Failure Management in Optical Networks

Francesco Musumeci

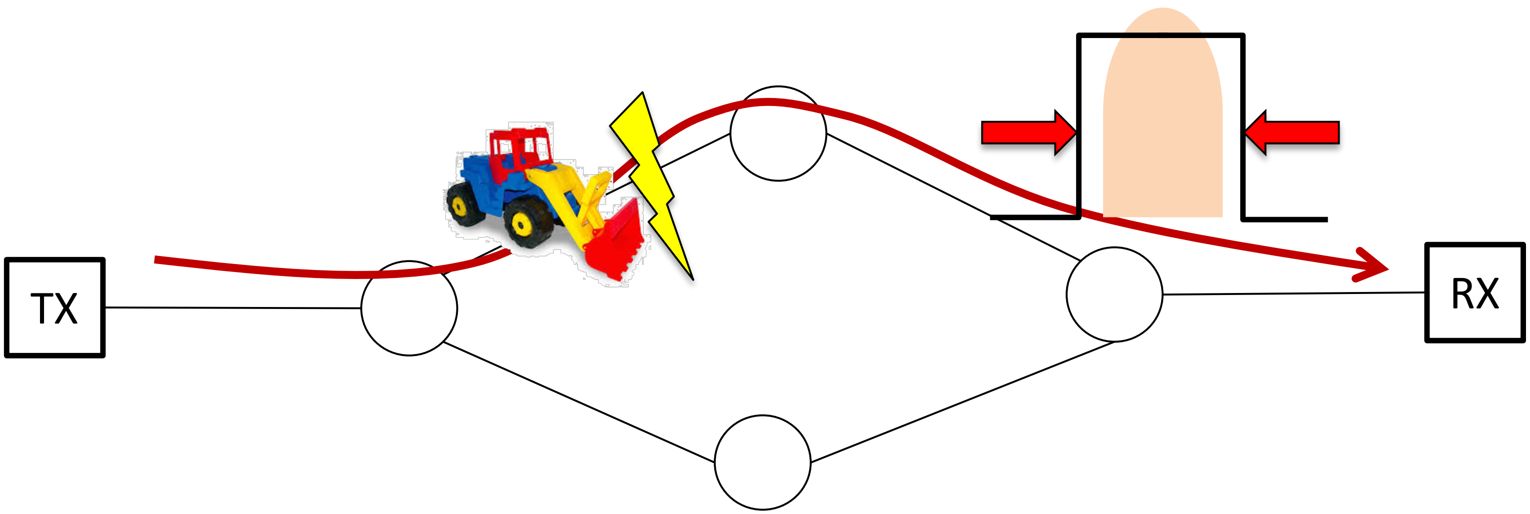
Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB)

Politecnico di Milano, Milano, Italy

Failure management in optical networks
Background

Two main failure types in optical networks

- Hard-failures
  - Sudden events, e.g., fiber cuts, power outages, etc.
  - Unpredictable, require «protection» (*reactive procedures*)
- "Soft"-failures:
  - Gradual transmission degradation due to equipment malfunctioning, filter shrinking/misalignment…
  - Trigger early network reconfiguration (*proactive* procedures)



Failure management in optical networks
Background

- Monitoring signal quality in several network locations is expensive
- OSNR (Optical Signal to Noise Ratio) and/or BER (Bit Error Rate) at the receiver are relatively "cheap" to measure, and are directly impacted in case of failures

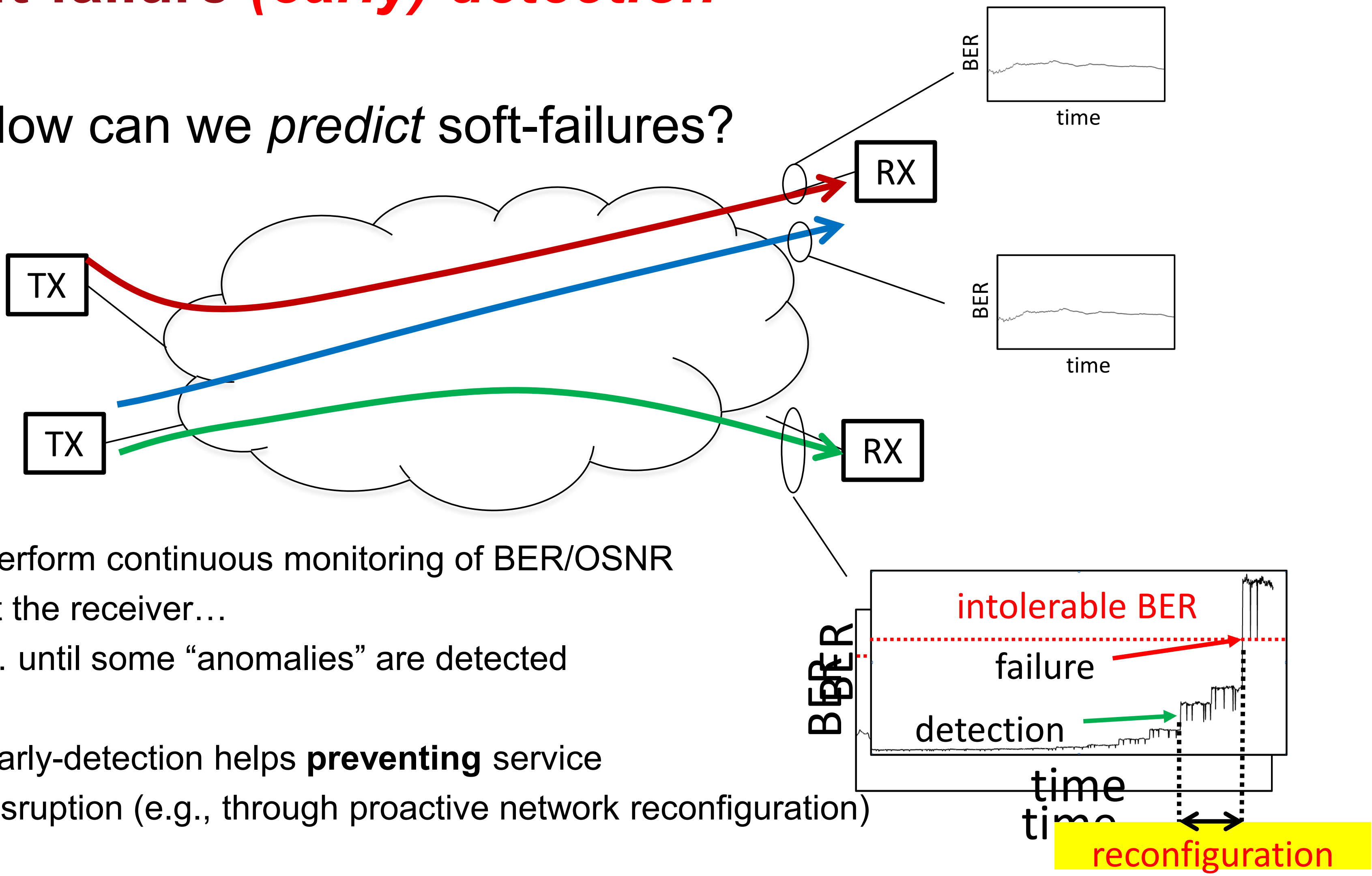
In this lab, two types of soft-failures:

- **Excessive attenuation**, due to fiber ageing, optical amplifiers malfunctioning, etc.
- **Extra filtering**, due to filters misalignment, laser drift (carrier wavelength used by transmitting devices is not perfectly aligned with the DWDM grid), optical switches malfunctioning, etc.

- Different flavours of failure management in optical networks

Soft-failure *(early) detection*

- How can we *predict* soft-failures?

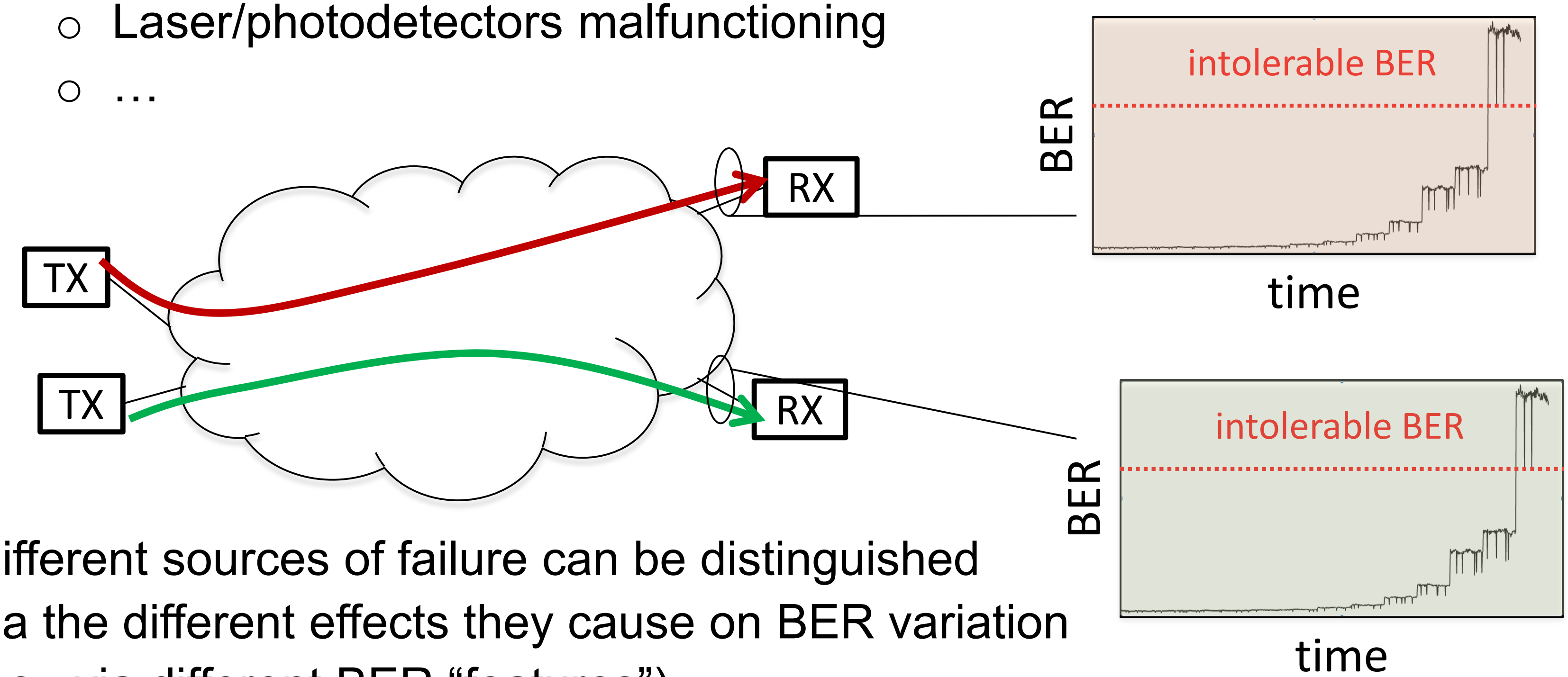


Perform continuous monitoring of BER/OSNR at the receiver…
… until some “anomalies” are detected

Early-detection helps **preventing** service disruption (e.g., through proactive network reconfiguration)

Soft-failure *cause identification*

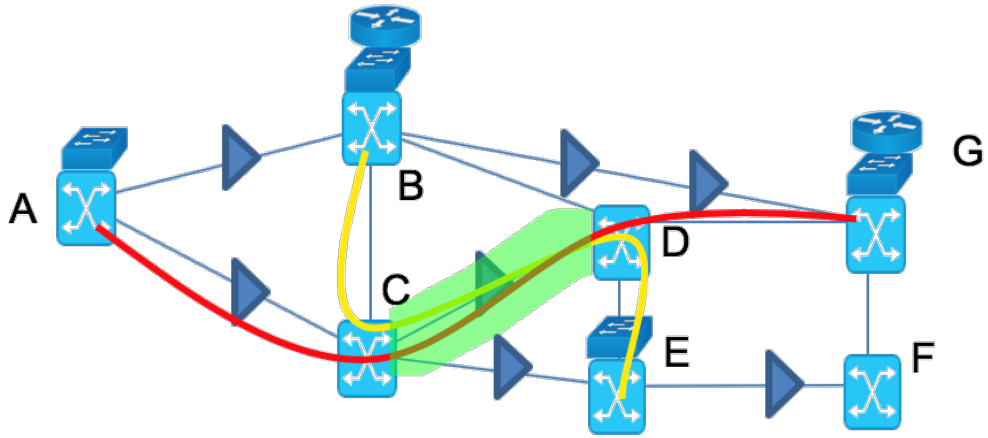
- How can we identify the *cause* of the failure?
  - Failures can be caused by different sources
    - Filters shrinking/misalignment
    - Excessive attenuation (e.g., due to amplifier malfunctioning)
    - Laser/photodetectors malfunctioning
    - …



Different sources of failure can be distinguished via the different effects they cause on BER variation (i.e., via different BER "features")

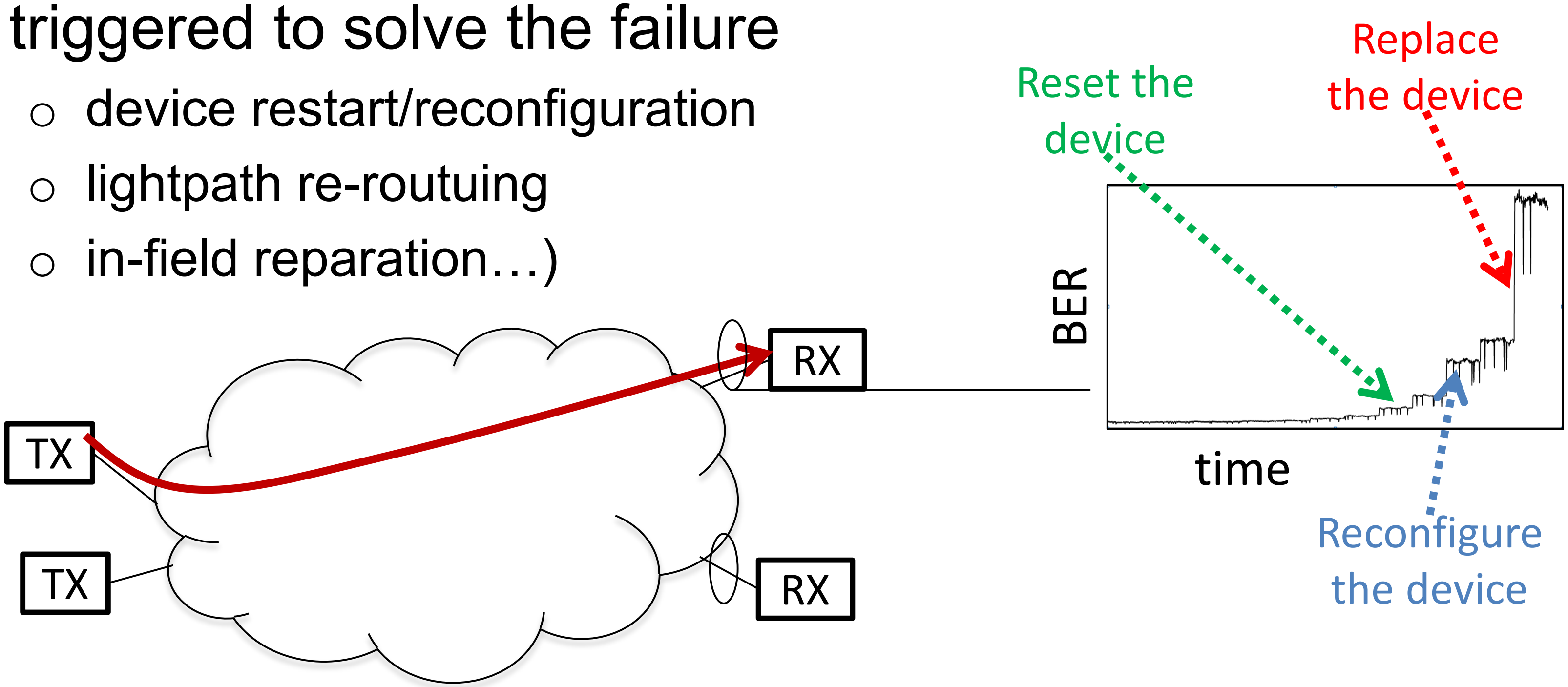
Soft-failure *localization*

- How can we identify the location of the failure?
  - A single failure may affect multiple lightpaths
    - Leverage information on failure-cause on each lightpath in combination with routing information
    - No need for monitoring in the entire network (monitors can be deployed only at the receivers)
  - Even with information on a single lightpath (e.g., at the receiver), does the impact change with distance between failure and receiver?



Soft-failure *magnitude estimation*

- What is the failure magnitude (i.e., severity)?
  - Different failures magnitude can affect the network differently
  - According to the severity, different actions can be triggered to solve the failure
    - device restart/reconfiguration
    - lightpath re-routuing
    - in-field reparation…)



Handling soft-failures

Summary

Lab's focus

1. **(Early) Detection** (Whether or not?)
   - Predict/assess if OSNR/BER is/will be intolerable
   - Allows early/quick activation of proactive procedures

2. **Identification** (Which cause?)
   - e.g., filter misalignment, laser drift, fiber bending, amplifier malfunctioning ..
   - Reduced Mean Time To Repair (MTTR)

3. **Localization** of soft-failures (Where?)
   - e.g., which node/link along the path?

4. **Magnitude estimation** (How much?)
   - Triggers the proper reaction (e.g., device restart/reconfiguration, lightpath re-routuing, in-field reparation…)

Failure management in optical networks
Source paper

JOURNAL OF LIGHTWAVE TECHNOLOGY, VOL. 37, NO. 16, AUGUST 15, 2019 4125

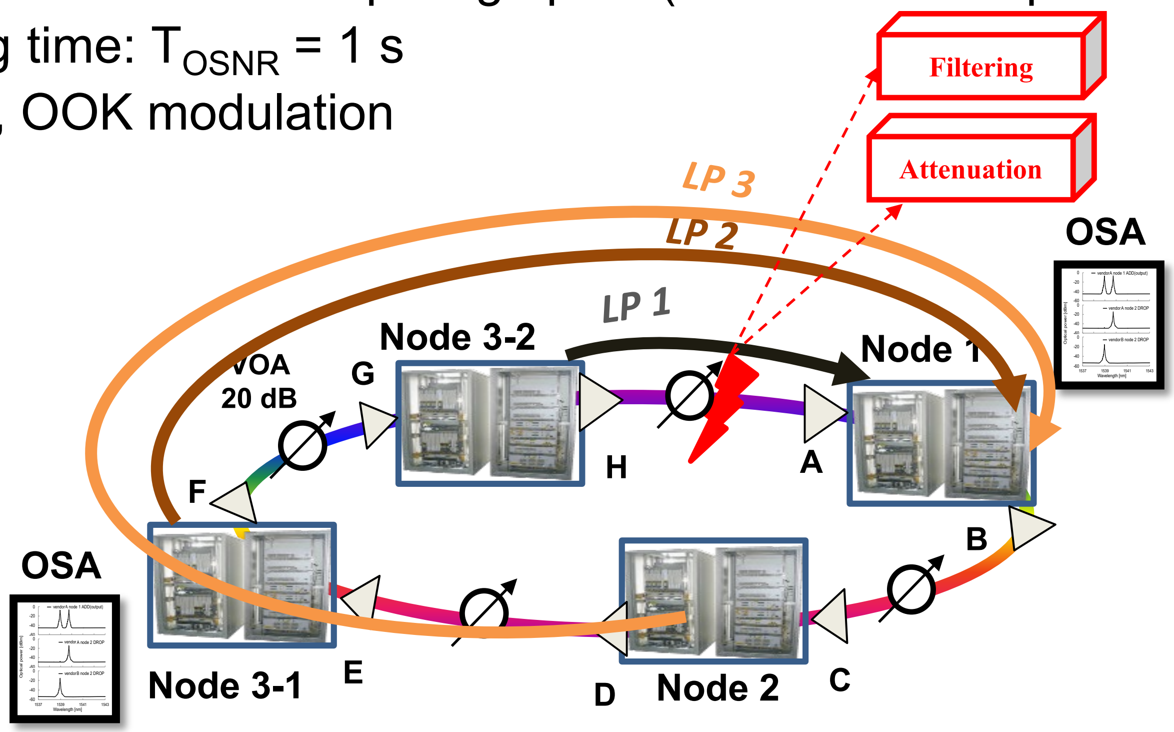
A Tutorial on Machine Learning for Failure Management in Optical Networks

Francesco Musumeci, Cristina Rottondi, Giorgio Corani, Shahin Shahkarami, Filippo Cugini, and Massimo Tornatore

(Invited Tutorial)

Testbed Setup

- Opt. Net. testbed with 4 Reconfigurable Optical Add/Drop Multiplexers (ROADMs)
  - OSNR collected for different lightpaths at different monitoring locations
  - Center-wavelength @194.8 THz, BW=100 GHz
  - 6 hours of measurement per lightpath (with some exceptions…)
  - Sampling time: T_{OSNR} = 1 s
  - 10 Gbps, OOK modulation



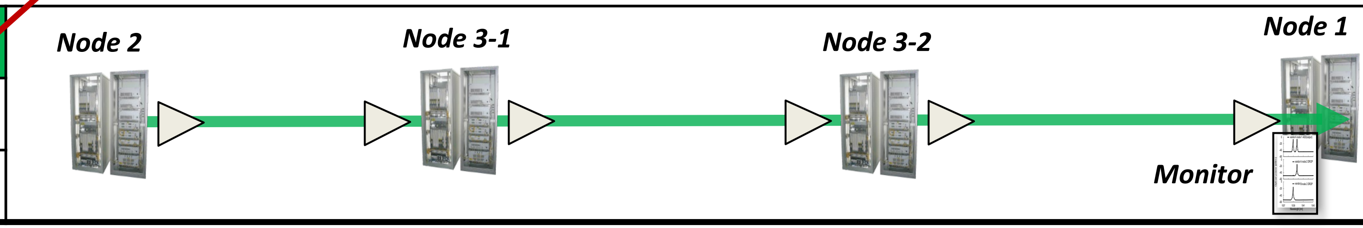
Failure management in optical networks

Dataset (1)

	Description	Magnitude	Location (link)
Scenario A	No failures	-	-
Scenario B	Excessive attenuation	11 dB	3-2 – 1
Scenario C	Extra filtering	12.5 GHz	3-2 – 1

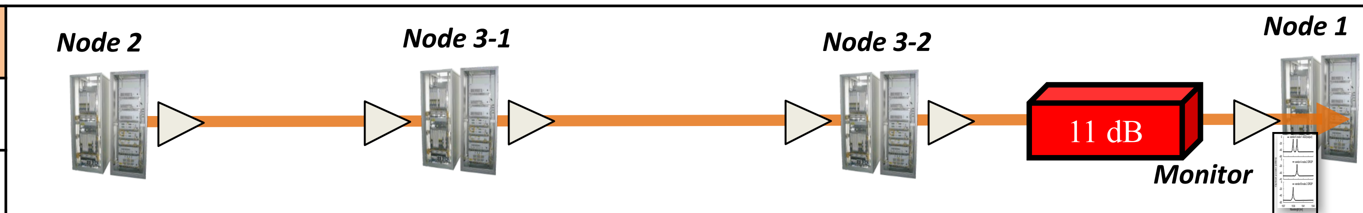
Lightpath ID [SRC]_[DEST]

Scenario A
2_1
21600

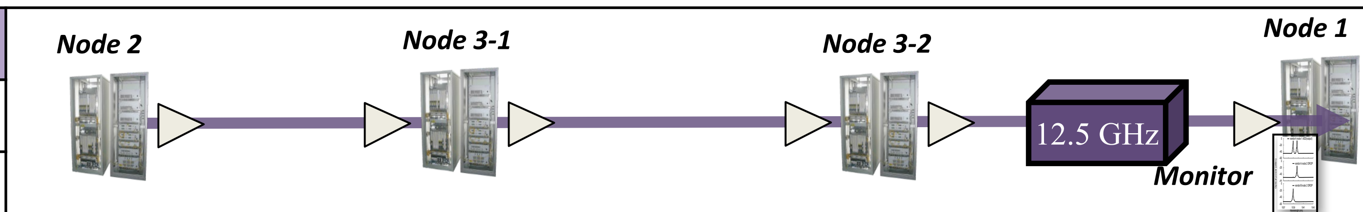


Number of OSNR samples (6hours)

Scenario B
2_1
21600



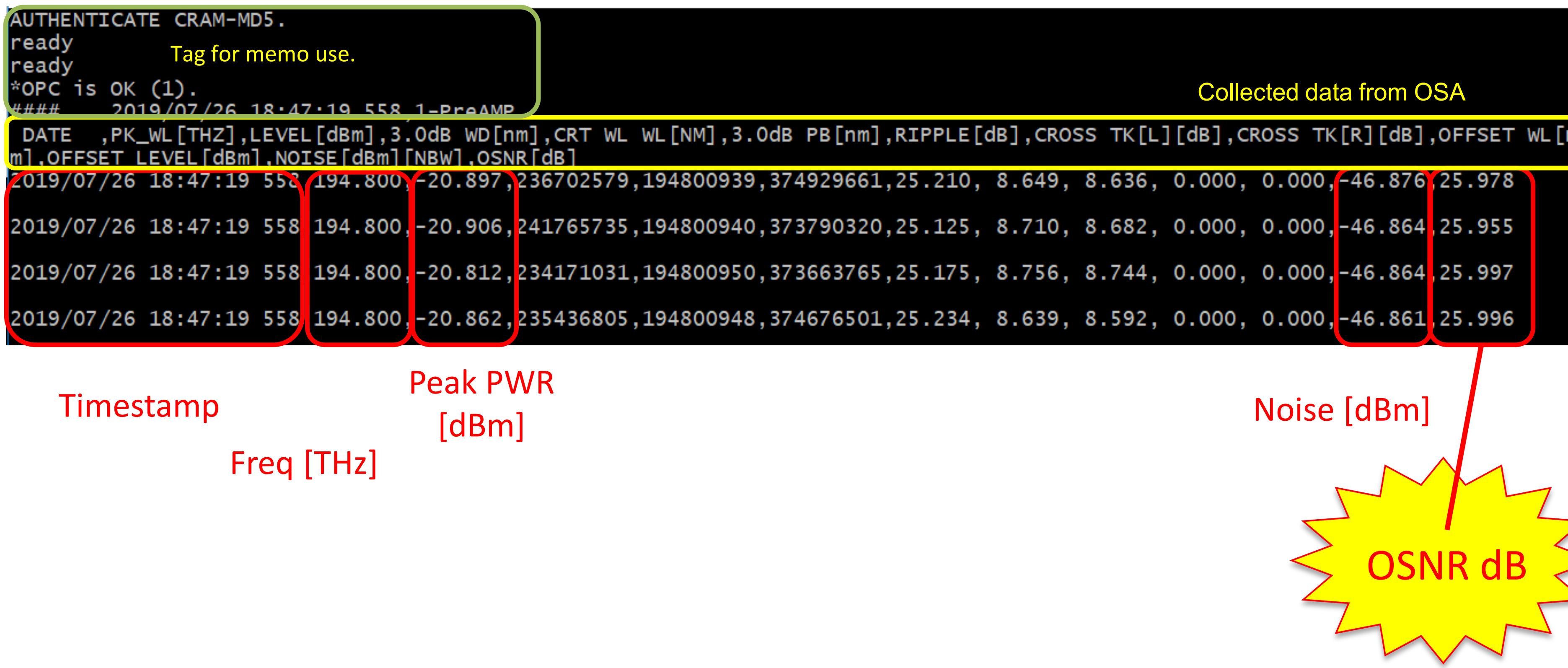
Scenario C
2_1
21600



Failure management in optical networks

Dataset (2)

- Raw dataset: how it looks like



Failure management in optical networks
Lab overview

Task 1. Raw data visualization

Task 2. Dataset creation

Task 3. Dataset loading

Task 4. Hyperparameters optimization and model training

Task 5. Testing and performance evaluation

Task 6. Putting things together: Failure detection

Task 7 (**HOMEWORK**). Putting things together: Failure identification

Appendix.
Data visualization with PCA: motivates results for task 6-7

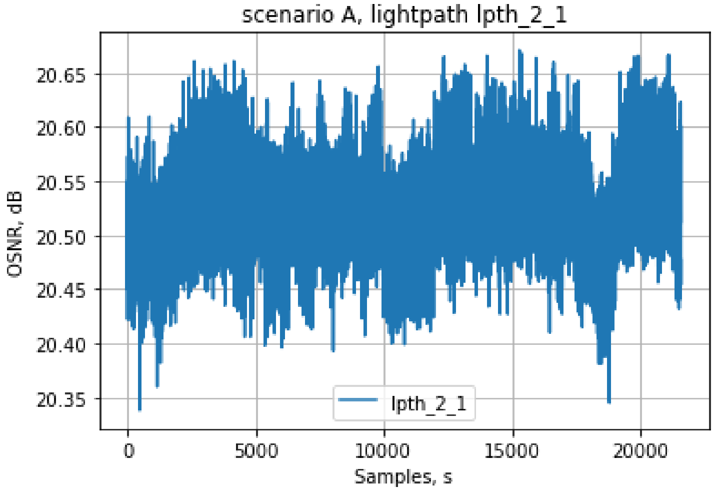
Failure management in optical networks
Task 1

1. Visualize OSNR data
   a) Define a function *plot_scenario*() that plots OSNR traces given a file name in input
      - See skeleton code for the details
   b) Use function *plot_scenario()* to plot OSNR figures for scenarios A-B-C
      - **Already given in skeleton code**

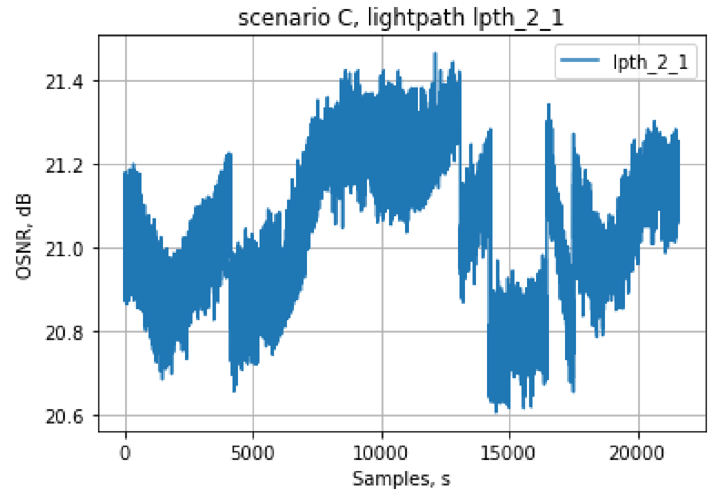
Failure management in optical networks

Task 1: expected outputs

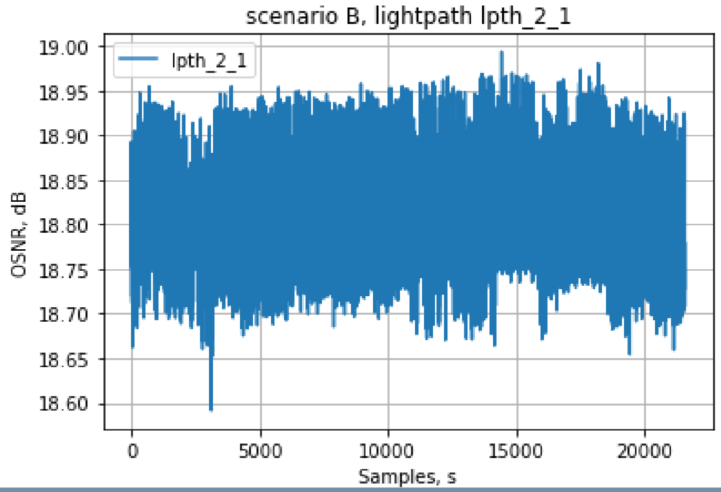
Scenario A
No failure



Scenario C
Extra filtering (12.5 GHz)



Scenario B
Excessive attenuation
(11 dB)



Failure management in optical networks
Task 2

Dataset creation

- Each data point will consist of a "window" of duration W, including a sequence of consecutive OSNR samples
- From each window, we compute the following features:

1. mean: $\mu = \frac{1}{W}\sum_{i=1}^{W} OSNR_i$
2. Root mean square:

$$\text{RMS} = \sqrt{\frac{1}{W}\sum_{i=1}^{W}(OSNR_i)^2}$$

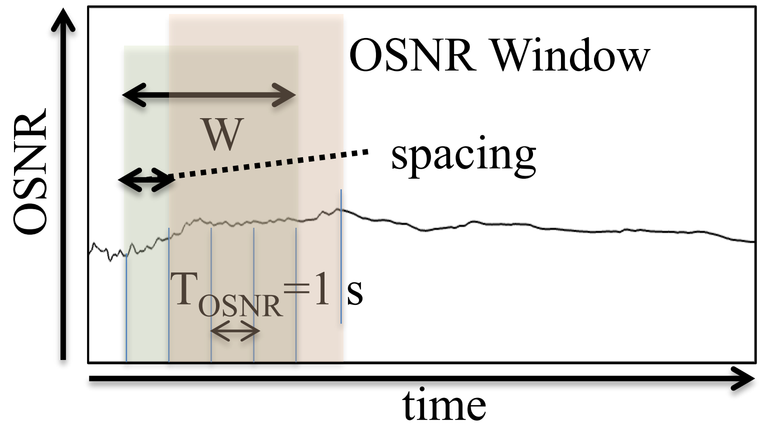
3. peak-to-peak:

$$Ptp = Max - min$$

4. Standard dev:

$$\text{STD} = \sqrt{\frac{1}{W}\sum_{i=1}^{W}|OSNR_i - \mu|^2}$$

5. Max: $M = \max_{i\in[1,W]}(OSNR_i)$
6. min: $m = \min_{i\in[1,W]}(OSNR_i)$



Failure management in optical networks

Task 2

2. Create the window dataset
   a) Define a function *create_window_dataset()* that takes in input raw data file, window length and spacing and creates a features .dat file in a subfolder called ‘Datasets'
      - See skeleton code for the details
      - **Task 2a’) is given in the skeleton to test function *create_window_dataset()***

   b) Use function *create_window_dataset()* to generate features files for scenarios A-B-C using window duration =10,20,...,100 seconds and window spacing = 1 second

Failure management in optical networks

Task 2: expected outputs

Scenario A - No failure – W=10

mean,RMS,ptp,std,max,min
20.509,20.509,0.123,0.032,20.549,20.426
20.512,20.512,0.124,0.034,20.55,20.426
20.513,20.513,0.124,0.036,20.55,20.426
…

Scenario B Excessive attenuation (11 dB) – W=10

mean,RMS,ptp,std,max,min
18.815,18.815,0.141,0.052,18.893,18.752
18.816,18.816,0.141,0.052,18.893,18.752
18.824,18.824,0.141,0.048,18.893,18.752
…

Scenario C Extra filtering (12.5 GHz) – W=10

mean,RMS,ptp,std,max,min
21.058,21.058,0.254,0.08,21.151,20.897
21.062,21.062,0.254,0.08,21.151,20.897
21.054,21.054,0.248,0.074,21.145,20.897
…

Failure management in optical networks
Task 3

3. Load dataset
   a) Define a function *load_window_dataset()* that takes in input window data file, and label to be assigned and returns numpy arrays with features and labels
      - See skeleton code for the details
   b) Use function *load_window_dataset()* with datasets of scenarios A-B-C using **window length = 10 and spacing = 1**. Assign labels so as to generate a **binary dataset** useful for **failure detection problem** (0=normal, 1=failure)
      - **Already given in skeleton code**
   c) Do a 80/20 split of the dataset into train/test with balanced classes and check consistency between no of positives (i.e., datapoints with y=1) across train/test/full sets

Failure management in optical networks

Task 3b)-c): expected outputs

While appending data from the 3 scenarios

```
current shape of X: (21591, 6)
current shape of y: (21591,)
current shape of X: (43182, 6)
current shape of y: (43182,)
current shape of X: (64773, 6)
current shape of y: (64773,)
```

Final dataset (X,y)

```
print(X)
print(y)
print(X.shape)
print(y.shape)
```

```
[[20.509 20.509  0.123  0.032 20.549 20.426]
 [20.512 20.512  0.124  0.034 20.55  20.426]
 [20.513 20.513  0.124  0.036 20.55  20.426]
 ...
 [21.143 21.143  0.157  0.043 21.233 21.076]
 [21.142 21.142  0.157  0.043 21.233 21.076]
 [21.144 21.144  0.157  0.042 21.233 21.076]]
[0 0 0 ... 1 1 1]
(64773, 6)
(64773,)
```

Train/test datasets

```
print(X.shape)
print(y.shape)
print(Xtrain.shape)
print(ytrain.shape)
print(Xtest.shape)
print(ytest.shape)
```

```
(64773, 6)
(64773,)
(51818, 6)
(51818,)
(12955, 6)
(12955,)
# positives in y = 43182
# positives in ytrain = 34545
# positives in ytest = 8637
```

Failure management in optical networks
Task 4

4. Algorithms **hyperparameters optimization** and **training**
   a) Define function *train_classifier_logistic()* that performs hyperparameter optimization with 5-fold crossvalidation and training and returns the trained model
      - See skeleton code for the details
      - **Part of the function is given in the skeleton**
      - **Task 4a') is given in the skeleton to test function *train_classifier_logistic()***

Failure management in optical networks

Task 4a): expected output

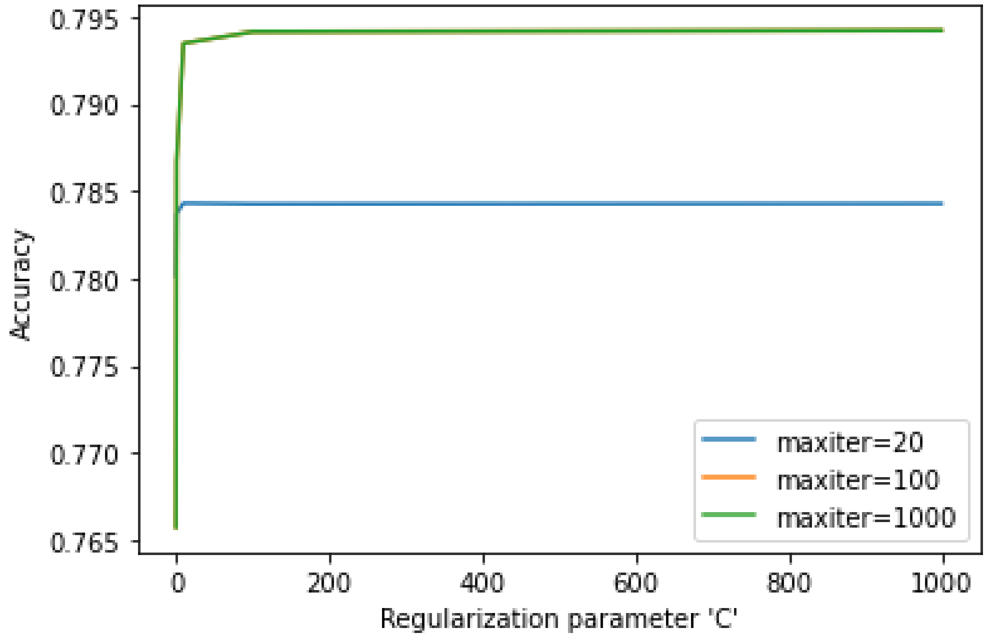
Crossval time [s]: 35.141

Best hyperparams during crossval: {'regularization': 1000, 'max_iter': 100, 'score': 0.7942222357373689}

[[0.78013447 0.76573791 0.76573791]
 [0.78372395 0.78671519 0.78671519]
 [0.7843222 0.79348889 0.79348889]
 [0.7842836 0.79414504 0.79414504]
 [0.7843029 0.79422224 0.79422224]]

Score of the final model: 0.7945115596896831

Training time [s]: 0.608



Failure management in optical networks
Task 4

4. Algorithms **hyperparameters optimization** and **training**
   b) Consider a **new dataset *(X_norm, y)* instead of *(X,y)* with normalized features** and repeat train/test split and hyperparameters selection with crossvalidation with the normalized dataset

   Expected output:

What are the main changes you observe before and after normalization?

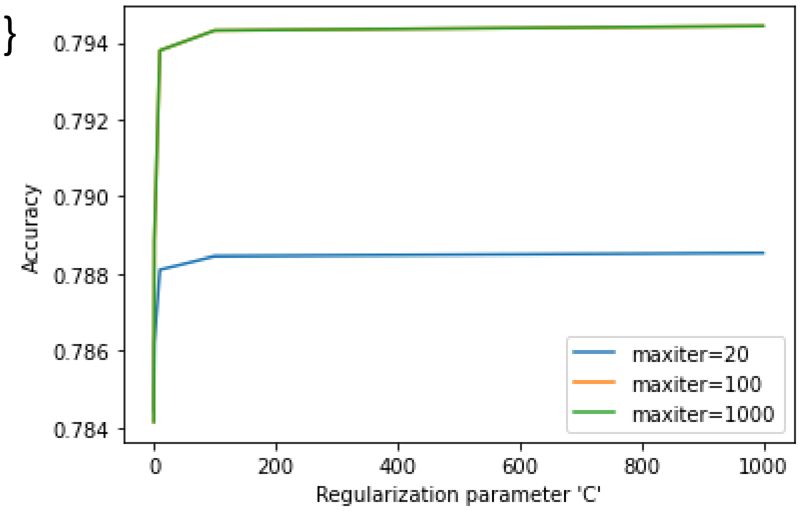
Crossval time [s]: 16.492

Best hyperparams during crossval: {'regularization': 1000, 'max_iter': 100, 'score': 0.7944345034044575}

```
[[0.78445725 0.78414848 0.78414848]
 [0.78619413 0.7888573  0.7888573 ]
 [0.78810466 0.79379768 0.79379768]
 [0.78845202 0.79431872 0.79431872]
 [0.78852921 0.7944345  0.7944345 ]]
```

Score of the final model: 0.7946852445096299

Training time [s]: 0.273



- **From now on, we will always use the normalized dataset (X_norm,y)**

Failure management in optical networks

Task 4

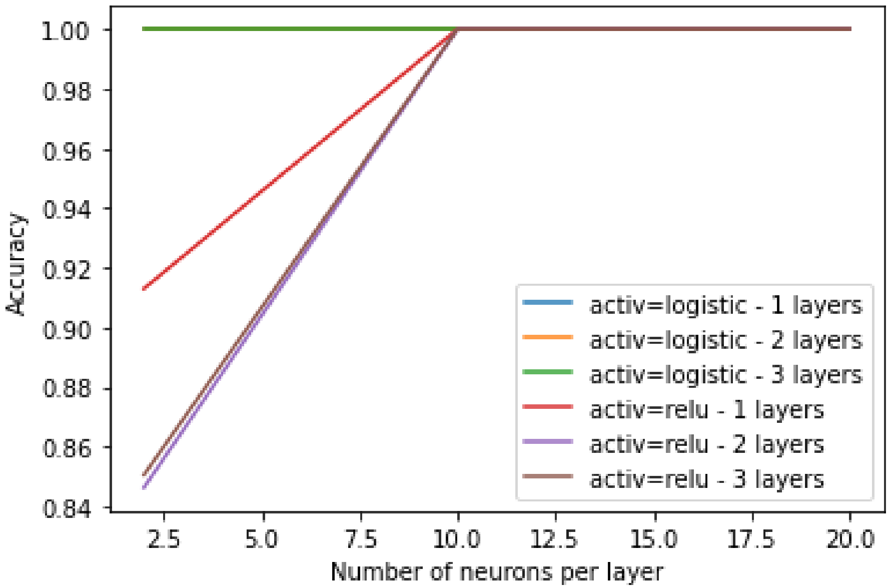
4. Algorithms **hyperparameters optimization** and **training**
   c) Define function *train_classifier_DNN()* that performs hyperparameter optimization and training for DNN with given hyperparameters space
      - See skeleton code for the details
   d) Use function *train_classifier_DNN()* to perform hyperparameters selection with crossvalidation for DNN and **considering the normalized dataset retrieved in task 4b)**
      - **Already given in skeleton code**

Failure management in optical networks

Task 4c)-d): expected output

```
Crossval time [s]: 730.453
Best hyperparams: {'activation': 'logistic', 'layers': 1, 'neurons': 2, 'score': 1.0}
[[[1.         1.         1.        ]
  [1.         1.         1.        ]
  [1.         1.         1.        ]]

 [[0.91296387 1.         1.        ]
  [0.84625627 1.         1.        ]
  [0.85051708 1.         1.        ]]]
```




```
Score of the final model: 1.0
Best activation: logistic.
Best no of layers: 1.
Best no of neurons: 2
Training time [s]: 18.067
```

Failure management in optical networks

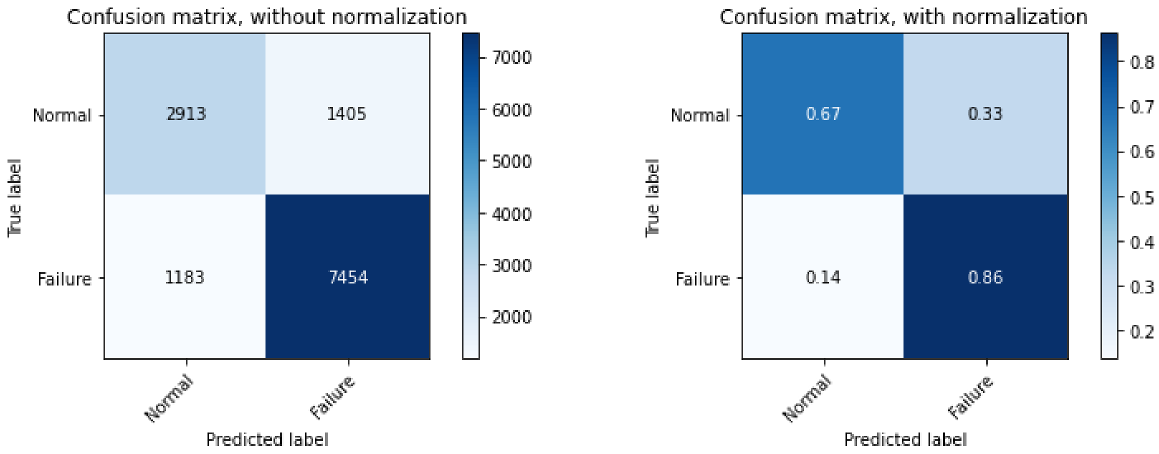
Task 5

5. Test and performance evaluation
   a) Define function *performance_eval()* that takes in input ground truth and predicted labels, prints results in a result file passed in input, and returns global metrics
      - See skeleton code for the details
   b) Perform prediction using the optimized logistic regression and DNN models and **considering the normalized dataset retrieved in task 4b)**. Then, evaluate performance of the two models using function *performance_eval()*

Failure management in optical networks

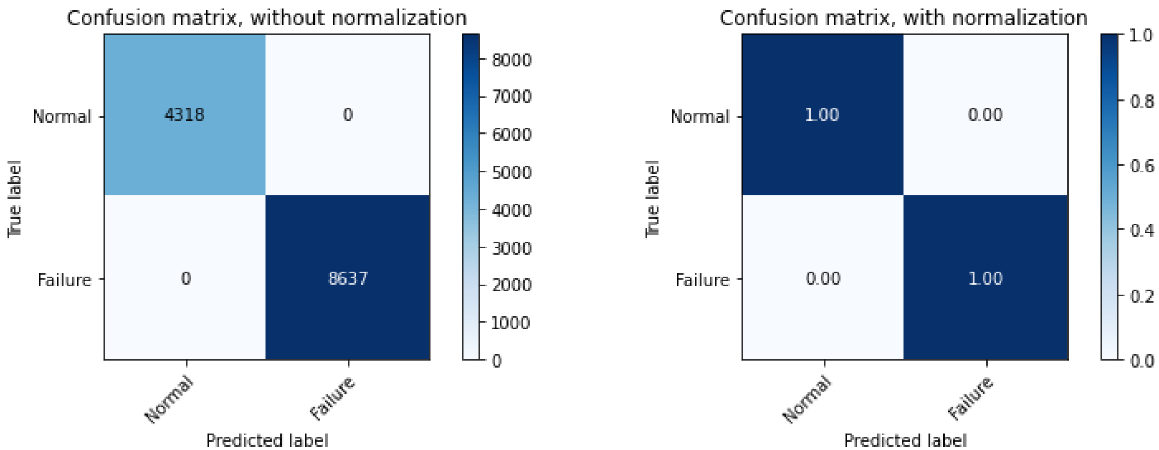
Task 5: expected outputs

Logistic regression



Logistic regression metrics (accuracy, global precision, global recall, global f1score): (0.8002315708220764, 0.798000045310873, 0.8002315708220764, 0.7988635583610862)

DNN



DNN metrics (accuracy, global precision, global recall, global f1score): (1.0, 1.0, 1.0, 1.0)

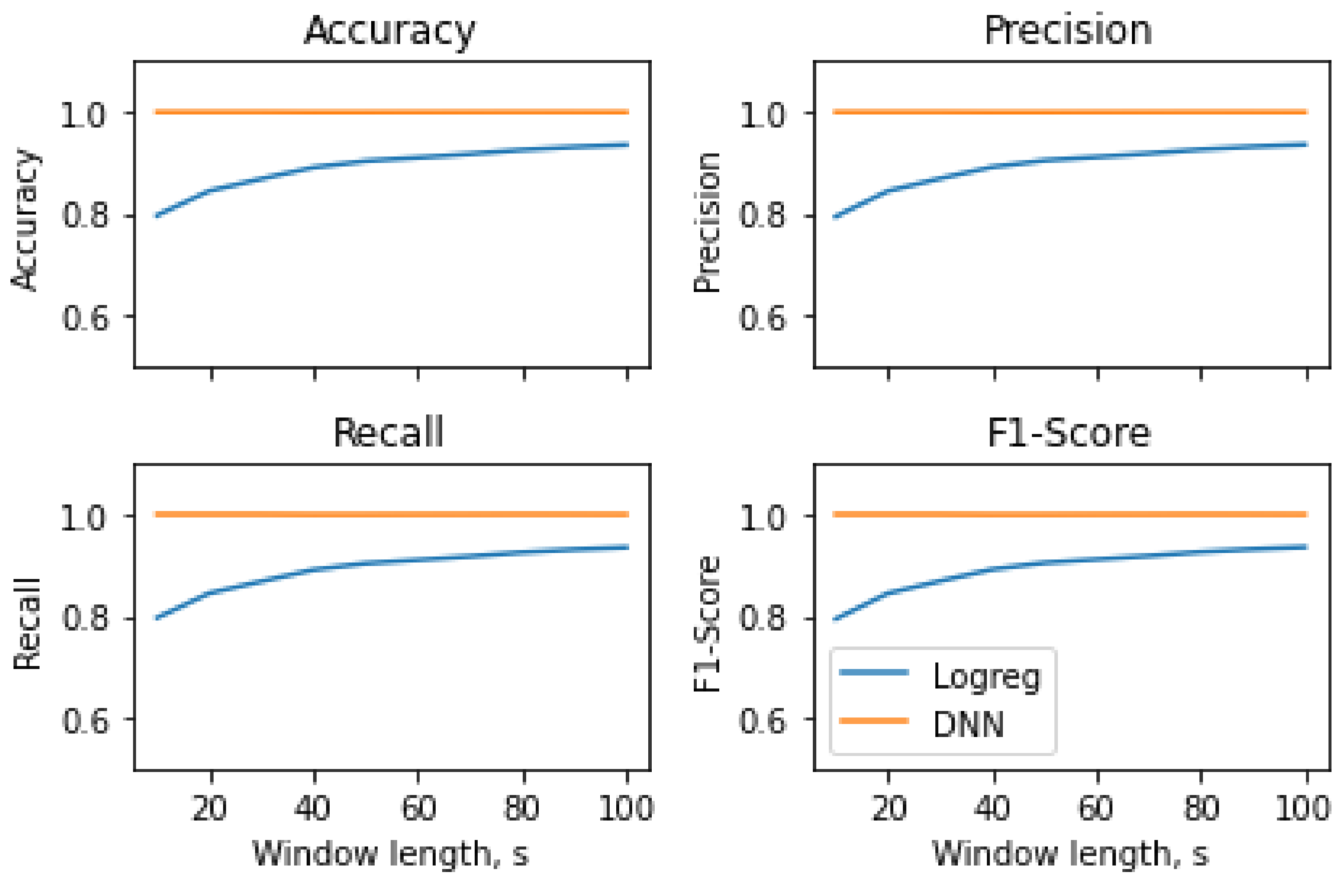
Failure management in optical networks

Task 6

6. Putting things together**: failure detection**
   a) For window length in *range(10,101,10)* and *window spacing* = *1*, repeat dataset load (task 3b), data normalization and dataset split (task 4b), training of logistic regression and DNN with best hyperparameters **(as those obtained in tasks 4b) and 4d), i.e., no need to re-do hyperparameter optimization)**, test and performance evaluation (task 5b), store global results (accuracy, precision, recall and F1-score) in proper lists for each ML algorithm
   b) Plot the 4 metrics vs window length in 4 separate graphs, where each graph includes one curve for each ML algorithm
      - **Already given in skeleton code**

Failure management in optical networks

Task 6b): expected outputs



Failure management in optical networks

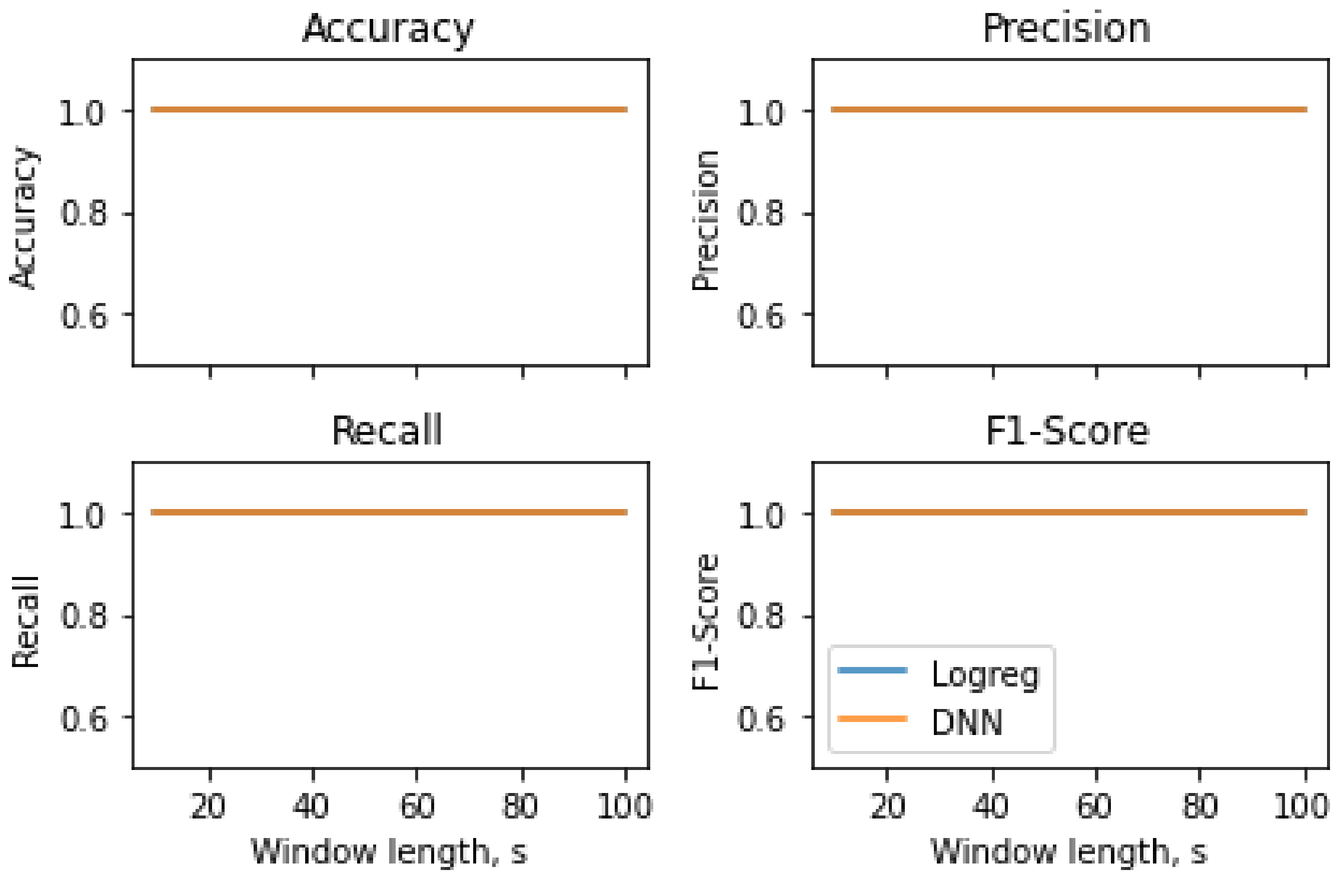
Task 7 – **HOMEWORK (max 1 point)**

7. Putting things together**: failure identification**

   a)-b) Repeat tasks 6a)-6b) but considering only failure classes (scenario B: Attenuation, scenario C: Filtering)

Failure management in optical networks

Task 7: expected outputs – **HOMEWORK (max 1 point)**



We found that logistic regression for failure detection fails (has lower accuracy) for some window sizes (e.g. length = 10 s) compared to failure identification.
Why?

Failure management in optical networks
Appendix

Doing a transformation with PCA and 2 components, we are able to visualize data on a 2D graph

- **Already given in skeleton code**

What can we observe and conclude regarding the previous question?

We found that logistic regression for failure detection fails (has lower accuracy) for some window sizes compared to failure identification.
Why?
ANSWER→