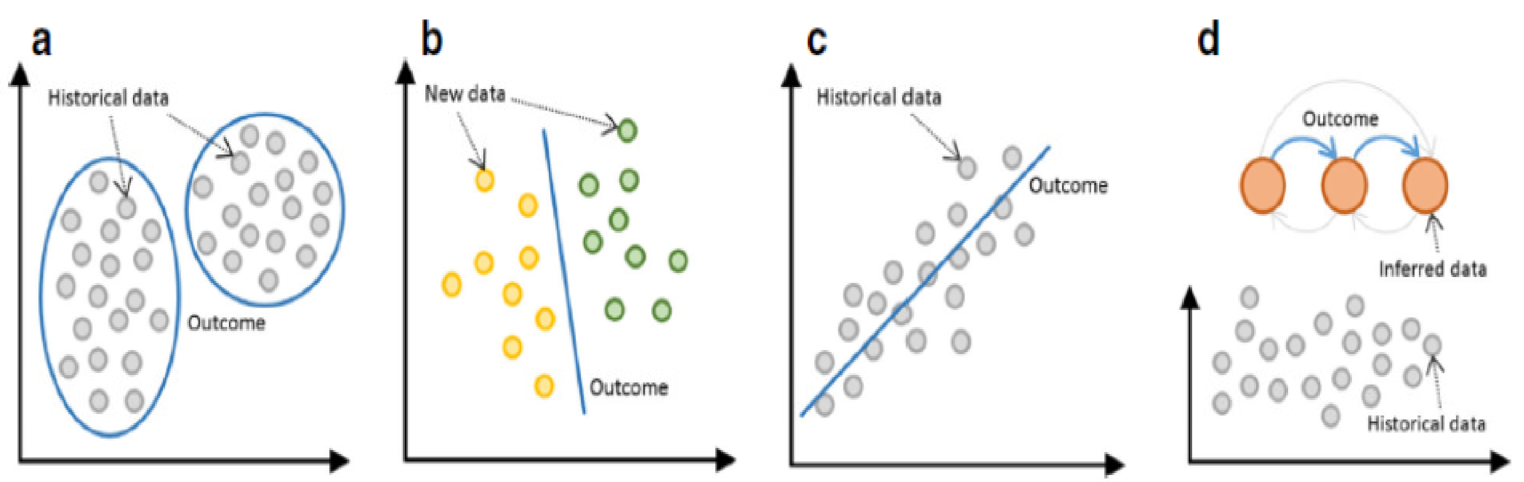
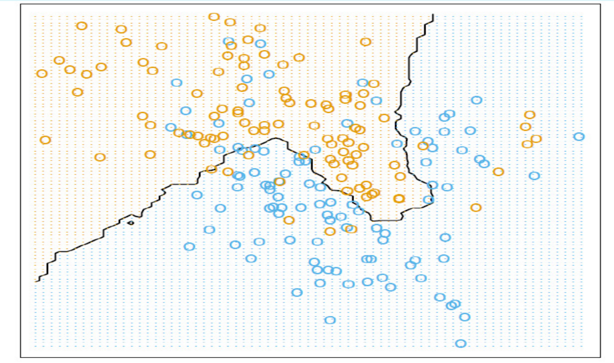


Fig. 2 Problem categories that benefit from machine learning. **a** Clustering. **b** Classification. **c** Regression. **d** Rule extraction

Source: R. Boutaba et al., "A comprehensive survey on machine learning for networking: evolution, applications and research opportunities" Journal of Internet Services and Applications (2018) 9:16 https://doi.org/10.1186/s13174-018-0087-2

3. Machine learning summary

- **3.6 Data set**
  - in ML an ***universal dataset*** is assumed to exist, containing all the possible data pairs + probability distribution of appearance in the real world
  - in real apps., only a *subset* of the universal dataset is observed (because of reasons such as: memory limits, etc.)
    - this ***acquired*** dataset = ***training set*** and used to learn the properties and knowledge of the universal dataset
      - generally, vectors in the training set are assumed *independently and identically distributed* sampled (i.i.d) from the universal dataset
    - to examine the learning performance, another dataset may be reserved for testing, called the ***test set***



Two-class dataset example

- two measurements of each sample are extracted
- each sample is a 2-D vector

Source: Wei-Lun Chao, Machine Learning Tutorial, 2011, http://disp.ee.ntu.edu.tw/~pujols/Machine%20Learning%20Tutorial.pdf

3. Machine learning summary

- **3.6 Data set** (cont'd)
  - **General dataset types**
  - **Labeled dataset** $\mathbb{D}$ **:** $X = \{\boldsymbol{x}^{(n)} \in R^d\}_{n=1}^N, \quad Y = \{y^{(n)} \in R\}_{n=1}^N$

$\{\boldsymbol{x}^{(n)} \in R^d, y^{(n)} \in R\}_{n=1}^N$, where each $\{\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}\}$ is called a data pair.

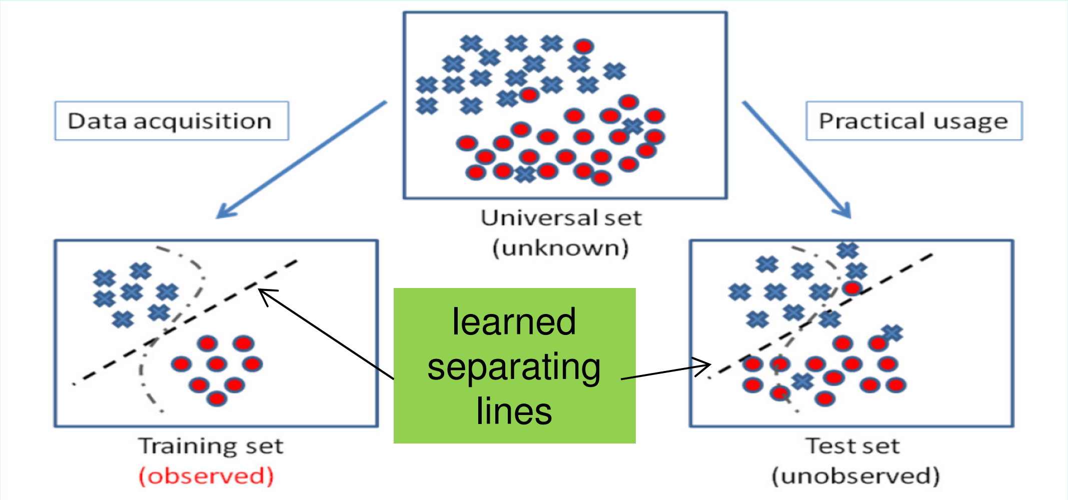
Unlabeled dataset $\mathbb{D}$ **:** $X = \{\boldsymbol{x}^{(n)} \in R^d\}_{n=1}^N$

  - X denotes the ***feature set*** containing N samples. Each sample is a ***d-dimensional vector*** d-dimensional vector $\boldsymbol{x}^{(n)} = [x_1^{(n)}, x_2^{(n)}, \ldots\ldots, x_d^{(n)}]^T$
  - Each dimension of a vector is called an *attribute, feature, variable, or element*
  - Y = the ***label set,*** recording what label a feature vector corresponds (e.g., color of the points in the previous picture)
    - In some applications, the label set is *unobserved* or *ignored*

Source: Wei-Lun Chao, Machine Learning Tutorial, 2011, http://disp.ee.ntu.edu.tw/~pujols/Machine%20Learning%20Tutorial.pdf

3. Machine learning summary

- **3.6 Data set** (cont'd)
- **Data set types illustration**
  - A subset of universal set ***is observed*** through the ***data acquisition process***,
    - **used for training** (training set)
    - Two ***learned separating lines*** are shown in the training set and test set
    - these two lines give
      - 100% accuracy on the training set
      - they may perform differently in the test set (the curved line shows higher error rate



Source: Wei-Lun Chao, Machine Learning Tutorial, 2011, http://disp.ee.ntu.edu.tw/~pujols/Machine%20Learning%20Tutorial.pdf

3. Machine learning summary

- **3.7 Machine learning methods (II)**
- **Supervised learning (SML)**
- Given a set of N training examples of the form $\{(x_1, y_1),..(x_n,y_n)\}$ such that
  - x_i is the feature vector of the i-th example and y_i is its label (i.e., class),
  - **a learning algorithm seeks a function g:X->Y where X is the input space and Y is the output space**
- The **function g** is an element of a space of possible functions G, usually called the **hypothesis space**
- It is convenient to represent g using a **scoring function f: X x Y-->R**
  - such that **g is defined** as that function which (returning the y value) **gives the highest score**: $g(x)=\arg\max_y f(x,y)$

Example: y_i is the output associated with the vector x_i

Given: Training data: $(x_1, y_1), \ldots, (x_n, y_n)$ / $x_i \in \mathbb{R}^d$ and y_i is the label.

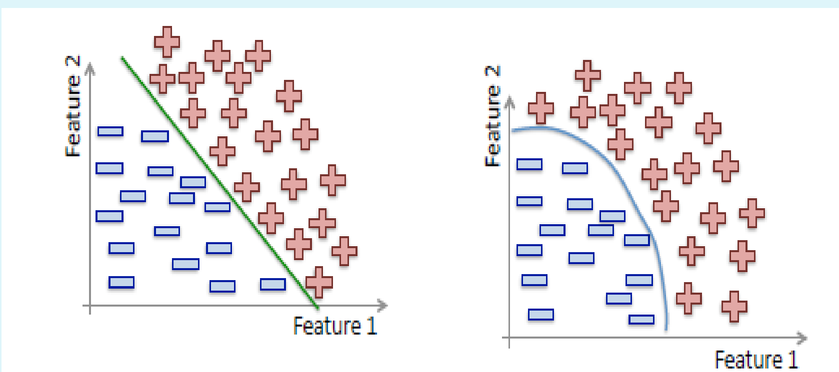
example $x_1 \rightarrow$	x_{11}	x_{12}	...	x_{1d}	$y_1 \leftarrow$ label
...	...	...	...	...	...
example $x_i \rightarrow$	x_{i1}	x_{i2}	...	x_{id}	$y_i \leftarrow$ label
...	...	...	...	...	...
example $x_n \rightarrow$	x_{n1}	x_{n2}	...	x_{nd}	$y_n \leftarrow$ label

Source: Machine Learning Basic Concepts https://courses.edx.org/asset-1:ColumbiaX+CSMM.101x+1T2017+type@asset+block@AI_edx_ml_5.1intro.pdf

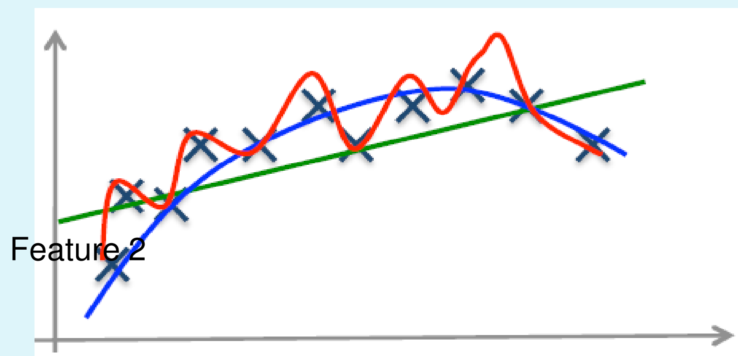
3. Machine learning summary

- **3.7 Machine learning methods (II)**
- **Supervised learning (SML)** (cont'd)
- **SML algorithms-main applications: classification and regression**
  - **Classification** problem: if each feature vector x is corresponding to a label $y \in L, L = \{l_1, l_2, \ldots\ldots, l_c\}$ *(c ranges* from 2 to hundreds)
  - **Regression** problem: if each feature vector x is corresponding to a real value y (belongs R)

Classification



Regression



Feature 1

Source: Machine Learning Basic Concepts https://courses.edx.org/asset-1:ColumbiaX+CSMM.101x+1T2017+type@asset+block@AI_edx_ml_5.1intro.pdf

3. Machine learning summary

- **3.7 Machine learning methods (II)**
- **Supervised learning (SML)** (cont'd)
- **Variants of SML algorithms**

  - **k-Nearest Neighbors** (k-NN) - can be used for classification and regression
    - k-NN is a non-linear method where the input consists of training samples in the input space (labeled)
    - an *Euclidian distance is defined*
    - a new data point (sample) is classified by **considering the majority vote** of the labels of its k- nearest neighbors

  - **Generalized Linear Models (GLM)-** it describes a linear relationship between the output and one or more input variables

  - **Naive Bayes (NB)** - used for classification and is based on Bayes theorem, i.e., calculating probabilities based on the prior probability. The main task is to classify new data points as they arrive

3. Machine learning summary

- **3.7 Machine learning methods (II)**
- **Supervised learning (SML)** (cont'd)
- **Variants of SML algorithms**
- **Support Vector Machines (SVMs)** - are inspired by statistical learning theory for estimating multidimensional functions
  - Math. optimization problem, solvable by known techniques
  - **Problem**: given m training samples ((x1; y1);…; (xm; ym)), the goal is to learn the parameters of a function which best fit the data

- **Artificial Neural Network (ANN)** is a statistical learning model where the interconnected nodes represent the neurons producing appropriate responses
  - The basic idea is to efficiently train and validate a neural network. Then, the trained network is used to make a prediction on the test set

- **Decision Trees (DT)** is a flow-chart model in which
  - each internal node represents a test on an attribute
  - each leaf node represents a response
  - branch represents the outcome of the test
- **Usage examples for SVM, ANN, DT**: *classification and regression*

3. Machine learning summary

- **3.7 Machine learning methods (II)**
- **Supervised learning (SML)** (cont'd)
- **Variants of SML algorithms**

  - **Similarity learning -** closely related to regression and classification
    - Goal: learning from examples, using a *similarity function* that measures how similar or related two objects are
    - Applications examples: ranking, recommendation systems, visual identity tracking, face/speaker verification

3. Machine learning summary

- **3.7 Machine learning methods (II)**
- **Example 1 of ML: Supervised learning**
- ***k-Nearest Neighbors*** *(k-NN) – classification example*
  - **Assumption**: the features used to describe the domain points are relevant to their labeling in a way that makes close-by points likely to have the same label
    - Euclidian distance is used
  - k-NN figures out a label on any test point without searching for a predictor within some predefined class of functions
  - **Idea:**
    - memorize the training set, then
    - predict the label of any *new instance/data_point* on the basis of the labels of its closest neighbors in the training set
      - a new *data_point* is classified by a majority vote of its k- nearest neighbors

Source: S. Shalev-Shwartz and S.Ben-David, Understanding Machine Learning: From Theory to Algorithms 2014, Cambridge University Press

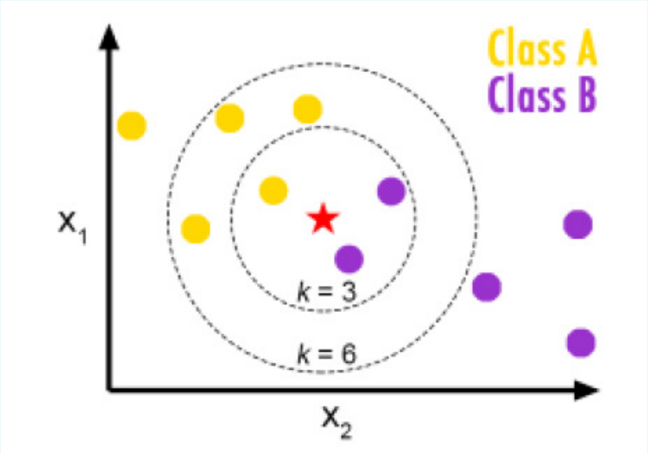
3. Machine learning summary

- **3.7 Machine learning methods (II)**
- **Example 1 of ML: Supervised learning** (cont'd)
- ***k-Nearest Neighbors*** *(k-NN) – classification example (cont'd)*
  - Let it be an instance domain, X and "points" $\mathrm{x} \in \mathcal{X}$ $\mathcal{X} = \mathbb{R}^d$
  - Define : Euclidean distance, $\rho(\mathrm{x}, \mathrm{x}') = \|\mathrm{x} - \mathrm{x}'\| = \sqrt{\sum_{i=1}^d (x_i - x_i')^2}$
  - Let $S = (\mathrm{x}_1, y_1), \ldots, (\mathrm{x}_m, y_m)$ be a sequence of training examples.

    let $\pi_1(\mathrm{x}), \ldots, \pi_m(\mathrm{x})$ be a reordering of $\{1, \ldots, m\}$ $\rho(\mathrm{x}, \mathrm{x}_{\pi_i(\mathrm{x})}) \leq \rho(\mathrm{x}, \mathrm{x}_{\pi_{i+1}(\mathrm{x})})$

> **k-NN**
>
> **input:** a training sample $S = (\mathrm{x}_1, y_1), \ldots, (\mathrm{x}_m, y_m)$
> **output:** for every point $\mathrm{x} \in \mathcal{X}$,
> return the majority label among $\{y_{\pi_i(\mathrm{x})} : i \leq k\}$

Example
Decision on classification of the new test ***Star*** point :
K=3→ Class B
K= 6 → Class A



Source: S. Shalev-Shwartz and S.Ben-David, Understanding Machine Learning: From Theory to Algorithms 2014, Cambridge University Press

3. Machine learning summary

- **3.7 Machine learning methods (II)**
- **Example 1 of ML: Supervised learning** (cont'd)

- ***k-Nearest Neighbors*** *(k-NN) – classification example (cont'd)*

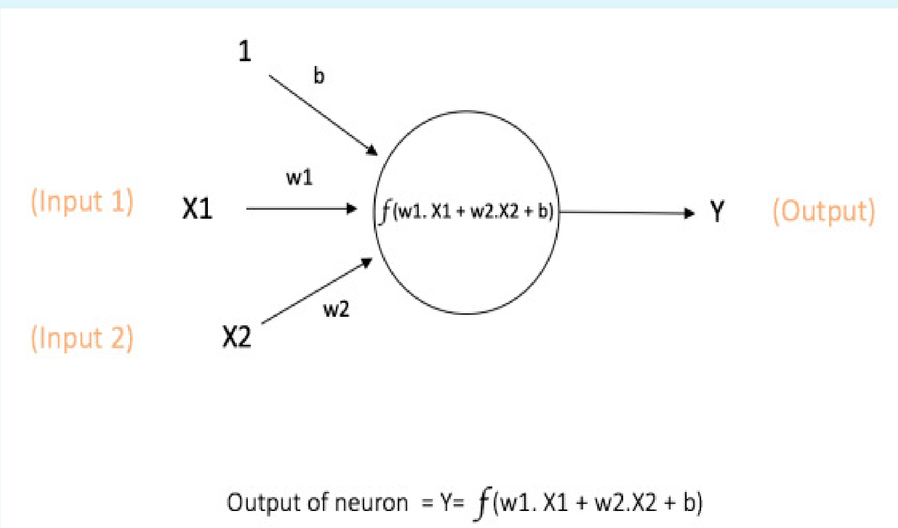
- k-NN can be also used for regression applications

- **K-NN Pros:**
  - Simple to implement
  - Works well in practice
  - Does not require to build a model, make assumptions, tune parameters
  - Can be extended easily with news examples

- **K-NN Cons:**
  - Requires large space to store the entire training dataset.
  - Slow! Given n examples and d features. The method takes **O**(n x d) to run
  - Dimensionality problem

3. Machine learning summary

- **3.7 Machine learning methods (II)**
- **Example 2 of ML: (Artificial) Neural Networks (ANN) – summary**
  - ANN- - computational model inspired by the biological neural networks in the human brain
  - The basic unit of computation: **neuron (node)**
  - The node applies an **activation (non-linear) function *f*** to the weighted sum of its inputs (including a bias)
  - The ***bias*** provides every node with a trainable constant value (in addition to the normal inputs)

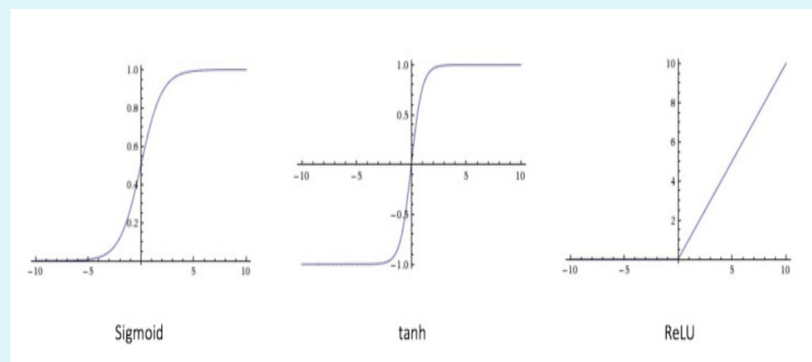


Examples of activation functions

Sigmoid: $\sigma(x) = 1 / (1 + \exp(-x))$

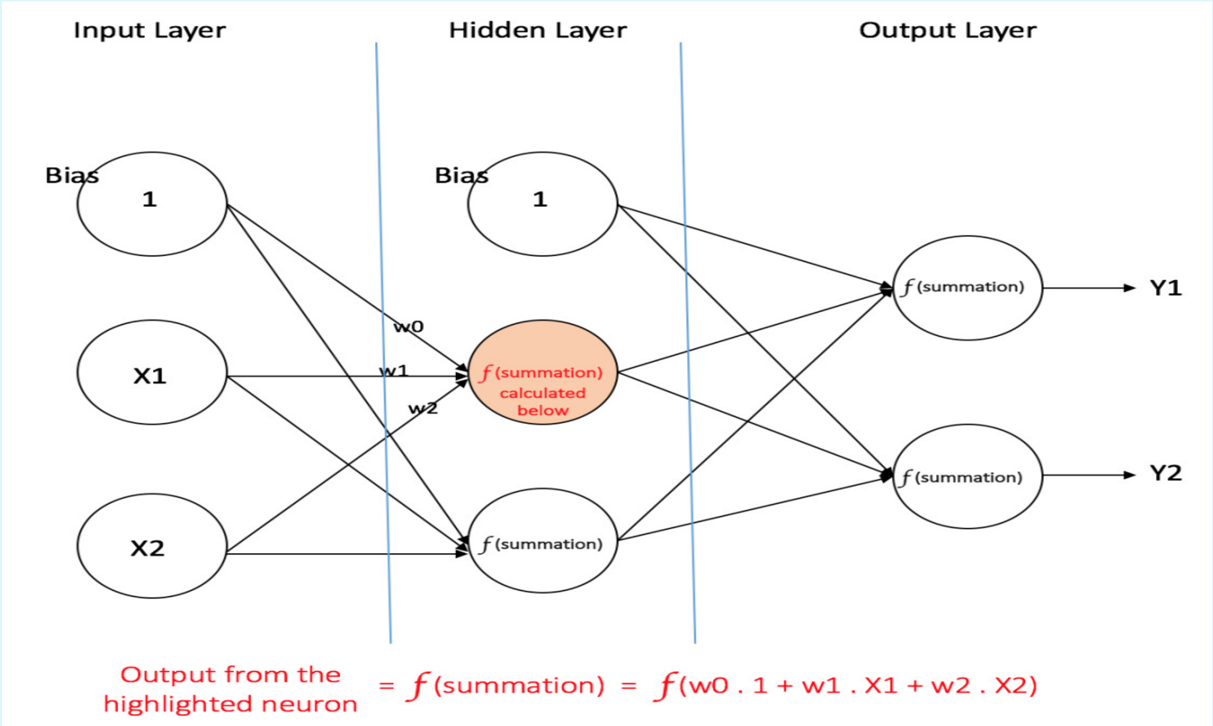
tanh: $\tanh(x) = 2\sigma(2x) - 1$

ReLU: Rectified Linear Unit. $f(x) = \max(0, x)$



3. Machine learning summary

- **3.7 Machine learning methods (II)**
- **Example 2 of ML: (Artificial) Neural Networks (ANN) – summary** (cont'd)
- **Feed-forward Neural Network**
  - Simplest type of ANN containing multiple neurons (nodes) arranged in layers
  - **Multi Layer Perceptron (MLP)** –has one or more hidden layers



Source: Ujjwalkarn, A Quick Introduction to Neural Networks, https://ujjwalkarn.me/2016/08/09/quick-intro-neural-networks/, 2016

3. Machine learning summary

- **3.7 Machine learning methods (II)**
- **Example 2 of ML: (Artificial) Neural Networks (ANN) – summary** (cont'd)
- **Feed-forward Neural Network**
  - Given a set of features X = (x1, x2, …) and a target y, a MLP can learn the relationship between the features and the target, e.g. for classification or regression.
- **Training the MLP: The Back-Propagation (BP) Algorithm**
  - BP of errors - is one way to train an ANN
  - BP is a supervised training scheme
  - Learning a function $f(\boldsymbol{X})$ to map given inputs $\boldsymbol{X}$ to desired outputs y
  - Training with 'labeled' data: each example input $\boldsymbol{X}(t)$ has a label $y(t)$ ('correct' output)
  - The error E between $f_t(\boldsymbol{X}(t))$ and $y(t)$ is used to adapt f
    - and compute $f_{t+1}(\boldsymbol{X})$
    - Method: gradient based adjustment of the perceptron weights to correct errors
  - **After training, one can use f for unlabeled data**

3. Machine learning summary

- **3.7 Machine learning methods (II)**
- **Example 2 of ML: (Artificial) Neural Networks (ANN) – summary** (cont'd)
- **Convolutional Neural Networks** (Deep NN)
  - **Goal: Increasing the NN running speed**
  - **Layers**
    - *Convolutional layers (CL)*
      - Every neuron has just a very limited number of inputs to the vicinity of a corresponding neuron in the previous layer
      - All neurons in a layer use the same set of weights
    - *Pooling layers (PL)*
      - Neighboring neurons are merged (max, sum, etc.)
  - The fully connected layer (MLP) at the end connects all split components of layers
  - Learning is performed by using back-propagation

- **Advantage:** large networks can be composed by using these building blocks

Source: J,Quittek, Artificial Intelligence in Network Operations and Management,
https://networking.ifip.org/2018/images/2018-IFIP-Networking/Keynote-III-J-Quittek-Slides.pdf

3. Machine learning summary

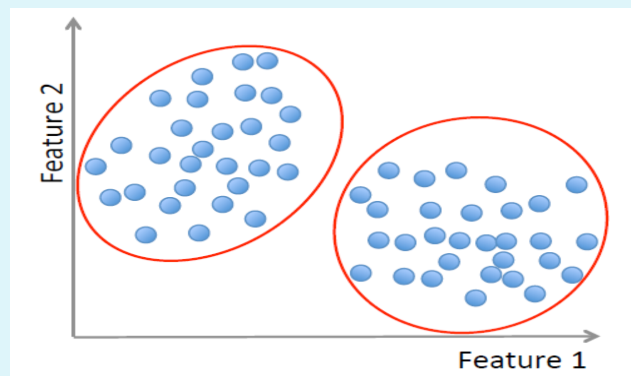
- **3.7 Machine learning methods (II)**
- **Unsupervised learning (UML)**
  - **Objective**: **to find a pattern in input data**
    - The goal is to construct representation of inputs that can be used for prediction on future inputs
    - An algorithm explores input data **without knowing an explicit output correct data** (e.g., explores demographic data to identify patterns)
    - **The training set is the unlabeled dataset**
    - Usage: at clustering, probability density estimation, finding association among features, dimensionality reduction

Training data: "examples" x.

$$x_1,\ldots,x_n,\ \ x_i \in X \subset \mathbb{R}^n$$

- Clustering/segmentation:

$$f : \mathbb{R}^d \longrightarrow \{C_1,\ldots C_k\} \text{ (set of clusters).}$$



Source: Machine Learning Basic Concepts https://courses.edx.org/asset-1:ColumbiaX+CSMM.101x+1T2017+type@asset+block@AI_edx_ml_5.1intro.pdf

3. Machine learning summary

- **3.7 Machine learning methods (II)**
- **Unsupervised Machine Learning (UML)** (cont'd)
- **Clustering**
  - Aims at **identifying groups** of data to build representation of the input
  - Methods to create clusters by grouping the data are: non-overlapping, hierarchical and overlapping clustering methods
  - **Algorithms: *K-means, Self-organizing Maps (SOMs), Fuzzy C-means, Gaussian mixture models***
- **Dimensionality Reduction**
  - Some problems need to reduce the dimension of the original data
  - Common methods to reduce the number of features in the dataset
    - **Feature Extraction (FE)** - e.g., **Principal component analysis (PCA)**
    - **Feature Selection (FS) -** e.g. **Sparse (SPCA)**
- **Anomaly Detection** - identifies events that do not correspond to an expected pattern. The machine selects the set of unusual events
  - Common methods: Rule based systems (similar to DTs); Pruning techniques: identify outliers, where there are errors in any combination of variables

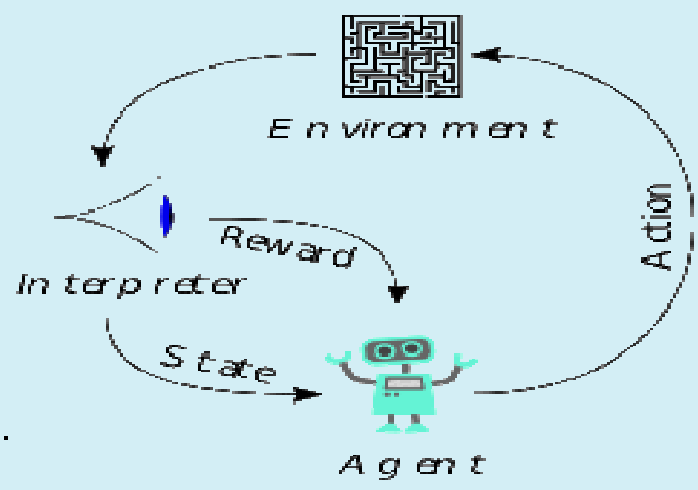
Source: Jessica Moysen and Lorenza Giupponi, "From 4G to 5G: Self-organized Network Management meets Machine Learning", arXiv:1707.09300v1 [cs.NI] 28 Jul 2017

3. Machine learning summary

- **3.7 Machine learning methods (II)**
- **Reinforcement learning (RL)-** general-purpose framework for decision-making
  - Elements: **Environment, Agent**
  - The **agent** is an entity that **perceives** and **acts** upon the **environment**
  - The environment changes due to the agent's actions and possibly other factors outside the agent's influence
  - The agent perceives the ***state (s)*** of the environment (a potentially incomplete observation), and must decide which ***action (a)*** to take based on that information, such that the accumulation of rewards (*r*) it receives from the environment is maximized
  - RL is also is called *approximate dynamic programming*, or *neuro-dynamic programming*

An RL algorithm consists of two phases:

Training phase: is to tell the agent which action shall be taken under a given environment from a series of trials.
Inference phase: the agent takes appropriate actions according to the experience learned during the training.



3. Machine learning summary

- **3.7 Machine learning methods (II)**
- **Reinforcement learning (RL**)- (cont'd)
- **RL advantages-drawbacks**
  - High Accuracy: RL-based solutions are data-driven
    - decisions are based on the profiling of the environment, without man-made assumptions or statistics.
  - RL requires lots of data and a long time to train
  - Once trained, an RL-based controller can make fast inferences
  - Ever evolving: the RL controller can still learn by itself from the effects of actual control.
    - It can self-adjust after facing new environments and accumulate new experiences to evolve
  - Easy implementation - as RL algorithms do not require solving complicated mathematical models and simply require obtaining the data needed.

See: D.Zeng, et al., "Resource Management at the Network Edge: A Deep Reinforcement Learning Approach", IEEE Network, May/June 2019,pp.26-33

3. Machine learning summary

- **3.7 Machine learning methods (II)**
- **Reinforcement learning (RL)** (cont'd)- math. definition
- The agent-environment interaction is typically formulated as a **Markov Decision Process (MDP)**, usable also in this context of dynamic programming techniques

- **MDP is a generalized framework** to model decision-making problems in cases where the result is partially random and affected by the applied decision

- **MDP : 5-tuple as MDP = <S,A, P(s′|s, a),R, γ>,**
  - **S** - finite state space, **A** action set
  - **P(s′|s, a)** - transition probability **[(s,a), at time t]--> [s' at time t+1]**
  - **R(s, a)** - immediate reward **after** performing the **action a** under **state s**
  - **γ ∈ [0, 1]** - discount factor to reflect the diminishing importance of current reward on future ones

- **MDP goal:** to find a policy **a = π(s)** , ***π : S → A***
  - **that determines what action a is selected under state s,**
    - so as to maximize the **value function, V** defined as the expected discounted cumulative reward by the Bellman equation:

3. Machine learning summary

- **3.7 Machine learning methods (II)**
- **Reinforcement learning (RL)** (cont'd)- math. Definition
- **Bellman equation:** expected discounted cumulative reward

$$V^{\pi}(\hat{s}) = E_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k R(s^{(k)}, \pi(s^{(k)}))|s^{(0)} = \hat{s})\right]$$
$$= E_{\pi}\left[R(\hat{s}, \pi(\hat{s}))) + \gamma \sum_{s' \in S} P(s'|\hat{s}, \pi(\hat{s}))V^{\pi}(s')\right]$$

- **MDP : 5-tuple as MDP = <S,A, P(s′|s, a),R, γ>,**
  - **S** - finite state space, **A** action set
  - **P(s′|s, a)** - transition probability **[(s,a), at time t]--> [s' at time t+1]**
  - **R(s, a)** - immediate reward **after** performing the **action a** under **state s**
  - **γ ∈ [0, 1]** - discount factor to reflect the diminishing importance of current reward on future ones
  - Policy **a = π(s) , *π : S* → *A***

3. Machine learning summary

- **3.7 Machine learning methods (II)**
- **Reinforcement learning (RL)** (cont'd)
  - P(s′|s, a) could be known apriori with no random factors
    - but (see control theory, etc.) RL aims to obtain the **optimal policy** π^* , under condition of unknown and partially random dynamics
  - RL does not have explicit knowledge on how close is to its goal: -->
    - a balance is needed between **exploring new potential actions** and **exploiting the already learnt experience**
  - Examples of classical RL algorithms:
    - **Q-learning**
    - **actor-critic method**
    - **State–action–reward–state–action (SARSA)**
    - **temporal-difference (TD) update**, etc.
  - **Classification** of the RL algorithms: various criteria
    - **Model-based** versus **Model-free**

3. Machine learning summary

- **3.7 Machine learning methods (II)**
- **Reinforcement learning (RL)** (cont'd)
- **Model based** versus **model free**
  - **Model-based**
    - the **agent tries to learn the model of how the environment works** from its observations and then plan a solution using that model
    - After the agent gains an accurate model, it can use **a planning algorithm with its learned model to find a policy**
  - **Model-free**
    - the agent
      - does not directly learn how to model the environment (e.g., **Q-learning**) but estimates the **Q-values** (or roughly the value function) of each (s,a) pair
      - and derives the optimal policy by choosing the action yielding the *largest Q-value in the current state*
      - the model-free algorithm like Q-learning cannot predict the next state and value before taking the action

3. Machine learning summary

- **3.7 Machine learning methods (II)**
- **Reinforcement learning (RL)** (cont'd)

- **Classification** of the RL algorithms- various criteria (cont'd)
  - Two ways to **update** the **value function**
  - **Monte-Carlo**:
    - the agent updates its estimation for a pair (s,a), by calculating the mean return from a collection of episodes

  - **TD update** : it approximates the estimation by comparing estimates at two consecutive episodes
    - **Example**: TD update for Q-learning
      - The Q-value is updated as
      - $Q(s,a) \leftarrow Q(s,a) + \alpha(R(s,a) + \gamma \max_{a'} Q(s',a') - Q(s, a))$
        - where α is the learning rate.
      - The term $R(s, a) + \gamma \max_{a'} Q(s',a') - Q(s,a)$ is the TD error
        - (difference between the current (sampled) estimate $R(s,a) + \gamma \max_{a'} Q(s', a')$ and previous one Q(s,a)

3. Machine learning summary

- **3.7 Machine learning methods (II)**
- **Reinforcement learning (RL)** (cont'd)

- **Classification** of the RL algorithms- various criteria
  - **On-policy** versus **Off-policy** - depending on coupling of the function update and the update policy executed
  - Before updating the value function, the agent also needs to sample and learn the environment by performing some non-optimal policy
    - **Off policy update**: If the update policy is irrelevant to the sampling policy
    - **On-policy update**:
      - $Q(s,a) \leftarrow Q(s,a) + \alpha(R(s, a) + \gamma Q(s',a') - Q(s,a))$
      - where a′ and a need to be chosen according to the same policy

3. Machine learning summary

- **3.7 Machine learning methods (II)**

- **Deep Learning (DL)**
- General-purpose framework for *representation* learning
  - Given an objective
    - Learn representation that is required to achieve objective
    - Directly from raw inputs, using minimal domain knowledge

  - **DL use a cascade of *multiple layers* of nonlinear processing units (e.g ANN MLP)** for *feature extraction* and *transformation*. Each successive layer has inputs from the previous layer
  - learn in supervised (e.g., classification) and/or unsupervised (e.g., pattern analysis) modes
  - learn multiple levels of representations that correspond to different levels of abstraction; the levels form a hierarchy of concepts

- **Deep Reinforcement Learning**: **DRL = RL + DL (**general intelligence)
  - **RL** defines the **objective**; **DL gives the mechanism**

3. Machine learning summary

3.8 Machine Learning (ML) – summary (cont'd)

Supervised Learning	
Classification	K-Nearest Neighbours Generalized Linear Model Support Vector Machines Naive Bayes Neural Networks
Regression	K-Nearest Neighbours Generalized Linear Regression Support Vector Regression Neural Networks Decision Trees
Unsupervised Learning	
Clustering	Non-overlapping clustering Hierarchical clustering Overlapping clustering
Dimensionality Reduction	Feature Extraction Feature Selection
Anomaly Detection	Pruning techniques Rule-based systems
Reinforcement Learning	
Model-based	Dynamic Programming Monte Carlo
Model-free	Temporal Difference

Source: Jessica Moysen and Lorenza Giupponi, "From 4G to 5G: Self-organized Network Management meets Machine Learning", arXiv:1707.09300v1 [cs.NI] 28 Jul 2017

3. Machine learning summary

- **3.9 Machine Learning algorithms –typical use cases**

- **Supervised learning** - used for
  - ***regression and classification*** *computation*
    - algorithms: k-NN, neural networks (NN), deep NN
  - ***classifications*** (additional)
    - algorithms:, Bayesian classifier, support vector machine (SVM)

- **Unsupervised learning-** used for
  - ***density estimation***
    - algorithms: Bolzmann machine, Kernel density, Gaussian mixtures
  - ***dimensionality reduction***
    - algorithms: auto-associative NN, local linear embedding (LLE)
  - ***clustering***
    - algorithms: Spectral clustering, K-means, Principal component analysis

- **Reinforcement learning** (RL) –used for **real-time decisions**

CONTENTS

4. Use cases examples

- **4.1 Network and services management functions - relevant (candidates) ML techniques**
- **Planning and design**
  - **Functions**:
    - Classification of service requirements
    - Forecasting trend; user behavior
    - Parameters Configuration
  - **ML techniques: *Support vector machine; Gradient boosting decision tree; Spectral clustering; Reinforcement learning***
- **Operation and management**
  - **Functions:**
    - Clustering cells, users, devices
    - Routing, forwarding, traffic control
    - Decision making for dynamic resource control, policy formulation
    - Reconfiguration of parameters
  - **ML techniques**: ***K-mean clustering; Deep neural network; Reinforcement learning***

4. Use cases examples

- **4.1 Network and services management functions - relevant (candidates) ML techniques** (cont'd)
- **Monitoring**
  - **Functions:** Clustering of syslog data; Classification of operation modes; Forecasting resource utilization trend
  - **ML techniques *: Spectral clustering; K-mean clustering; Support vector machine; Deep neural network***

- **Fault detection**
  - **Functions:** Classification of operation data; Detection of network anomaly; Predicting unusual behavior
  - **ML techniques: *Principal component analysis; Independent component analysis; Logistic regression; Bayesian networks***

- **Security**
  - **Functions:** Clustering users and devices; Detecting malicious behavior; Intrusion detection
  - **ML techniques: *Deep neural network; Principal component analysis***

4. Use cases examples

- **4.1 Network and services management functions - relevant (candidates) ML techniques**
  - **Specific ML techniques** appropriate for **FCAPS** - Examples

- BN - Bayesian networks
- NN - Neural networks
- K-NN – K Nearest Neighbors
- DT - Decision trees
- DL - Deep Learning
- SVM - Support vector machines
- DNN - Deep NN
- RL - Reinforcement Learning

Management area	Management function	Machine learning techniques
Fault	Fault prediction	NN, k-NN, k-Means, DT, BN, SVM
	Fault localization	NN, k-NN, k-Means, DT
	Automated mitigation	BN, SVM
Configuration	Adaptive resource allocation	Q-Learning, Deep
	Adaptive service configuration	Q-Learning
Accounting	–	–
Performance	Traffic load and metrics prediction	(Ensemble) NN, BN, SVM,
	QoE-QoS correlation	DT, BN, SVM, Q-learning
Security	Misuse detection	NN, DT, BN, SVM
	Anomaly detection	(Ensemble) NN, DNN, *k*-NN, *k*-means, (Ensemble) DT, Ensemble BN, SVM

Source: Sara Ayoubi, et.al., Machine Learning for Cognitive Network Management, IEEE Comm.Magazine , January 2018, pp.158-165

4. Use cases examples

4.2 Network applications examples and relevant ML techniques

Networking application		Steps of ML workflow					
General objective	**Specific works**	**Problem formulation**	**Data collection**		**Data analysis**	**Offline model construction**	**Deployment and online inference**
			Offline collection	**Online measurement**			
Traffic prediction	traffic volume prediction	SL: prediction with Hidden-Markov Model (HMM)	Synthetic and real traffic traces with flow statistics	Observe the flow statistics	The flow count and the traffic volume are correlated	Training HMM model with Kernel Bayes Rule and Recurrent Neural Network with Long Short Term Memory unit	Take flow statistics as input and obtain the output of the traffic volume
Traffic classification	traffic classification	SL and USL: clustering and classification	Labeled and unlabeled traffic traces	Flow statistical features extracted from traffic flows	Zero-day-application exists and may degrade the classification accuracy	Find the Zero-day-application class and training the classifier	Inference with the trained model to output the classification results
Resource management	job scheduling	RL: decision making with deep RL	Synthetic workload with different patterns is used for training	The real time resource demand of the arrival job	Action space is too large and may have conflicts between actions	Offline training to update the policy network	Directly schedule the arrival jobs with the trained model

Adapted from source (see details there): M.Wang, et al., "Machine Learning for Networking: Workflow, Advances and Opportunities", IEEE Network Vol.32, Issue 2 , March-April 2018

4. Use cases examples

4.2 Network applications examples and relevant ML techniques (cont'd)

Networking application		Steps of ML workflow					
General objective	**Specific works**	**Problem formulation**	**Data collection**		**Data analysis**	**Offline model construction**	**Deployment and online inference**
			Offline collection	**Online mon**			
Network adaptation	routing strategy	SL: decision making with *Deep Belief Architectures (DBA)*	Traffic patterns labeling with routing paths computed by OSPF protocol	Online traffic patterns in each router	Pb.: difficult to characterize the I/O patterns to reflect the dynamic nature of large-scale het. networks	Take the Layer-Wise training to initialize and the backpropagation process to fine-tune the DBA structure	Record and collect the traffic patterns in each router periodically and obtain the next routing nodes from the DBAs
	general QoE optimization	RL: decision making with a variant of UCB Upper confidence bound) algorithm	Session quality information with features in large time scale	Session quality information in small time scale	Application sessions sharing the same features can be grouped	Backend cluster determines the session groups using CFA [5] with a long time scale	Front-end performs the group-based exploration-exploitation strategy in real time
	TCP congestion control	RL: decision making with a tabular method	Collect experience from network simulator	Calculate network state variables with ACK	Select the most relevant metrics as state variables	Given network assumption the generated algorithm interact with simulator to learn best actions according to states	Directly implement the generated algorithm to corresponding network environment

Adapted from source (see details there): M.Wang, et al., "Machine Learning for Networking: Workflow, Advances and Opportunities", IEEE Network Vol.32, Issue 2 , March-April 2018

4. Use cases examples

4.2 Network applications examples and relevant ML techniques (cont'd)

Networking application		Steps of ML workflow					
General objective	**Specific works**	**Problem formulation**	**Data collection**		**Data analysis**	**Offline model construction**	**Deployment and online inference**
			Offline collection	**Online mon.**			
Perf prediction	Video QoE optimization	USL: clustering with self-designed algorithm	Datasets consisting of quality measurements are collected from public CDNs	Session features as input, e.g.: Bitrate, CDN, Player, etc.	Similar sessions are with similar quality determined by critical features	Critical feature learning in minutes scale and quality estimation in tens of seconds	Look up feature-quality table to respond to real-time query
Perf prediction	throughput prediction	SL: prediction with HMM	Datasets of HTTP throughput measurement from iQIYI	users' s session features as input	Sessions with similar features tend to behave in related pattern	Find set of critical feature and learn a HMM for each cluster of similar sessions	new session is mapped to the most similar session cluster and corresponding HMM are used to predict throughput

Adapted from source (see details there): M.Wang, et al., "Machine Learning for Networking: Workflow, Advances and Opportunities", IEEE Network Vol.32, Issue 2 , March-April 2018

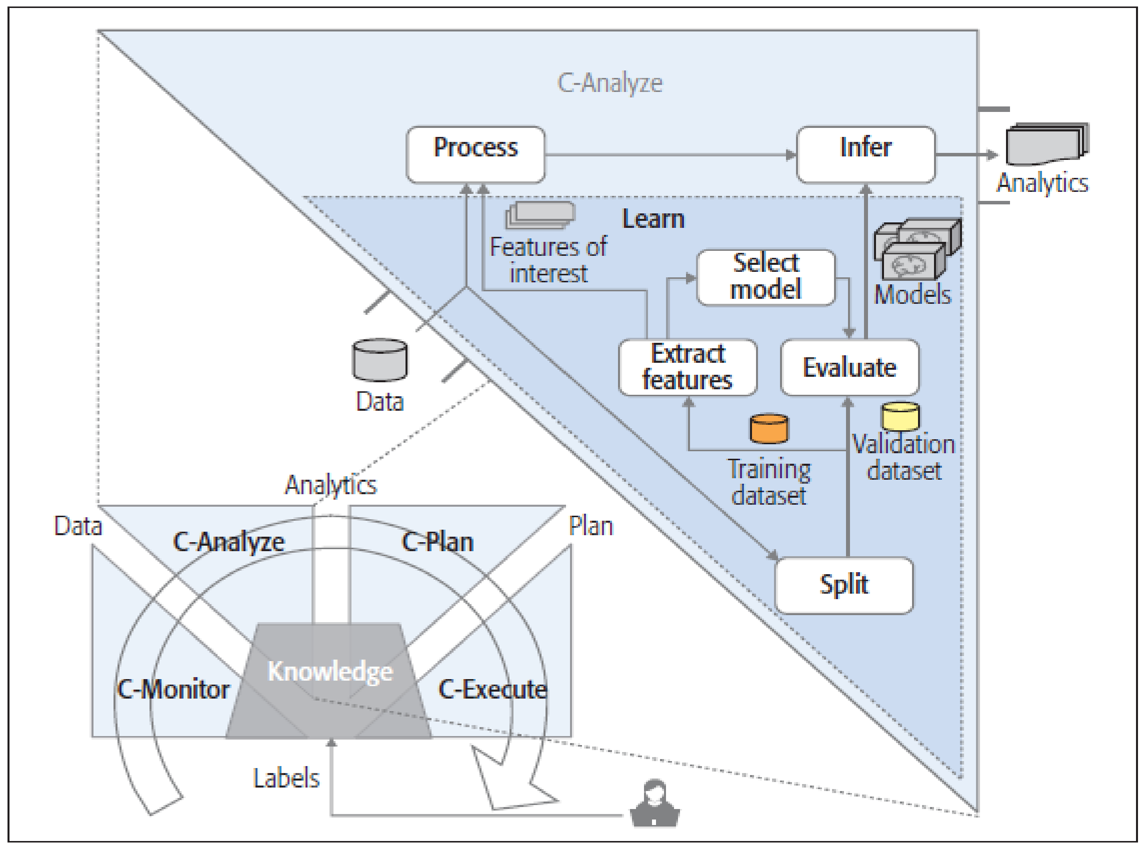
4. Use cases examples

4.3 Examples of management architectures with ML

- **Example 1: MAPE- full cognitive loop**

Source: Sara Ayoubi, et.al., Machine Learning for Cognitive Network Management, IEEE Comm.Magazine , January 2018, pp.158-165

- Traditional – MAPE: only Analyze Phase included cognitive properties
- **Proposal : to introduce ML in all phases**
- ML: introducing **learning** and **inference** in every function.



4. Use cases examples

4.3 Examples of management architectures with ML

- **Example 1: MAPE- full cognitive loop** (cont'd)
- **C-Monitor:** **intelligent probing** (e.g., if overloaded network the probing rate is reduced and instead perform regression for data prediction
- **C-Analyze**: **detects or predicts changes** in the network environment (e.g., faults, policy violations, frauds, low performance, attacks)
- **C-Plan: -use** ML to develop an **intelligent automated planning** (AP) engine reacting to changes in the network by selecting or composing a change plan
  - **Used in slice updates**
- **C-Execute:** schedules the generated plans; actions to be done in case that execution of a plan fail
  - **RL** is –naturally- applied: **C-Execute agent** could
    - exploit past successful experiences to generate optimal execution policies
    - explore new actions in case the execution plan fails

- **Closing the control loop** : monitoring the state of the network to measure the impact of the change plan

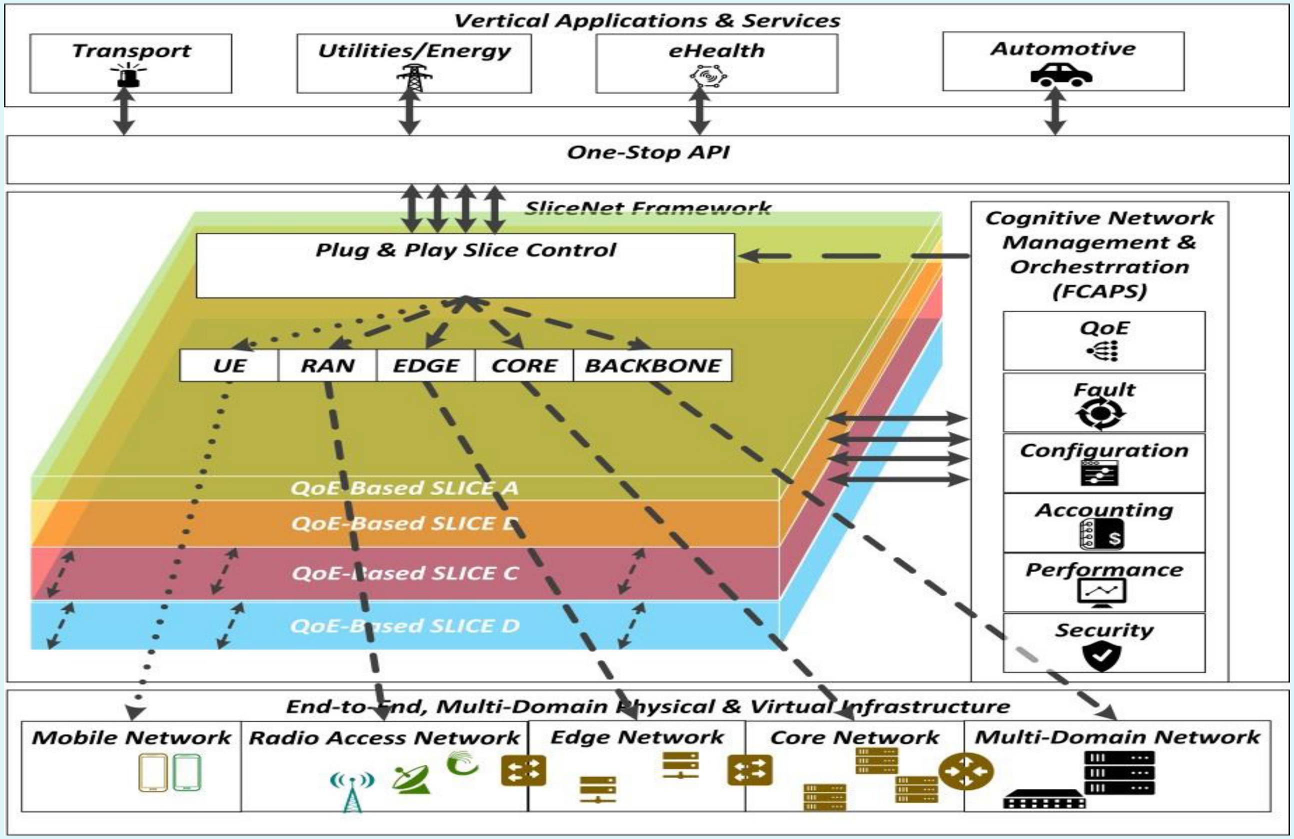
Source: Sara Ayoubi, et.al., Machine Learning for Cognitive Network Management, IEEE Comm.Magazine , January 2018, pp.158-165

4. Use cases examples

4.3 Examples of management architectures with ML

- **Example 2: SLICENET H2020 Phase 2 project** : *End-to-End Cognitive Network Slicing and Slice Management Framework in Virtualized Multi-Domain, Multi-Tenant 5G Networks. (2016)*

- *SLICENET architectural vision*



Source: https://5g-ppp.eu/slicenet

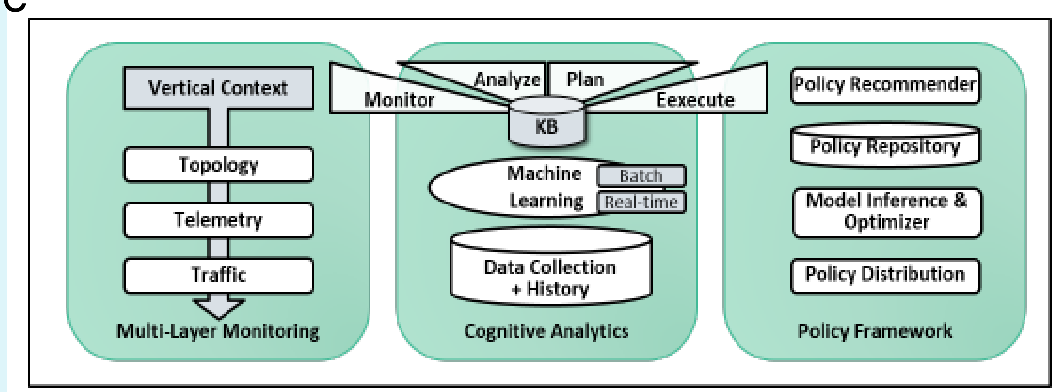
4. Use cases examples

4.3 Examples of management architectures with ML

- **Example 2: SLICENET H2020 Phase 2 project** (cont'd)
- SliceNet mgmt. : "verticals in the whole loop" approach, integrating the vertical perspective into the slice management process

- **SliceNet: fully automated slice M&O through AI/ML** and utilizing
  - the autonomous computing MAPE loop model
  - a vertically-informed multilayer QoE monitoring sub-plane
  - a slice-centric policy framework

- The vertical is integrated into the CogM process by providing
  - the perceived QoE: it enables *supervised ML* methods.
  - context: used to collect context-aware cross-layer information



Source: D. Lorenz, et. al., "SliceNet – Cognitive Slice Management Framework for Virtual Multi-Domain 5G Networks", https://www.systor.org/2018/pdf/systor18-21.pdf

4. Use cases examples

4.3 Examples of management architectures with ML

- **Example 3: 5G Network Slice Broker**
- **Problem solved**: *mapping heterogeneous service requirements onto the available network resource*
  - **5G Network Slice Broker (SB)-** mediator between external tenants and mobile network management
  - Slice management and SB should meet some 3GPP Slicing requirements
    - **Network Slice Templates** (NSTs) are available for different services
    - Each NST includes own SLAs
    - Broker: Receive NSL requests from tenants through a Network Exposure Function (NEF)
    - SB performs **Admission Control (AC)** based-on Slice Request NSTs
    - Use NG2 interfaces to monitor KPIs and configure network slice on RAN facilities

Source: V.Sciancalepore, K.Samdanis,et.al.,Mobile Traffic Forecasting for Maximizing 5G Network Slicing Resource Utilization, Infocom 2017, http://www.sciancalepore.info/files/infocom2017_ssc.pdf
J,Quittek, Artificial Intelligence in Network Operations and Management, https://networking.ifip.org/2018/images/2018-IFIP-Networking/Keynote-III-J-Quittek-Slides.pdf

4. Use cases examples

4.3 Examples of management architectures with ML

- **Example 3: 5G Network Slice Broker** (cont'd)
- **Slice Broker concepts and architecture**
  - NSL requests are collected within a fixed negotiation time window
  - then the time window is closed, and slice requests are processed

  - **Admission Control (AC)** is necessary (considering the available resources and SLA request
  - Prediction of the tenants' traffic evolution in the near future increases the efficiency

  - **Slice Forecasting Module (SFM)** analyzes the network slices traffic patterns and provides information to the AC
    - **Machine Learning can be used in SFM**
    - If no forecasting is applied or during the ML training period the only information used are the SLA requests

Source: V.Sciancalepore, K.Samdanis,et.al.,Mobile Traffic Forecasting for Maximizing 5G Network Slicing Resource Utilization, Infocom 2017, *http://www.sciancalepore.info/files/infocom2017_ssc.pdf*
J,Quittek, Artificial Intelligence in Network Operations and Management, *https://networking.ifip.org/2018/images/2018-IFIP-Networking/Keynote-III-J-Quittek-Slides.pdf*

4. Use cases examples

4.3 Examples of management architectures with ML

- **Example 3: 5G Network Slice Broker** (cont'd)

- **Slice Broker concepts and architecture** (cont'd)
  - Different AC policies and algorithms can be used to select which NSL requests will be granted for the next time window
  - The list of granted slice requests is sent to the **Slice Scheduling Module**
    - SSM allocates NSL physical resources and monitors (with a penalty history function) the served traffic levels and potential SLA violations

- Feedback is provided to the forecasting module for adaptive behaviour

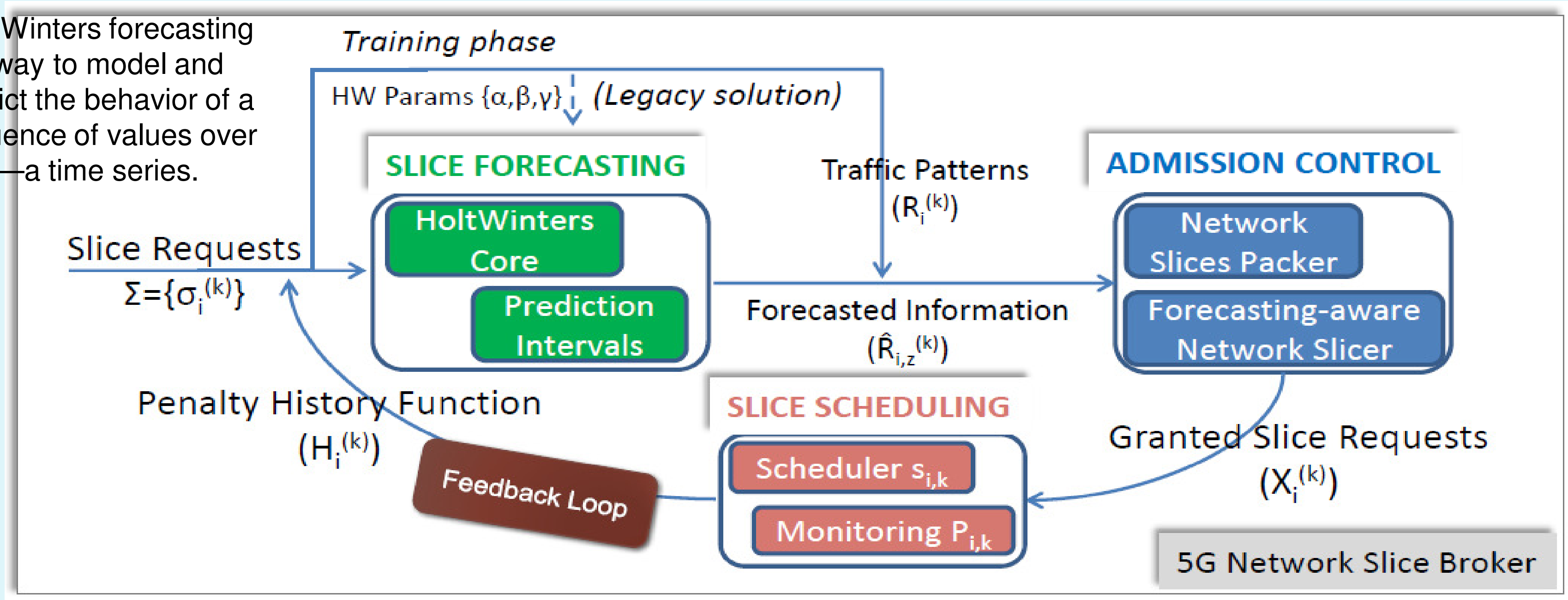
- **Slice Broker architecture -** functional blocks
    - Slice forecasting
    - Admission Control
    - Slice scheduling

Source: V.Sciancalepore, K.Samdanis,et.al.,Mobile Traffic Forecasting for Maximizing 5G Network Slicing Resource Utilization, Infocom 2017, http://www.sciancalepore.info/files/infocom2017_ssc.pdf
J,Quittek, Artificial Intelligence in Network Operations and Management, https://networking.ifip.org/2018/images/2018-IFIP-Networking/Keynote-III-J-Quittek-Slides.pdf

4.3 Examples of management architectures with ML

- **Example 3: 5G Network Slice Broker** (cont'd)
  - Slice Broker architecture
  - ML- applicable mainly for traffic forecasting (but not only)

Holt-Winters forecasting is a way to model and predict the behavior of a sequence of values over time—a time series.



Source: V.Sciancalepore, K.Samdanis,et.al.,Mobile Traffic Forecasting for Maximizing 5G Network Slicing Resource Utilization, Infocom 2017, http://www.sciancalepore.info/files/infocom2017_ssc.pdf
J,Quittek, Artificial Intelligence in Network Operations and Management, https://networking.ifip.org/2018/images/2018-IFIP-Networking/Keynote-III-J-Quittek-Slides.pdf

4. Use cases examples

4.3 Examples of management architectures with ML

- **Example 4: Resource management (RM) optimization for 5G slicing based on Deep Reinforcement Learning (e.g., DQL)**
- **General goal:** optimization for **Radio Resource (RR)** and **Virtualized Network Functions (VNF)** design in **Core network (CN)**
  - RAN and CN have different optimization goals
  - RM should consider several variables; a weighted summation of these variables can be defined as the reward for the learning agent
  - **Radio access optimization**: RM allocates resource blocks (RBs) to one slice, to get spectral efficiency (SE) – aiming to high rate and small delay
  - **Core network** (usually optical): design optimization of common or dedicated VNFs in order to realize good packet forwarding for each specific slice with minimal scheduling delay
  - Needed: balancing the relative importance of resource utilization (e.g, SE) and QoE satisfaction ratio

  - Solution: RM problem could be formulated as $\mathbf{R = \alpha \cdot SE + \beta \cdot QoE}$, where α and β denotes the importance of SE and QoE.

Source: R.Li, Z.Zhao, et al., "Deep Reinforcement Learning for Resource Management in Network Slicing", arXiv:1805.06591v3 [cs.NI] 21 Nov 2018

4. Use cases examples

4.3 Examples of management architectures with ML

- **Example 4: RM optimization for 5G slicing based on DQL** (cont'd)
  - **Equal or Prioritized Scheduling**
  - IETF has defined the **common control network function (CCNF)** to all or several slices
    - e.g.: access and mobility management function (AMF); network slice selection function (NSSF) (to select core network slice instances)
    - The CCNF might equally treating flows from different slices, or might differentiate flows
      - e.g., high priority for ultra-reliable low-latency communications (URLLC))
    - To balance the **resource utilization (RU)** and the **waiting time (WT)** of flows, the objective goal could be written as a weighted summation of RU and WT.

4. Use cases examples

4.3 Examples of management architectures with ML

- **Example 4: RM optimization for 5 network slicing based on DQL** (cont'd)
- **Radio Resource Slicing**
  - **How to apply DQL for RR slicing ?**

  - **Given**: a list of existing slices 1, • • • ,N sharing the aggregated bandwidth W and having fluctuating demands d = (d1, • • • , dN),
    - DQL tries to give a bandwidth sharing solution w = (w1, • • • ,wN), so as to **maximize the long-term reward expectation E{R(w,d)}**
      - the notation E(•) is the the expectation of the argument, that is,
      - $\arg_w \max E\{R(w, d)\} = \arg_w \max E\{\alpha \cdot SE(w,d) + \beta \cdot QoE(w,d)\}$

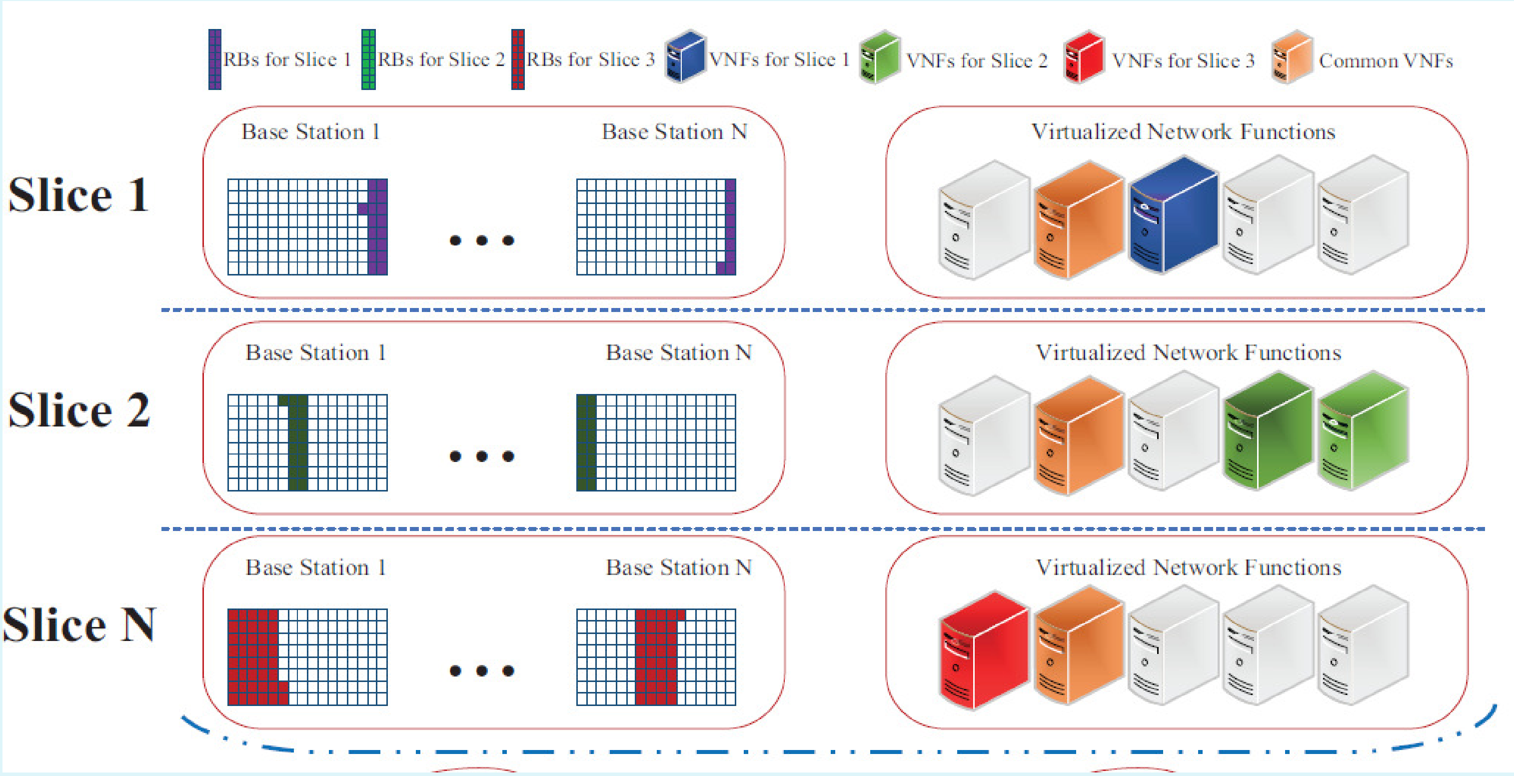
  - s.t.: $w = (w_1, \cdots ,w_N)$; $w_1 + \cdots + w_N = W$; $d = (d_1, \cdots , d_N)$
    - $d_i \sim$ Certain Traffic Model, $\forall i \in [1, \cdots ,N]$

  - The above problem challenge is the **volatile demand variations without having known a priori due** to the traffic model.
  - **DQL is the matching solution for this problem**

4. Use cases examples

4.3 Examples of management architectures with ML

- **Example 4: RM optimization for 5 network slicing based on DQL** (cont'd)
  - Resource management targets in RAN and Core network
    - RB= Resource block; VNF = Virtualized Network Function



Source: R.Li, Z.Zhao, et al., "Deep Reinforcement Learning for Resource Management in Network Slicing", arXiv:1805.06591v3 [cs.NI] 21 Nov 2018

4. Use cases examples

4.3 Examples of management architectures with ML

- **Example 4: RM optimization for 5 network slicing based on DQL** (cont'd)
- A summary of key settings in DQL for network slicing (simulation study)

Example: Mapping from Resource Management for Network Slicing to DRL		
	Radio Resources Slicing	**Priority –based Core Network slicing**
State	The number of arrived packets in each slice within a specific time window	The priority and time-stamp of last arrived five flows in each service function chain (SFC)
Action	Allocated bandwidth to each slice	Allocated SFC for the flow at current time-stamp
Reward	Weighted sum of SE and QoE in 3 sliced bands	Weighted sum of average time in 3 SFCs

Source: R.Li, Z.Zhao, et al., "Deep Reinforcement Learning for Resource Management in Network Slicing", arXiv:1805.06591v3 [cs.NI] 21 Nov 2018

4. Use cases examples

4.3 Examples of management architectures with ML

- **Example 4: RM optimization for 5 network slicing based on DQL** (cont'd)
- A summary of key settings in DQL for network slicing (simulation study)
- DQL uses the mapping shown in the Table to **optimize the weighted summation of system SE and slice QoE.** It is performed a round-robin scheduling method within each slice at the granularity of 0.5 ms.
- The system sequentially allocates the bandwidth of each slice to the active users within each slice every 0.5ms.
- Besides, the bandwidth allocation is adjusted to each slice per second.
- The DQL agent updates its Q-value neural network every second
- Details on the algorithm and results – in:
  - *Source: R.Li, Z.Zhao, et al., "Deep Reinforcement Learning for Resource Management in Network Slicing", arXiv:1805.06591v3 [cs.NI] 21 Nov 2018*

4. Use cases examples

4.3 Examples of management architectures with ML

- **Example 5: A model-free DRL approach for RM at the network edge**
  - *Source: D.Zeng, et al., "Resource Management at the Network Edge: A Deep Reinforcement Learning Approach", IEEE Network, May/June 2019, pp.26-33*
  - **Edge computing** (alternative to centralized CC) process tasks by exploring various resources at the network edge
  - **Edge servers** (e.g., servers with cellular base stations) are geo-distributed in different locations, enabling performance/cost improvement by processing data close to their sources
  - **Problem : mobility-aware service migration and energy control in edge computing and design a DRL-based algorithm** to deal with the network dynamics, aiming at operational cost minimization
    - **Possible solution: to host latency-sensitive services, in the form of virtual machines (VMs), close to the data sources**
    - A model-free RL approach, edge computing RM framework
      - It does not require any prior knowledge on the network dynamics or statistics
      - It can automatically learn the network dynamics and make appropriate control decisions accordingly at runtime

4. Use cases examples

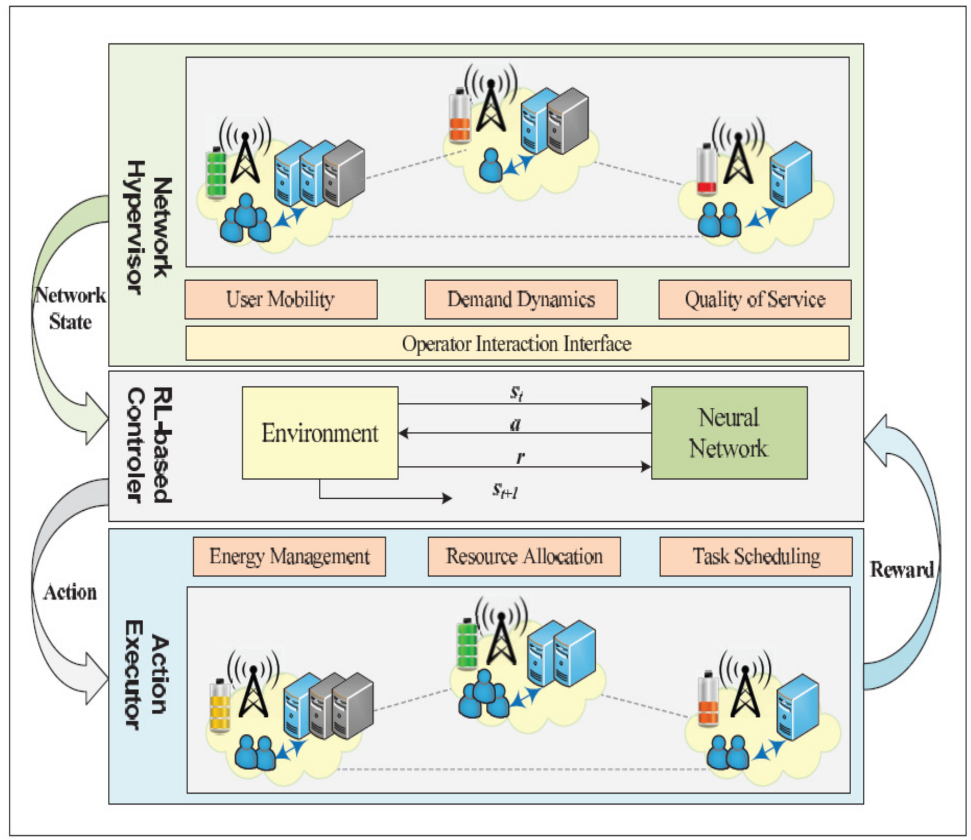
4.3 Examples of management architectures with ML

- Example 5: A model-free DRL approach for RM at the network edge
- The RL-based edge RM framework

▪**Network hypervisor (HV) : monitors and aggregates the states** of the elements in the EC environment (e.g., user locations, server workloads, statuses of: service placement, spectrum allocation, BS power; link bandwidth consumption, electricity prices, user association

  ▪This **info is needed for an RL-based controller** to profile the environment so as to make control decisions

▪The HV provides a programmable interface to the network operator for handling the interaction (network operator – RL_based controller)



4. Use cases examples

4.3 Examples of management architectures with ML

- **Example 5: A model-free DRL approach for RM at the network edge**
- **The RL-based edge RM framework** (cont'd)
  - **The RL-based controller (RL-C)**:
    - **runs a RL/DRL algorithms;**
    - requires lots of data to train (not manually designed statistics but simply data required)
    - learns (from the history and look into the future) according to the objective set (e.g., improving the QoS, lowering the OPEX, reducing power consumption).
    - It is not recommended to train the RL-C online , in order to avoid unstable and bad decisions during the initial learning
      - One may apply historical data and simulator/emulator to train an early-version controller offline
  - Once well trained, the RL-C can take decisions based on the rt environment state and its accumulated experience
  - The RL-C becomes smarter while working online

4. Use cases examples

4.3 Examples of management architectures with ML

- **Example 5:A model-free DRL approach for RM at the network edge**
- **The RL-based edge RM framework** (cont'd)
  - **The action executor**:
    - installed in each controllable EC element (e.g., edge server, base-station, user equipment, router, and so on)
    - communicates with the RL-C to obtain decisions, and accordingly execute the derived action
    - once the action is taken, the executor calculates the reward obtained on each network node and reports it back to the RL-C to update the control agent so as to make it more intelligent and efficient

4. Use cases examples

4.3 Examples of management architectures with ML

- **Example 5: A model-free DRL approach for RM at the network edge**
- **What type of RL-controller?**
- The basic RL solution is Q-learning (QL),which chooses actions according to the Q-values stored in its two-dimensional Q-table.
- QL can solve some control problem with discrete actions
- QL: powerful, simple, but weak in generality and scalability
  - As the states (the number of edge servers, the number of users, the number of edge services, etc.) rise, the Q-table may exponentially increase.
- **Solution for scalability** : deep Q network (DQN) which introduces a neural network to estimate the Q-value function
  - DQN can be applied to the large-scale problem
- Note: Neither QL nor DQN can cope with continuous problems
  - Hence, **deep deterministic policy gradient (DDPG), (advanced DRL)** employs the actor-critic model and catreat continuous control problems, e.g.: workload balancing, task offloading, resource allocation, traffic optimization, and energy scheduling

CONTENTS

5. Conclusions and research challenges

- **Challenges in Using Machine Learning**
  - Representative Datasets
  - Speed vs. accuracy
  - Ground truth (refers to the accuracy of the training set's classification for SML techniques)
  - Incremental Learning
  - Scalability issues
  - Security of Machine Learning

- **Challenges in Autonomic Network Management in cognitive context**
  - Orchestration of Cognitive Management Functions
  - Cooperation between Cognitive Mgmt and SDN, NFV environment
  - Selection of the most convenient ML techniques for 5G M&C

- Thank you !
- Questions?

References

1. Panwar N., Sharma S., Singh A. K. 'A Survey on 5G: The Next Generation of Mobile Communication' Elsevier Physical Communication, 4 Nov 2015, http://arxiv.org/pdf/1511.01643v1.pdf
2. 5GPPP Architecture Working Group , View on 5G Architecture, Version 2.0, December 2017
3. 3GPP TS 23.501 V15.2.0 (2018-06), System Architecture for the 5G System; Stage 2, (Release 15)
4. 3GPP TS 28.530 V1.2.1 (2018-07), Management and orchestration of 5G networks; Concepts, use cases and requirements
5. 3GPP, TR 23.799 V14.0.0 (2016-12), Study on Architecture for Next Generation System (Release 14)
6. 3GPP TR 28.801 V15.1.0 (2018-01), Study on management and orchestration of network slicing for next generation network, (Release 15)
7. A.Galis, "Network Slicing - Management Challenges", IRTF 102 – NMRG, 2018
8. J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J.J. Ramos-Munoz, J. Lorca, J. Folgueira, Network "Slicing for 5G with SDN/NFV: Concepts, Architectures and Challenges" , IEEE Communications Magazine, 2017, Citation information: DOI 10.1109/MCOM.2017.1600935
9. ETSI GS NFV 003 V1.2.1 (2014-12), Network Functions Virtualisation (NFV);Terminology for Main Concepts in NFV, http://www.etsi.org/deliver/etsi_gs/NFV/001_099/003/01.02.01_60/gs_NFV003v010201p.pdf
10. ETSI GS NFV 002 v1.2.1 2014-12, NFV Architectural Framework
11. ETSI GR NFV-EVE 012 V3.1.1 (2017-12), Release 3; NFV Evolution and Ecosystem; Report on Network Slicing Support with ETSI NFV Architecture Framework

References

12. ETSI GR NFV-IFA 022 V3.1.1 (2018-04) Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Report on Management and connectivity for Multi-Site Services
13. Q. Li, G. Wu, A. Papathanassiou, U. Mukherjee , An end-to-end network slicing framework for 5G wireless communication systems, https://arxiv.org/abs/1608.00572
14. 5G Americas, Network Slicing for 5G Networks & Services, 2016, http://www.5gamericas.org/files/3214/7975/0104/5G_Americas_Network_Slicing_11.21_Final.pdf
15. K.Katsalis, N.Nikaein, Andy Edmonds, Multi-Domain Orchestration for NFV: Challenges and Research Directions, 2016 15th International Conference on Ubiquitous Computing and Communications and 2016 International Symposium on Cyberspace and Security (IUCC-CSS), https://ieeexplore.ieee.org/document/7828601
16. X. Foukas, G. Patounas, A.Elmokashfi, and M.K. Marina, Network Slicing in 5G: Survey and Challenges, IEEE Communications Magazine, May 2017, pp.94-100
17. P. Rost et al, Network Slicing to Enable Scalability and Flexibility in 5G Mobile Networks, IEEE Communication Magazine, 2017
18. G. Nencioni et al., Orchestration and Control in Software-Defined 5G Networks: Research Challenges, Wiley, Wireless Communications and Mobile Computing Volume 2018, Article ID 6923867, pp. 1-19, https://doi.org/10.1155/2018/6923867https://www.hindawi.com/journals/wcmc/2018/6923867/
19. Jinjin Gong, Lu Ge, Xin Su, Jie Zeng, Radio Access Network Slicing in 5G, in Advances in Intelligent Systems and Computing · March 2017 , https://www.researchgate.net/publication/315848047
20. Ibrahim AFOLABI, et al., Towards 5G Network Slicing over Multiple-Domains, IEICE Trans. Commun., Vol.E100–B, No.11 Nov. 2017

References

21. L. Geng , et.al., IETF- “Network Slicing Architecture draft-geng-netslices-architecture-02”, 2017
22. A.Galis and K.Makhijani, Network Slicing Landscape: A holistic architectural approach, orchestration and management with applicbility in mobile and fixed networks and clouds, v1.0, Network Slicing Tutorial – IEEE NetSoft 2018 – Montreal 29th June2018.
23. Ibrahim Afolabi, Tarik Taleb , Konstantinos Samdanis, Adlen Ksentini, and Hannu Flinck,Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies, and Solutions IEEE Comm. Surveys & Tutorials, Vol. 20, No. 3, 2018
24. T.Taleb, I.Afolabi, K.Samdanis, and F. Z.Yousaf, "On Multi-domain Network Slicing Orchestration Architecture & Federated Resource Control", http://mosaic-lab.org/uploads/papers/3f772f2d-9e0f-4329-9298-aae4ef8ded65.pdf
25. 5GPPP Network Management & Quality of Service Working Group, “Cognitive Network Management for 5G”, 2017
26. V. P. Kafle, et. al., “Consideration on Automation of 5G Network slicing with Machine Learning” , ITU Caleidoscope Santafe 2018
27. “Machine Learning Tutorial for Beginners”, https://www.guru99.com/machine-learning-tutorial.html
28. Machine Learning Basic Concepts https://courses.edx.org/asset-1:ColumbiaX+CSMM.101x+1T2017+type@asset+block@AI_edx_ml_5.1intro.pdf
29. Wei-Lun Chao, Machine Learning Tutorial, 2011, http://disp.ee.ntu.edu.tw/~pujols/Machine%20Learning%20Tutorial.pdf
30. Jessica Moysen and Lorenza Giupponi, “From 4G to 5G: Self-organized Network Management meets Machine Learning”, arXiv:1707.09300v1 [cs.NI] 28 Jul 2017
31. Sara Ayoubi, et.al., Machine Learning for Cognitive Network Management, IEEE Comm.Magazine , January 2018, pp.158-165
32. D. Lorenz, et. al., “SliceNet – Cognitive Slice Management Framework for Virtual Multi-Domain 5G Networks”, https://www.systor.org/2018/pdf/systor18-21.pdf

References

33. S. Shalev-Shwartz and S.Ben-David, Understanding Machine Learning: From Theory to Algorithms 2014, Cambridge University Press
34. N.Dupuis, K.Lee, Introduction to Machine Learning, https://www.phusewiki.org/docs/Frankfut%20Connect%202018/ML/Papers/ML17-phuse-paper-ml01-19265.pdf
35. J,Quittek, Artificial Intelligence in Network Operations and Management, https://networking.ifip.org/2018/images/2018-IFIP-Networking/Keynote-III-J-Quittek-Slides.pdf
36. R.Johansson, Applied Machine Learning Lecture 1: Introduction, Univ. of Gothenburg, 2019, http://www.cse.chalmers.se/~richajo/dit866/lectures/l1/l1.pdf
37. Ujjwalkarn, A Quick Introduction to Neural Networks, https://ujjwalkarn.me/2016/08/09/quick-intro-neural-networks/, 2016
38. V.Sciancalepore, K.Samdanis,et.al.,Mobile Traffic Forecasting for Maximizing 5G Network Slicing Resource Utilization, Infocom 2017, http://www.sciancalepore.info/files/infocom2017_ssc.pdf
39. M.Wang, et al., "Machine Learning for Networking: Workflow, Advances and Opportunities", IEEE Network Vol.32, Issue 2 , March-April 2018
40. Gorka Vélez, et al., “Machine Learning for Autonomic Network Management in a Connected Cars Scenario”, Communication Technologies for Vehicles, Springer 2016, pp 111-120, https://link.springer.com/chapter/10.1007/978-3-319-38921-9_12
41. R.Li, Z.Zhao, et al., "Deep Reinforcement Learning for Resource Management in Network Slicing", arXiv:1805.06591v3 [cs.NI] 21 Nov 2018
42. A.Nowe and T.Brys, “A Gentle Introduction to Reinforcement Learning”, Springer International Publishing Switzerland 2016, DOI: 10.1007/978-3-319-45856-4 2
43. D.Zeng, et al., "Resource Management at the Network Edge: A Deep Reinforcement Learning Approach", IEEE Network, May/June 2019,pp.26-33

Machine Learning Techniques in Advanced Network and Services Management

- **List of Acronyms**

5G CN	Core Network
5G-AN	5G Access Network
5GS	5G System
AF	Application Function
AI	Artificial Intelligence
AMF	Access and Mobility Management Function
AS	Access Stratum
BN	Bayesian Networks
CA	Certificate Authority
CaaS	Cooperation as a Service
CC	Cloud Computing
CP	Control Plane
CRAN	Cloud based Radio Access Network
D2D	Device to Device communication
DL	Deep Learning
DN	Data Network
DNN	Deep Neural Network
DoS	Denial of Services
DP	Data Plane (User Plane UP)

Machine Learning Techniques in Advanced Network and Services Management

- **List of Acronyms**

DT	Decision Tree
ENaaS	Entertainment as a Service
ePDG	evolved Packet Data Gateway
FC	Fog Computing
IaaS	Infrastructure as a Service
INaaS	Information as a Service
IoT	Internet of Things
IT&C	Information Technology and Communications
k-NN	k-Nearest Neighbours
LADN	Local Area Data Network
LLC	Logical Link Control
LMF	Location Management Function
MANET	Mobile Ad hoc Network
M&C	Management and Control
MEC	Multi-access (Mobile) Edge Computing
ML	Machine Learning
N3IWF	Non-3GPP InterWorking Function
NaaS	Network as a Service
NAI	Network Access Identifier

List of Acronyms

NEF	Network Exposure Function
NF	Network Function
NFV	Network Function Virtualization
NN	Neural Networks
NSL	Network Slice
NSLI	Network Slice Instance
NS	Network Service
NSLID	Network Slice Instance Identifier
NSSAI	Network Slice Selection Assistance Information
NSSF	Network Slice Selection Function
NSSP	Network Slice Selection Policy
NWDAF	Network Data Analytics Function
OIF	Optical Internetworking Forum
ONF	Open Networking Foundation
PaaS	Platform as a Service
PCF	Policy Control Function
PKI	Public Key Infrastructure
QoE	Quality of Experience
RAN	Radio Access Network

Machine Learning Techniques in Advanced Network and Services Management

- **List of Acronyms**

RL	Reinforcement Learning
SaaS	Software as a Service
SD	Slice Differentiator
SDN	Software Defined Networking
SLA	Service Level Agreement
SM	Service Management
SMF	Session Management Function
SML	Supervised Machine Learning
S-MIB	Security Management Information Base
SMSF	Short Message Service Function
S-NSSAI	Single Network Slice Selection Assistance Information
SSC	Session and Service Continuity
SST	Slice/Service Type
SVM	Support Vector Machine
TNL	Transport Network Layer
TNLA	Transport Network Layer Association
TSP	Traffic Steering Policy

- **List of Acronyms**

UDM	Unified Data Management
UDR	Unified Data Repository
UML	Unsupervised Machine Learning
UPF	User Plane Function
V2X	Vehicle-to-everything
VANET	Vehicular Ad hoc Network
VLAN	Virtual Local Area Network
VM	Virtual Machine
WAT	Wireless Access Technologies
WSN	Wireless Sensor Network

- **Backup slides**

3. Machine learning summary

- **3.7 Machine learning methods (II)**
- **Reinforcement learning (RL)** (cont'd)
- **MDP - more general mathematical definition**
- MDP is a tuple **<S, A, T, γ, R>**
  - **S** = {s1, s2, . . .} is the *possibly infinite* set of states the environment can be in
  - **A** = {a1, a2, . . .} is the *possibly infinite* set of actions the agent can take
  - **T(s'|s, a)** , **(or P(s'|s,a))** is the *probability* of ending up in environment state s' after taking action a in state s
  - **γ** ∈ **[0, 1]** is the discount factor; it defines how important future rewards are
  - **R(s, a, s')** is the possibly stochastic reward given for a state transition from s to s' through taking action a
    - It **defines the goal of an agent** interacting with the MDP; it indicates the immediate quality of what the agent is doing
- RL is modelled as MDP, since the state signal is assumed to have the **Markov property** (of a stochastic process):
  - the conditional probability distribution of future states of the process (conditional on both past and present states) ***depends only upon the present state*****,** not on the sequence of events that preceded it

See also: A.Nowe and T.Brys, "A Gentle Introduction to Reinforcement Learning", Springer International Publishing Switzerland 2016, DOI: 10.1007/978-3-319-45856-4 2

3. Machine learning summary

- **3.7 Machine learning methods (II)**
- **Reinforcement learning (RL)** (cont'd)
- **Q-Learning details**
  - It is a **model-free, TD update, off-policy RL algorithm**
  - Three major steps:

  - 1) The **agent chooses an action a** under state **s** according to some policy like e-greedy
    - (i.e., the agent chooses the action with the **largest Q-value** Q(s, a) with a probability of **e**, and equally chooses the other actions with a probability of **[1–e]**/|**A**| ; |A| = size of the action space

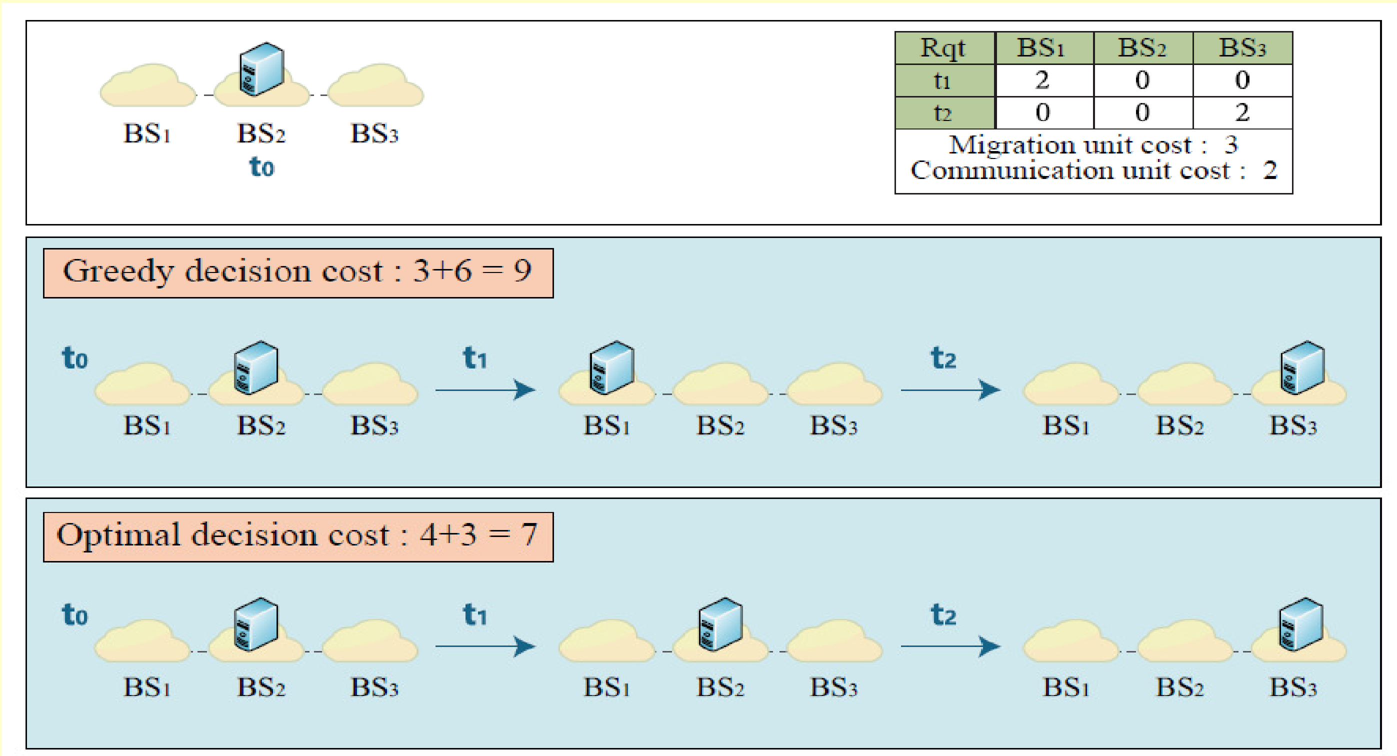
  - 2) The **agent learns the reward R(s,a)** from the environment
    - transitions s-> s′ is performed

  - 3) The **agent updates the Q-value function** in a TD manner as
    - $Q(s, a) \leftarrow Q(s, a) + \alpha\ (R(s, a) + \gamma\ \text{maxa}'Q(s, a) – Q(s,a))$.

4. Use cases examples

4.3 Examples of management architectures with ML

- **Example 5: A model-free DRL approach for RM at the network edge**



Source: D.Zeng, et al., "Resource Management at the Network Edge: A Deep Reinforcement Learning Approach", IEEE Network, May/June 2019, pp.26-33