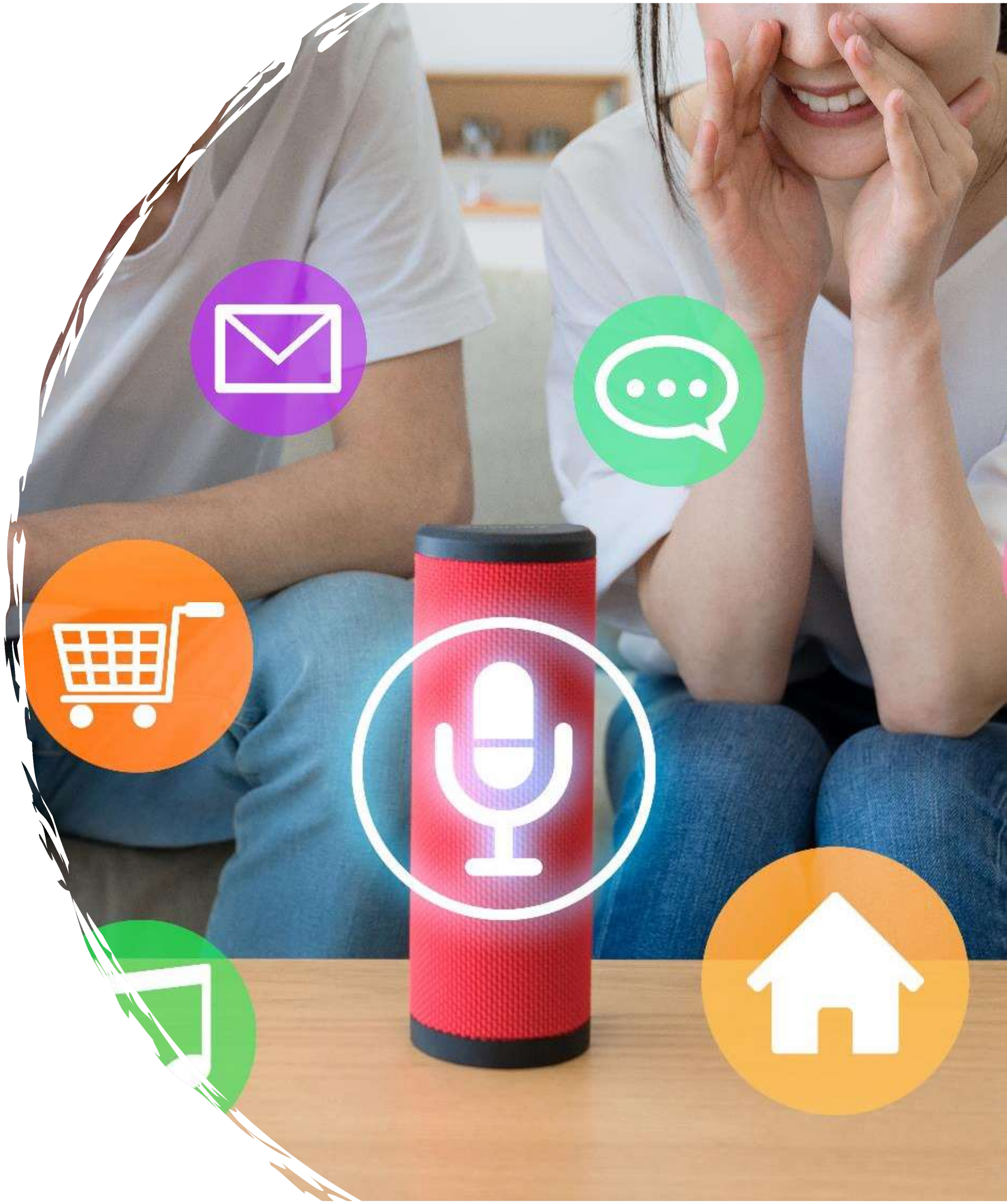
STQ Workshop

A View on ML for Performance, QoS, and QoE Evaluation

Presented by: Dr. Irina Cotanis, Technology Director, Office of the CTO, PM



22/11/2022

infovista

Agenda

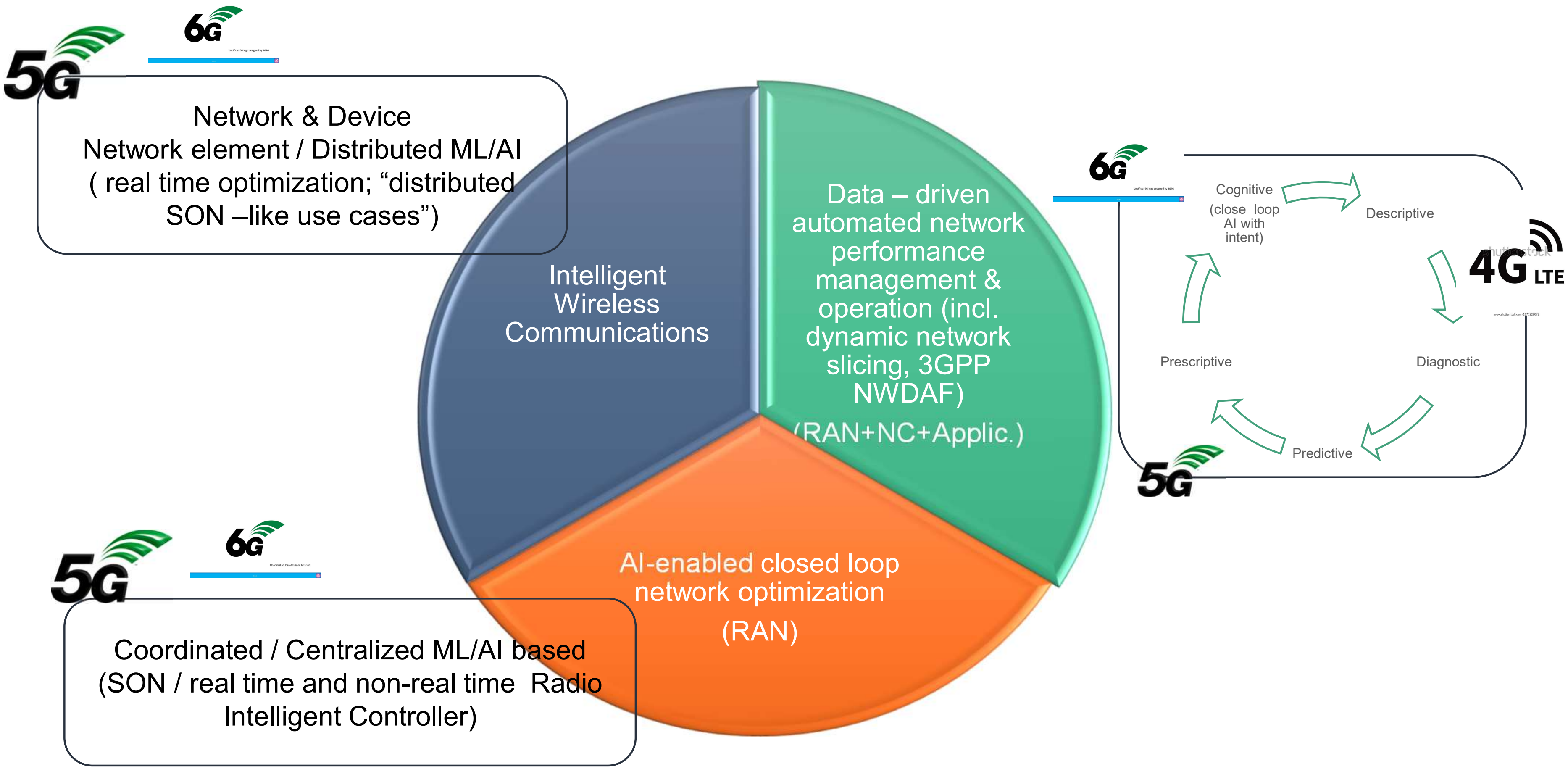
1. 5G call to action on using ML/AI techniques
2. ML algorithms and best practices for ITU-T SG12 use cases
3. Aspects of ML optimization: learning and validation
4. A glimpse to ML/AI standardization work within the context of mobile networks
5. Take away

5G call to action on using ML/AI techniques

infovista

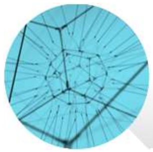
KNOW YOUR NETWORK™

The new mobile networks



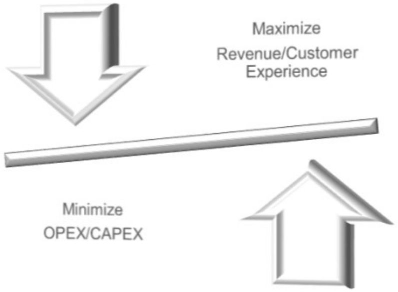
New emerging testing requirements and their derived trends

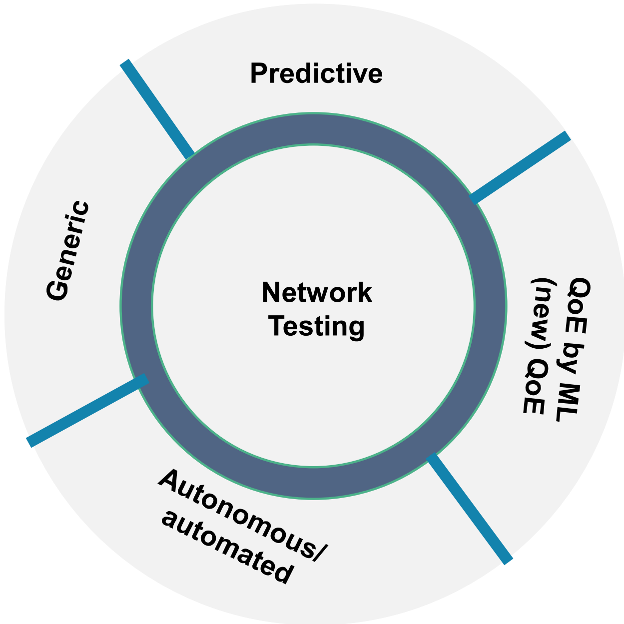
Cope with users/devices/services/ applications variety, diversity, dynamicity



Cope with network metrics complex inter-dependencies, new QoS/QoE dimensions

Cope with network embedded intelligence





ITU-T SG 12 call to action on ML based testing techniques

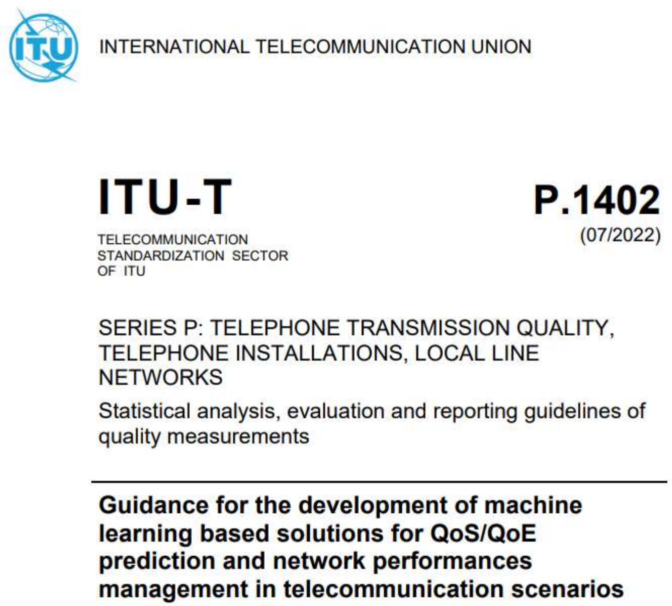
Mobile network evaluation → Testing requirements → AI/ML based testing techniques

- ML/AI algorithms to be applied for
  - Networks' performance evaluation, monitoring and troubleshooting techniques (overall testing)
  - Voice/video QoS/QoE prediction models

- Powerful techniques which inherently are very complex and therefore prone to misusage and misinterpretation and consequently showing high risks of drastically impacting their strengths and benefits.

- Need to carefully follow well defined guidelines when applying ML.

== >>> P.1402 recommendation introducing general guidelines for applying ML within the context of SG 12 work items which are suitable to these techniques.


INTERNATIONAL TELECOMMUNICATION UNION

ITU-T **P.1402**

TELECOMMUNICATION STANDARDIZATION SECTOR OF ITU

(07/2022)

SERIES P: TELEPHONE TRANSMISSION QUALITY, TELEPHONE INSTALLATIONS, LOCAL LINE NETWORKS

Statistical analysis, evaluation and reporting guidelines of quality measurements

Guidance for the development of machine learning based solutions for QoS/QoE prediction and network performances management in telecommunication scenarios

ITU-T P.1402: Guide for Development of Machine Learning Based Solutions

- General practices for applied ML
- Brief overview on ML
- Applied ML in ITU-T SG12
- Minimum requirements for ML based solutions
- Overview on ML optimization process
- Guidance on the evaluation and validation of ML based solutions in ITU-T SG12

ML algorithms and best practices for ITU-T SG12 use cases

infovista

KNOW YOUR NETWORK™

Most used ML algorithms applied within SG12 context

Linear regression / classifier (sML)

- When: thousand of features available
- Performance: less prone to overfitting

PCA (sML)

- When: wide range of features highly correlated, reduce dimens. with min info loss
- Performance: Easily overfitting on large amount of data

Logistic regression (sML)

- When: binary classification (binary labels)
- Performance: simplest non-linear classifier

Decision tress (sML)

- When: prediction models going from observations to conclusions, decision making
- Performance: easy to understand (e.g. RF)

K-means (uML)

- When: assign labels according to features, but labels not available (clustering)
- Performance: ranges of clustering methods with cons/pros call for careful decision for selection

NN (uML)

- When: loosely model human brain to recognize patterns, feature extraction for other ML algos, deep learning, video signals
- Performance: complex, prone to overfitting

Applied ML for SG12 use cases

Performance, QoS (PM, FM, CM, CS, CRM data sources)

- RCA/anomalies, performance trends detection
- Performance prediction
- e.g. ITU-T E series

QoE modeling (device/clients/network data sources)

- Conversational voice/video (OTT, carrier)
- Multimedia streaming (OTT video, gaming/VR)
- e.g. ITU-T P.565, P.1200 series

Rules of thumb for ML applicability

Training/learning database integrity and validity (data cleansing)

- sML vs. uML controllability

Training /validation database split process

- (x,y)% data split (e.g. 50-50%)
- z-fold cross validation, CV (higher z>> increased robustness, min. bias towards a specific use case)

Machine learning feature selection

- Significance of features' content and count (too many >>overfitting; too few >> underfitting)

Algorithms' accuracy / suitability for a specific application

- evaluation based on stats (e.g., RMSE, MAE, percentage of false positives/negatives)

or

representative (e.g., recall and precision charts).

ML overfitting/underfitting test

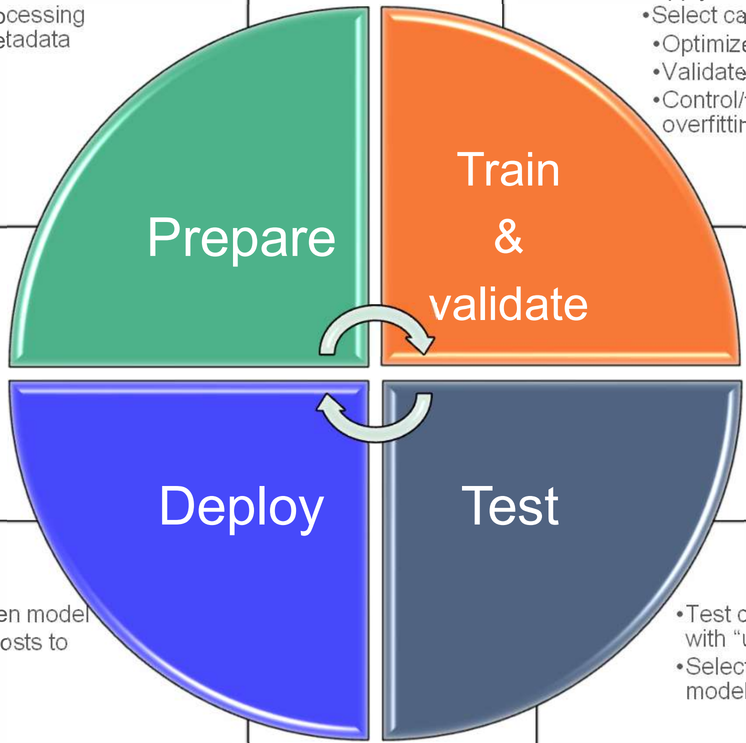
Main phases of ML based solutions development

- Selection of meaningful content / scenarios for the training/validation data set
  - Cleaning process
- Select the right sized data set
- Define the right features (inputs) and their number

- Find/select/create data
- Apply pre-processing (cleaning, metadata creation)

- Apply ML algorithm
- Select candidate model
  - Optimize features
  - Validate, tune
  - Control/test overfitting/underfitting

- Define the selection criteria – use case dependent
- Properly split training/validation data
- Use the appropriate techniques to avoid over/underfitting



- Memory, processing time, responsiveness (real time, non-real-time)
- Data sources/ inputs availability , pre-processing
- Data normalization if data sources /inputs come from different domains

- Deploy chosen model
- Application posts to API

- Test candidate model with "unknown" data
- Select "good enough" model

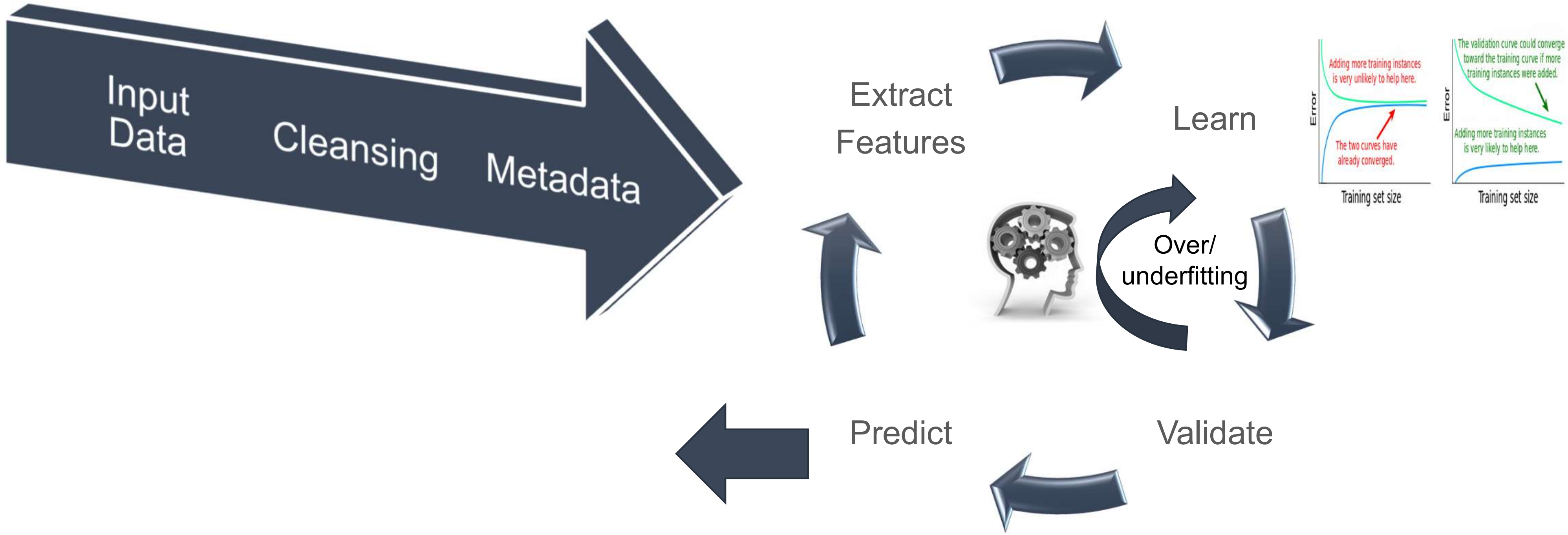
- Select the appropriate "unknown" data set, containing scenarios for which the ML has been trained for
- Define "good enough" performance
  - Loop back to training/validation if needed

Aspects of ML optimization: learning and validation

infovista

KNOW YOUR NETWORK™

Aspects of ML optimization: learning and validation



ML optimization: learning and validation

Learning

Optimization of model's parameters ("hyper parameters")

Cost function per hyperparameter

Trade-off

Highest accuracy/min variance (minimal underfitting)

Min bias (least overfitting)

Error

L_er=Learning (avg. loss over learning/training data)

V_er=Validation (avg. loss over validation data)

High (L_er -V_er) >>> high bias (strong overfitting)

High (L_er) >> high variance (or poor accuracy)

MSE $$\frac{1}{N}\sum_{i=1}^{N} L(y_i - f_w(x_i))^2$$

- N is the total number of observations in the learning set,
- L is the "loss function" (e.g. mean squared error, MSE) over each one of the learning observations
- f_w is the "fit function" (estimated values) using the learning/training data. The validation error is defined as the average loss over an independent test sample

ML optimization: learning and validation

Learning → Optimization of model's parameters ("hyper parameters") → Trade-off

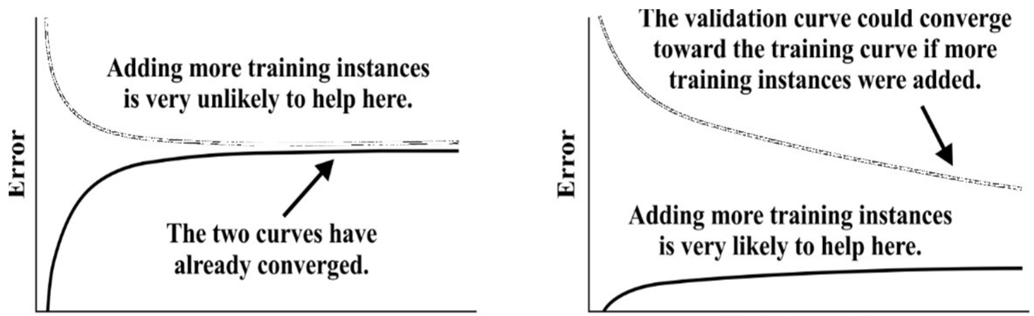
Cost function per hyperparameter

MSE $$\frac{1}{N}\sum_{i=1}^{N} L(y_i - f_w(x_i))^2$$

- z-fold CV technique
  - Run gradually increasing the learning set's size,
  - Result: average value across z number of cases.
  - CV can reduce accuracy in some cases, but ensures robustness and assurance that more scenarios are covered when the split learning (training) and validation data is performed.

- Learning curved based optimization – learning technique dependent

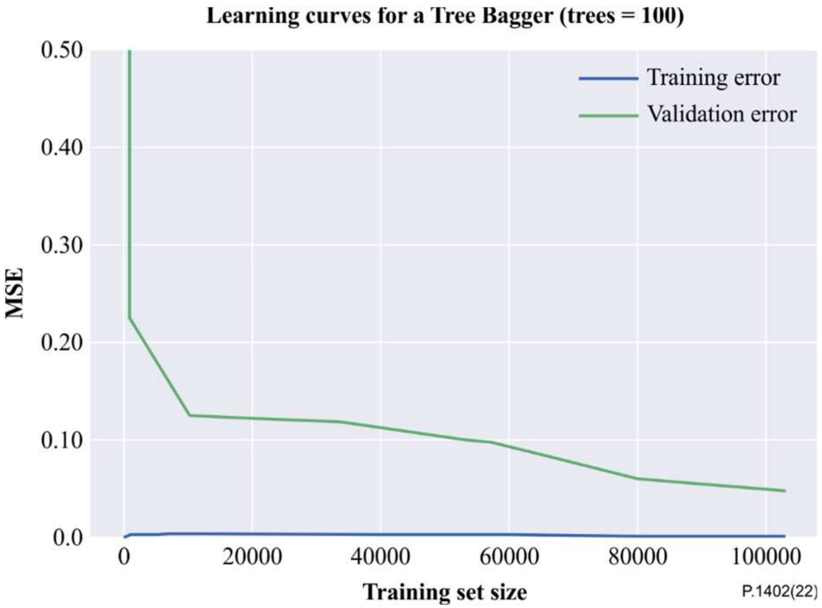
Learning curves

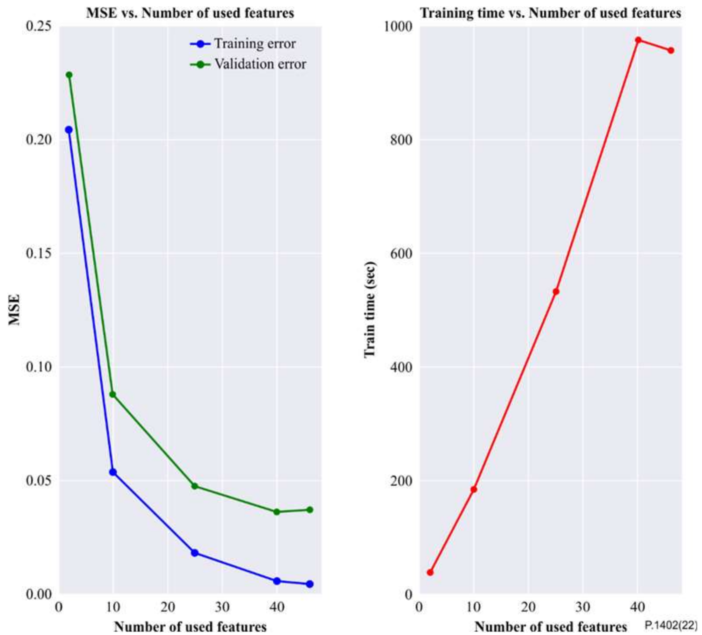


Hyperparameters depending on the ML technique/algorithm (#samples, #features, Param_k_MLtechnique/algo)

ML optimization: learning and validation

- EXAMPLE: Decisions trees (#sample,#features, # trees, #leaves, bootstrap)





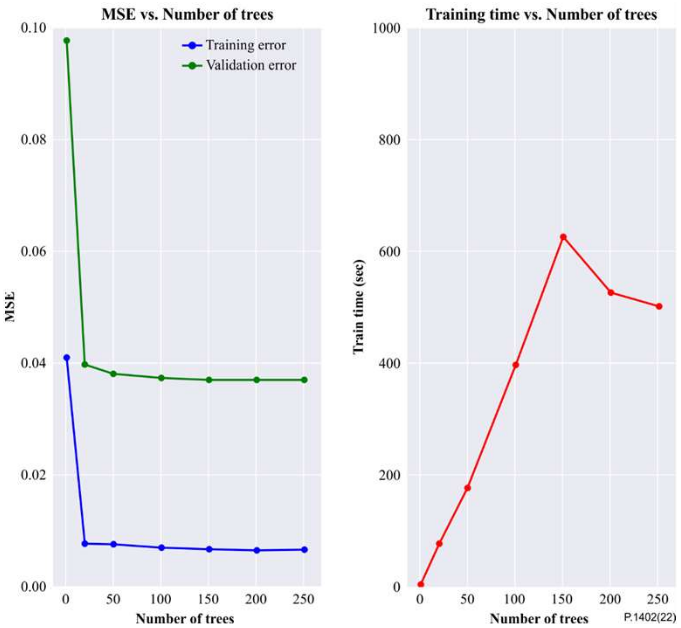
5-CV; random samples for each of the training sizes are taken five times, a prediction is determined and the results are averaged

Trend of V_er suggests that overall accuracy can be improved by increasing the number of samples, but only very slightly.

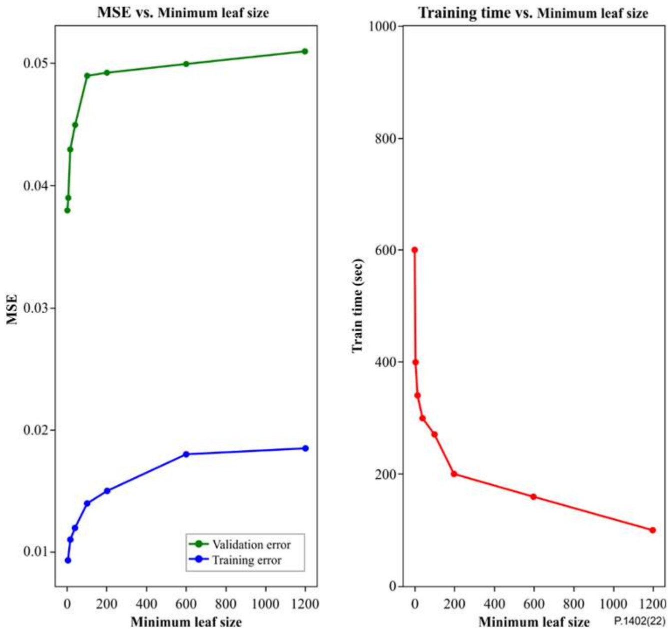
Using 47 instead of 40 features slightly increases the variance, meaning that the model performed slightly worse when the trees had to include the artificially injected redundant feature.
When 40 features are used, the model selects the best ranked features, but with increased training time.
When 46 feature are used >> minimal L_er, V_er >> no overfitting at a fairly good training time of less than 15 min.

ML optimization: learning and validation

- EXAMPLE Decisions trees (#sample,#features, # trees, #leaves, bootstrap)



#100 trees optimal for no overfitting at a fairly good time for training, below 7 min.
>#100 trees V_er remains rather constant, but increased risk of biasing the algorithm towards the training data as the complexity of the model increases



Large # leaves ("large tree") usually fits the training data too well, but performs poorly on an unknown dataset >> high (L_er -V_er) >> overfitting
Small # leaves ("shallow tree") typically performs poorly during the training process, high L_er

Tree size shrinks >> L-er, V_er increase, while the training time decreases as the model becomes less complex. The smallest variance is generated with a minimum leaf size of 5, with a fairly good training time of around 7 min.

A glimpse at ML/AI standardization work

infovista

KNOW YOUR NETWORK™

A glimpse at ML/AI main standardization work

A GLOBAL INITIATIVE

NetWork Data Analytics Function (NWDAF – network core)
Reactive (or Responsive) Analytics; Proactive Network Analytics; and Proactive Subscriber Analytics

World Class Standards

Framework based architecture
Open interfaces, APIs
Use cases for network automation, closed loop autonomy

NFV Management and orchestration (MANO)
OpenNetworkAutomationPlatform ONAP, Generic Autonomic Network Architecture GANA model
ZSM (Zero-touch network and Service Management)
Experiential Network Intelligence (ENI)

ML/AI architecture and data flows for 5G (FocusGroupML5G/SG13)
Autonomous Networks - Use cases (FocusGroup AN/SG13)
ML/AI QoE modeling (SG12)

5G Intelligent Service Operations
(framework, use cases)
(BSS, OSS)

Radio Intelligent Controller (RIC)
Service Management Orchestration (SMO)

https://www.5gamericas.org/wp-content/uploads/2019/11/Management-Orchestration-and-Automation_clean.pdf

Take away

infovista

KNOW YOUR NETWORK™

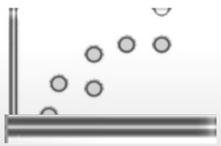
Take away

Technology revolution

- ML embedded: intelligent wireless communications, AI-closed optimi loop, data driven performance management
- ML based testing
- ML applied to SG12 work item – call to action for guidance for applied ML

At a glance on ML technology

- sML / uML techniques for mobile networks
- ITU-T SG 12 uses cases (QoE modelling, QoS performance) and common algorithms (e.g. DT, LinRgr, LogRgr, PCA, K-means, NN)
- Best practices: data validity, cleaning, data split, feature selection, modelling accuracy and bias



ML optimization: leaning and validation

- Cost function: “loss” expressed through L_er and V_er,
- Tradeoff: accuracy vs. bias
- Learning curves-based optimization: ML technique and algorithm dependent to be applied for model’s “hyperparameters” (nr. samples, nr. features, Parameters_k)


Thank you!
www.infovista.com