POLITECNICO
MILANO 1863

# Network Data Analysis Laboratory

## Proposed projects

Francesco Musumeci

Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB)

Politecnico di Milano, Milano, Italy

2023-2024

# General requirements (1/2) – Projects tasks

- All projects **must** include the following main steps:
  1. Raw data visualization/analysis
     - Does data suggest anything (ML algorithm, proper features)?
  2. Data preprocessing (e.g., generate features from raw data)
  3. ML models optimization and training
     - Hyperparameters tuning with cross-validation
  4. Testing the performance and evaluating different scenarios (specified in the assignment), e.g.:
     - What is the impact of including/excluding different features?
     - What is the impact of feature normalization?
     - What is the impact of training set size on algorithms' performance?
     - What happens if some features are missing for some data points?

# General requirements (2/2) – Methodology

- All projects **must** include (for 12 points)
  - 2 or 3 ML algorithms (usually recommended in the assignment):
    - a more complex one: Neural Network or a tree-based model (e.g., Random Forest, XGBoost)…
    - a "simpler" one: linear/logistic regression, K-nearest-neighbors
  - Performance metrics: MSE, MAE, Accuracy, Precision, Recall, F-score, training duration, …

- Additionally, projects **may** include "advanced" tasks (we will discuss in another lab in detail) (for 3 points)
  - Transfer learning: train on one domain, test on another domain (e.g., different datasets, different tasks, etc.)
  - Federated learning: compare "global" models (all data available at one location) vs. "local" models that share knowledge through Federated Learning
  - Explainability: apply XAI (eXplainable Artificial Intelligence) frameworks to interpret/explain model reasoning and validate the model

  …

# Projects

1. EDFA profile MIMO regression
2. EDFA profile MISO regression
3. EDFA profile multi-span regression
4. Application flow clustering
5. QoT estimation in optical networks
6. Optical failure localization
7. Traffic forecasting
8. Traffic forecasting: local vs. global
9. Data-driven ML models: the more, the merrier?! (I)
10. Data-driven ML models: the more, the merrier?! (II)

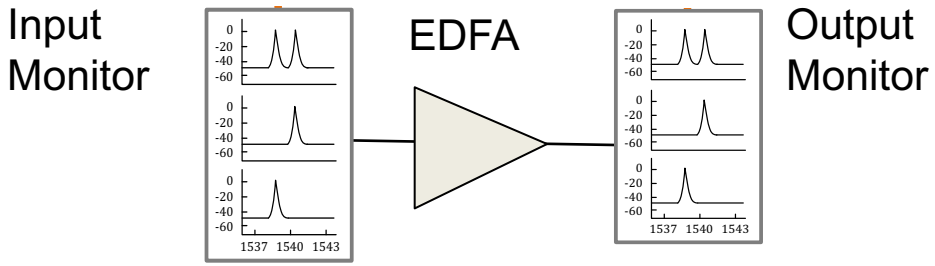
# Project #1-2-3 – EDFA profile regression
# Background

- Optical amplifiers (Erbium-Doped Fiber Amplifier, EDFA) compensate power attenuation in optical fibers and guarantee sufficient power at the receiver
- Input power profile: $P_{in}(\Lambda) = \{P_{in}(\lambda_1), P_{in}(\lambda_2), \dots, P_{in}(\lambda_N)\}$
- Output power profile: $P_{out}(\Lambda) = \{P_{out}(\lambda_1), P_{out}(\lambda_2), \dots, P_{out}(\lambda_N)\}$
- Gain is not the same across the wavelengths
  - It is **linearly tilted** to compensate **for different fiber attenuation at different wavelengths**
  - It has additional **ripple due to imperfections of the production process**
- Complex transfer function: $P_{out}(\Lambda) = f\left(P_{in}(\Lambda)\right)$

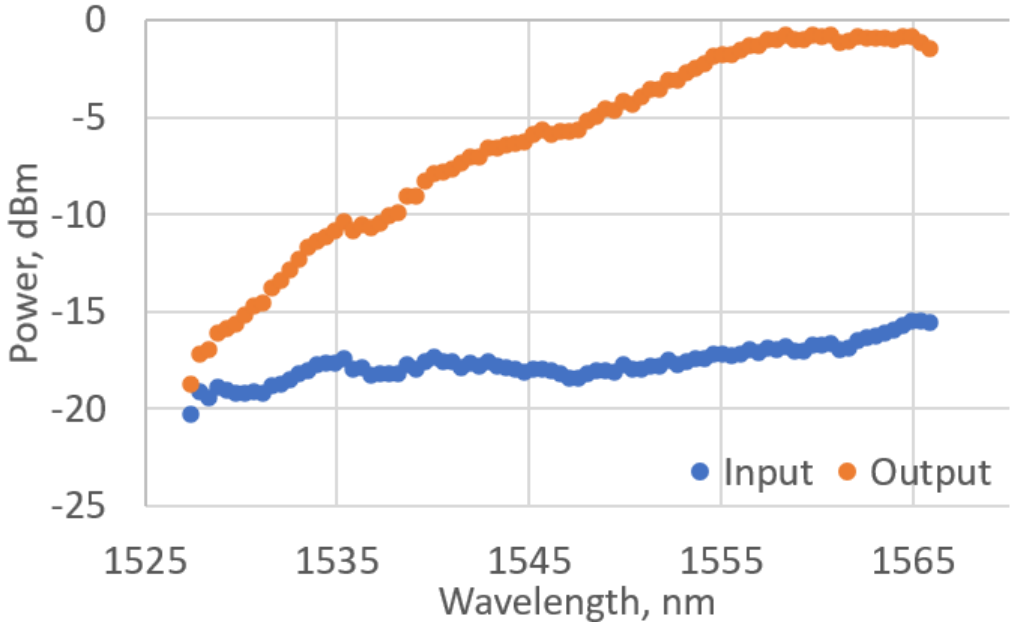
- If we can estimate $P_{out}$ from $P_{in}$ along signal path before launching the signal, we can choose the best wavelength for transmission:
  - To have the flat power profile at the receiver
  - To have highest SNR at the receiver

- However, there is no known analytical model for $f$. **Can we use a ML-model instead?**

# Project #1-2-3 – EDFA profile regression
# Dataset 1




- From TUD, December 2020 [1, 2]
- Input/output power profiles for a single EDFA
  - Different gain settings:
    - Total input power to the EDFA varies in the [−9; 9] dBm range
    - Total output power is 15 dBm
  - One power measurement in each of 84 channels
  - 16497 entries with different power profiles
  - Synthetic input power profiles – not real on/off channels

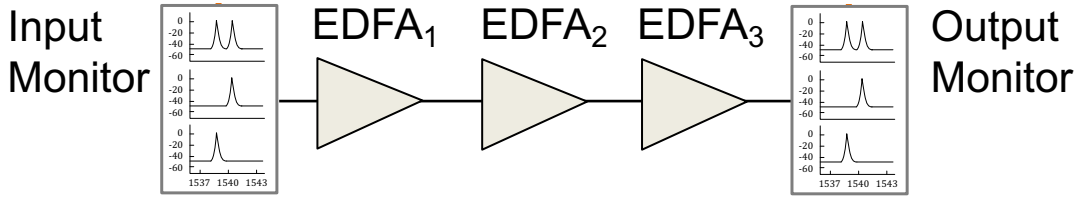



- Dataset structure:

| Profile Id | Total Power In | Total Power Out | Input profile | | Output Profile | |
|---|---|---|---|---|---|---|
| | | | P. Ch. 1 | P. Ch. N | P. Ch. 1 | P. Ch. N |

[1] https://data.dtu.dk/articles/dataset/Input-output_power_spectral_densities_for_three_C-band_EDFAs_and_four_multispan_inline_EDFAd_fiber_optic_systems_of_different_lengths/13135754/1
[2] https://ieeexplore.ieee.org/document/9333297

# Project #1-2-3 – EDFA profile regression
# Dataset 2




- From TUD, December 2020 [1,2]
- Input/output power profiles for a line of 3 EDFAs
  - Different gain settings:
    - Total input power to each EDFA varies in the [-4; 0] dBm range
    - Total output power of each EDFA is 15 dBm
  - One power measurement in each of 84 channels
  - 2500 entries with different power profiles
  - Synthetic input power profiles – not real on/off channels

- Dataset structure:

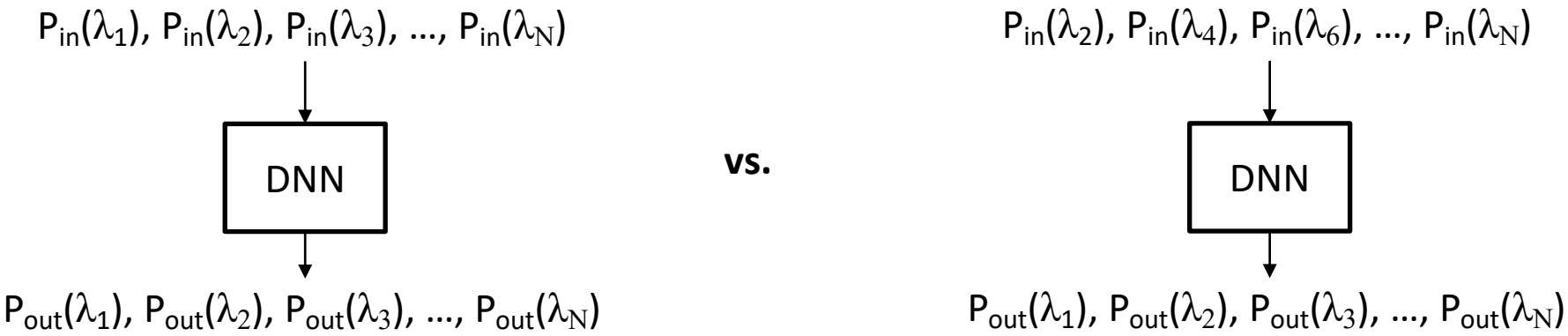
| Profile Id | Total Power In $EDFA_1$ | Total Power In $EDFA_2$ | Total Power In $EDFA_3$ | Total Power Out $EDFA_1$ | Total Power Out $EDFA_2$ | Total Power Out $EDFA_3$ | Input profile | | Output Profile | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | P. Ch. 1 | P. Ch. N | P. Ch. 1 | P. Ch. N |


[1] https://data.dtu.dk/articles/dataset/Input-output_power_spectral_densities_for_three_C-band_EDFAs_and_four_multispan_inline_EDFAd_fiber_optic_systems_of_different_lengths/13135754/1
[2] https://ieeexplore.ieee.org/document/9333297

# Project #1 – EDFA profile MIMO regression Assignment

- **Brief summary**:
  - Given input power profile, predict output power profile: **regression with multiple inputs and multiple outputs**
  - Use Deep Neural Network regression
  - Use Dataset 1

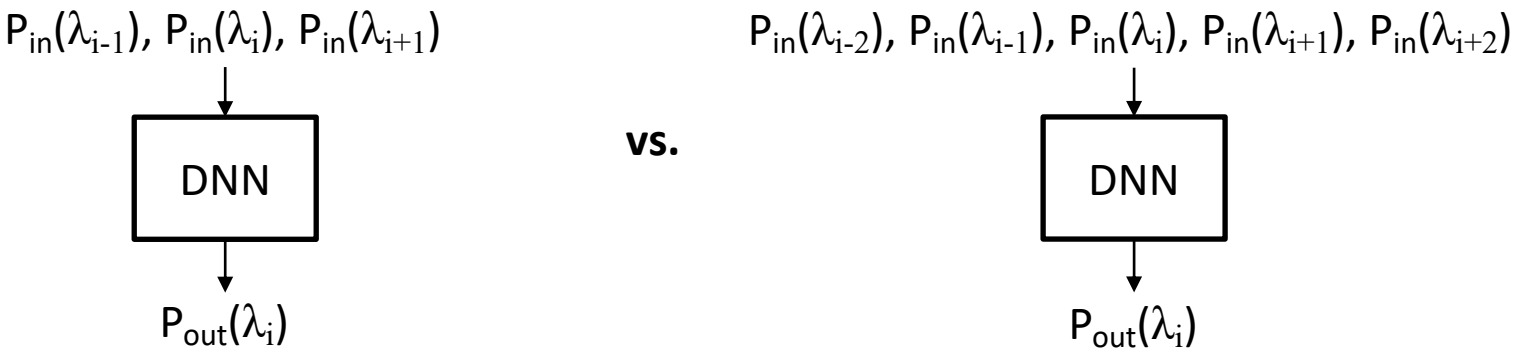
- How many samples of the input profile do we need to characterize EDFA behavior?
  - Sample input power profiles every 1st/2nd/5th/10th/… channel
  - Train regressors with different input dimensions to predict **full output power profile**
- **Compare the accuracy of output power profile prediction and model complexity**



# Project #2 – EDFA profile MISO regression Assignment

- **Brief summary**:
  - Given input power profile, predict output power for one channel: **regression with multiple inputs and single output**
  - Use Deep Neural Network regression
  - Use Dataset 1

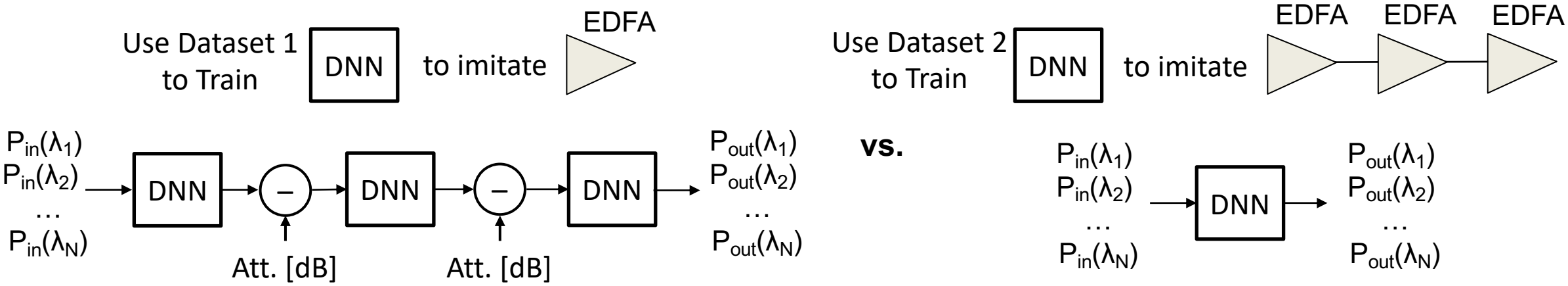
- How many samples do we need to predict output power for 1 channel?
  - Select **a random Channel Under Test (CUT)** (try channels in different parts of the spectrum)
  - Sample input power of 2/4/10/… channels neighboring CUT
  - Train the regressor with different input dimensions to predict **power of CUT**
- **Compare the accuracy of output power prediction and model complexity**



# Project #3 – EDFA profile multi-span prediction Assignment

- **Brief summary**:
  - Given input power profile, predict output power profile: **regression with multiple inputs and multiple outputs**
  - Use Deep Neural Network regression
  - **Use Dataset 1 and Dataset 2**

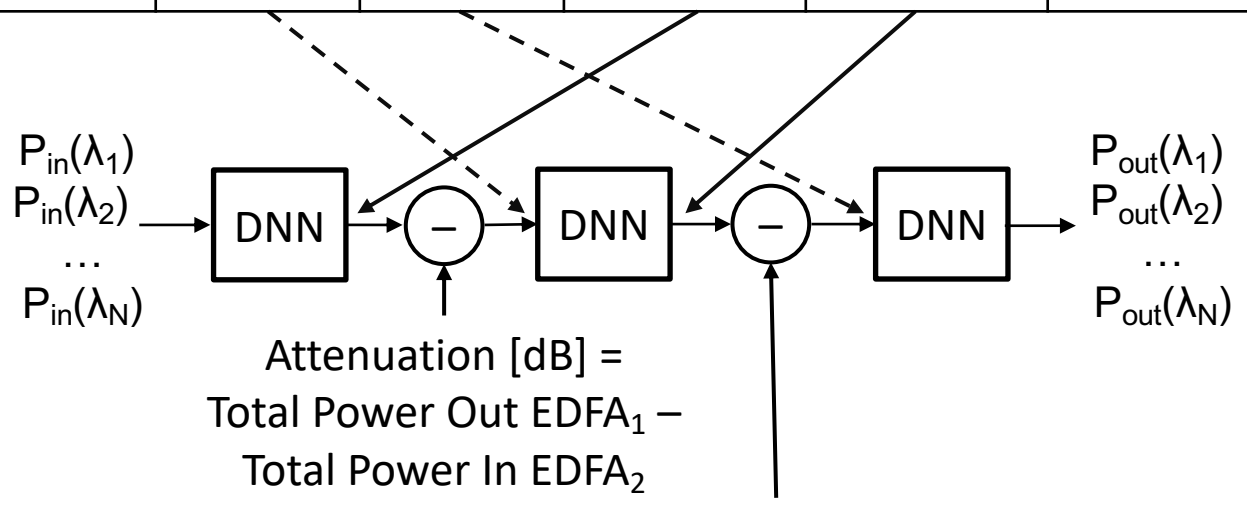
- How can we predict output power profile after 3 EDFAs with the highest accuracy?
  - Train a model to predict **full output power profile** of 1 EDFA and stack 3 models sequentially (use Dataset 1)
  - Train a model to predict **full output power profile** for a multi-span system (use Dataset 2)
  - Compare the accuracy of output power profile prediction and model complexity



# Project #3 – EDFA profile multi-span prediction
# Hint

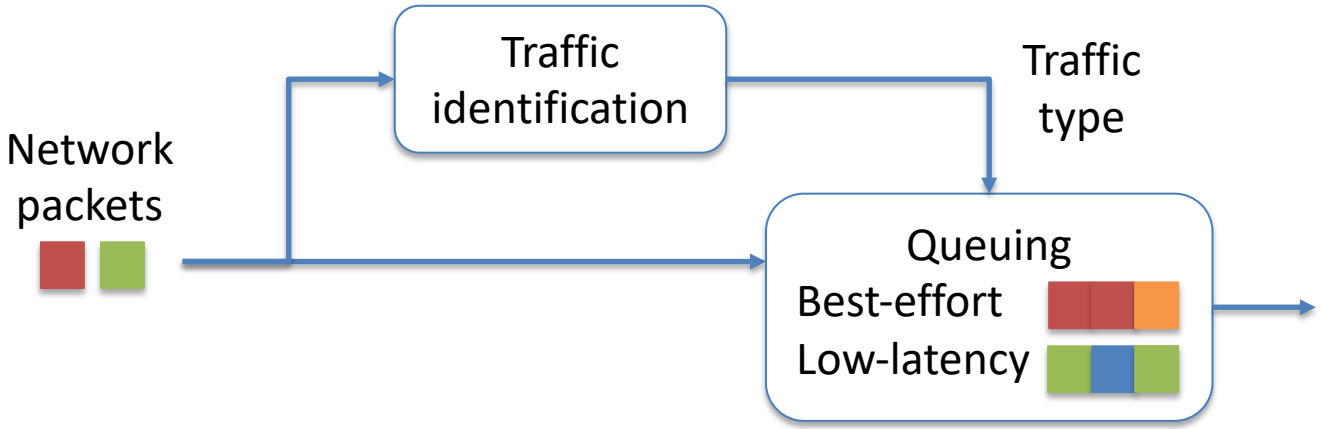
- Estimate fiber attenuation between the EDFAs in Scenario 1 using total input/output powers from the dataset
- Attenuation [dB] = Total Power Out $EDFA_i$ [dBm] – Total Power In $EDFA_{i+1}$ [dBm]

| Profile Id | Total Power In $EDFA_1$ | Total Power In $EDFA_2$ | Total Power In $EDFA_3$ | Total Power Out $EDFA_1$ | Total Power Out $EDFA_2$ | Total Power Out $EDFA_3$ | Input profile | | Output Profile | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | P. Ch. 1 | P. Ch. N | P. Ch. 1 | P. Ch. N |



# Project #4 – Application flow clustering
# Background

- Network operators may use **different priorities for different traffic types**
  - E.g., for congestion control, resource allocation, security [1]
- Otherwise, high-capacity applications (e.g., video-streaming) will use all the available link bandwidth, while latency-sensitive applications will struggle




- Can we automatically deduce what traffic types are there and what apps belong to each type?
- Can we use ML to **cluster** traffic flows of different applications?

[1] https://arstechnica.com/tech-policy/2017/11/comcast-throttling-bittorrent-was-no-big-deal-fcc-says/

# Project #4 – Application flow clustering
# Dataset

- IP network traffic flows for several applications (e.g., Teamviewer/Whatsapp/Youtube)
- 87 features: IP addresses, application ports, number of packets, flow duration, …

| Flow.ID | Source.IP | Source.Port | Destination.IP | Destination.Port | Protocol | Timestamp | Flow.Duration | … | L7Protocol | ProtocolName |
|---|---|---|---|---|---|---|---|---|---|---|
| 10.200.1.100-10.200.7.199-123-123-17 | 10.200.7.199 | 123 | 10.200.1.100 | 123 | 17 | 27/04/201708:48:11 | 845 | | 9 | NTP |
| 10.200.7.196-37.252.232.54-37184-443-6 | 10.200.7.196 | 37184 | 37.252.232.54 | 443 | 6 | 27/04/201708:50:44 | 185938 | | 148 | TEAMVIEWER |
| 172.16.255.200-10.200.7.4-53-52931-17 | 10.200.7.4 | 52931 | 172.16.255.200 | 53 | 17 | 27/04/201708:45:29 | 29999290 | | 5 | DNS |

Flow features

Labels

[1] https://figshare.com/articles/dataset/AppClassNet_-_A_commercial-grade_dataset_for_application_identification_research/20375580
[2] https://dl.acm.org/doi/10.1145/3561954.3561958

# Project #4 – Application flow clustering
# Dimensionality reduction

- Dimensionality reduction can sometimes improve the quality of clustering [1]
- The **goal is to retain as much "structure"/information in the data as possible**
  - Points that are far away from each other in the original high-dimensional space are also far away in the low-dimensional space, and vice-versa for points close to each other

There are different linear and nonlinear data transformations:

- **PCA** – transforms the data into uncorrelated principal components, which contain decreasing amounts of information as measured by variance
- **t-SNE** – finds a low-dimensional embedding in which a relative distance between points i and j matches with the one in the original high-dimensional space
- **PaCMAP** – finds a low-dimensional embedding using three kinds of pairs of points: neighbors, mid-near points and further points


[1] https://www.cs.cornell.edu/courses/cs4780/2022fa/slides/curse_of_dim_clustering_annotated.pdf

[2] https://scikit-learn.org/stable/modules/manifold.html#manifold

[3] https://github.com/YingfanWang/PaCMAP

# Project #4 – Application flow identification Assignment

- **Brief summary**:
  - Given flow features as input, cluster the applications: **clustering ML problem**

- (12 points*) Compare different clustering algorithms (e.g., KMeans, DBSCAN)
  - Experiment with hyperparameters (e.g., number of clusters in KMeans, distance metrics in DBSCAN)
  - Which apps are clustered together?
    - E.g., you find that 50% of YouTube flows are clustered together with 20% of OneDrive flows
  - What are the similarities between features of the flows that are grouped together?

- (3 points*) Advanced task
  - Evaluate the effect of dimensionality reduction on the above questions
    - Use at least 2 dimensionality reduction algorithms (e.g., PCA, t-SNE, PaCMAN)

* No advanced task among federated learning, transfer learning, XAI as they are not applicable to this project

# Project #5 – QoT prediction in optical networks
## Background

- Optical signals can be characterized by a Quality of Transmission (QoT) metric (e.g., Signal-to-Noise Ratio)
- SNR is used to configure the Modulation Format (MF)

- In network planning we must assign MFs before launching the signal into the network and measuring SNR
- We also must choose between different candidate paths for the signal
- SNR along different paths is different, as there are more/less optical amplifiers and interfering channels

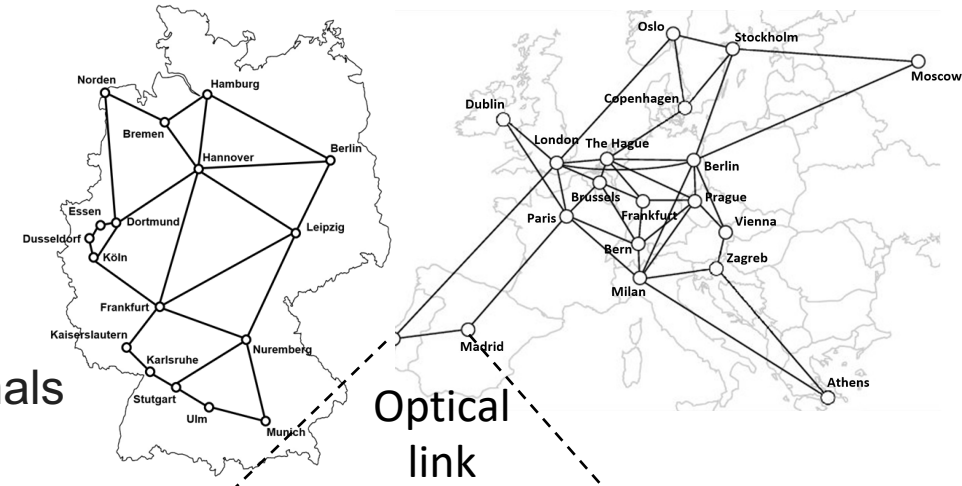
- **How can we predict SNR?**

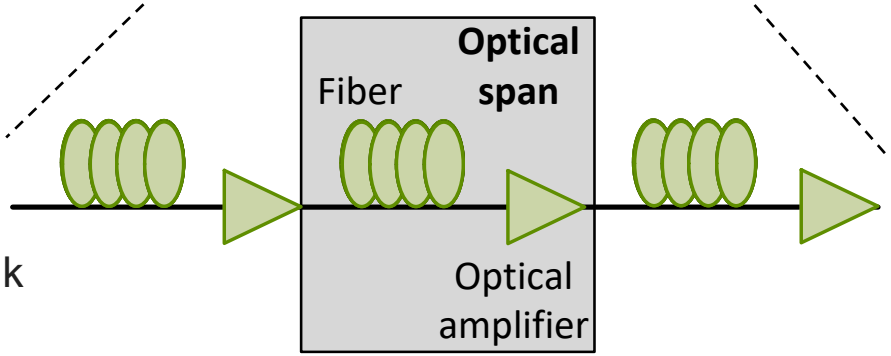
| **SNR prediction** | Accuracy with imprecisely known path parameters | Margins (underutilization of network resources) | Must be trained on SNR measurements from the network |
|---|---|---|---|
| **Analytical model** | Low | High | No |
| **ML estimator** | High | Low | Yes |

# Project #5 – QoT prediction in optical networks Dataset

- Simulated dataset
- 17-node German and 21-node European networks
- Analytical model used to estimate interference between optical signals
  - The closer is the other signal, the higher is the interference
    - First Fit spectrum allocation: neighboring channel is usually occupied
    - Distance to the closest interferer is the same for all channels
  - The more signals in the fiber, the higher is the interference

- Dataset structure:
  - Optical links: fiber connecting any two nodes in the network
  - Optical spans: [60-80 km fiber span + amplifier] segment along the link

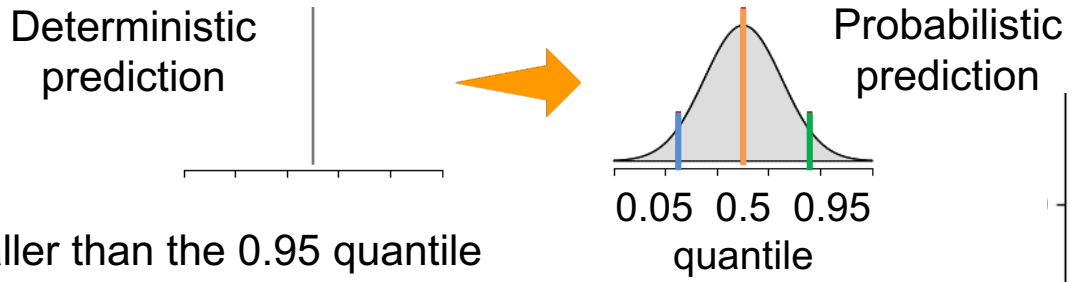


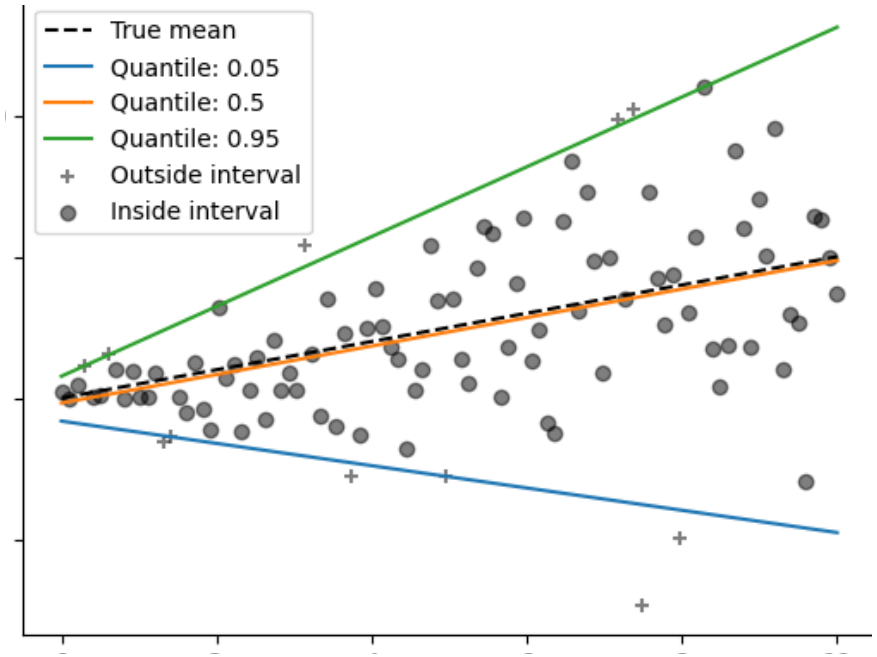




| Path features | | | Interference features | | | Label |
|---|---|---|---|---|---|---|
| Length (in km) of fiber in span 1 | … | Length (in km) of fiber in span N | Number of channels in link 1 | … | Number of channels in link M | SNR, dB |

# Project #5 – QoT prediction in optical networks
# Probabilistic regression

- Instead of predicting a single value in regression, we can predict a distribution of values
- There are multiple approaches: distribution regression, Bayesian NN and *quantile regression*




- 95% of samples are smaller than the 0.95 quantile
- In quantile regression, we modify the loss function to
  - penalize underestimations – high quantiles – less conservative prediction

  OR
  - penalize overestimations – low quantiles – more conservative prediction

- Implemented in scikit-learn for GradientBoostingRegressor:

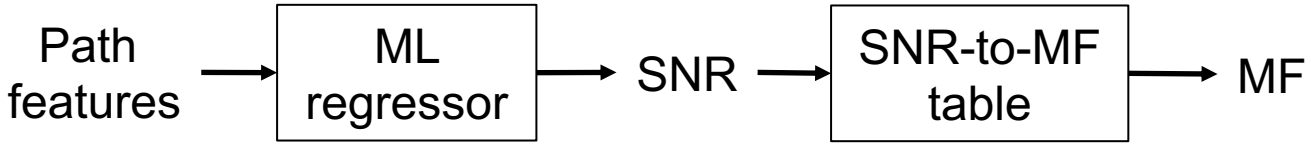
**loss**, ***default='squared_error'***
'quantile' allows quantile regression (alpha specifies the quantile)
**alpha**, ***default=0.9***
The alpha-quantile of the quantile loss function. Only if loss='quantile'. In the range (0.0, 1.0)

[1] https://scikit-learn.org/stable/auto_examples/ensemble/plot_gradient_boosting_quantile.html#sphx-glr-auto-examples-ensemble-plot-gradient-boosting-quantile-py
[2] https://ieeexplore.ieee.org/document/9355394

# Project #5 – QoT prediction in optical networks
# Assignment

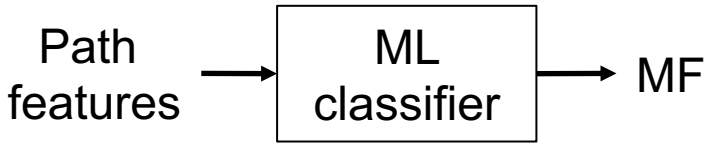
- **Brief summary**:
  - Given path features as input, choose MF: **regression or multi-class classification**
  - Use a tree model (e.g., GBT, LightGBM) as both regressor and classifier

1) Use a regressor to predict SNR, then map the MF




- Use probabilistic regression and assign MF based on low/high-quantile estimations of SNR

2) Use a classifier to predict MF




- **Compare the number of MF-over/under-estimations with the two approaches**

| MF | Required SNR, dB |
|---|---|
| QPSK | 8.7 |
| 8QAM | 12.8 |
| 16QAM | 15.2 |
| 32QAM | 18.2 |
| 64QAM | 21 |

# Project #6 – Optical failure localization
## 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)
  - Trigger early network reconfiguration (*proactive* procedures)

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)*

# Project #6 – Optical failure localization
# Background

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?

Project's focus

4. **Magnitude estimation** (How much?)
   - Triggers the proper reaction

   (e.g., device restart/reconfiguration, lightpath re-routuing, in-field reparation…)

# Project #6 – Optical failure localization
## Background

- Optical links can be many hundreds of km, so it is very important to quickly localize faults

- If there is extra attenuation in the span – amplifier gain will automatically increase[1]
- Change in the gain means that a different amount of amplifier noise is added to the signal
- Corresponding change in Optical SNR is detectable by monitoring devices

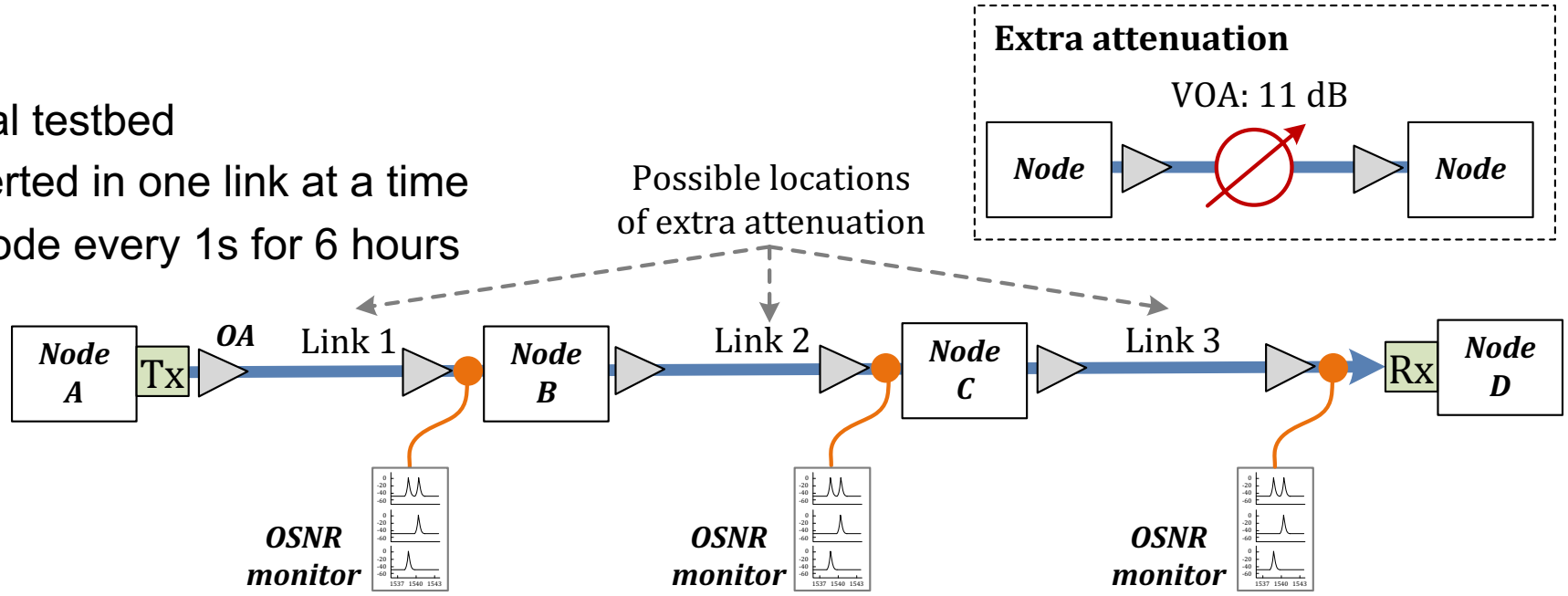
- OSNR will change differently for different span lengths and amounts of extra attenuation
- However, fault localization must work well for different networks and fault magnitudes
- One solution is to remove the mean component from the data and use only statistical variations of OSNR

1) **Can we use ML to localize extra attenuation in the optical line based on statistical variations of OSNR?**
2) **Can we benefit from having monitoring devices along the line and not only one at the receiver?**

[1] Of course, extra attenuation can also be localized by the increased gain of the amplifier, but where is fun in that? ☺

# Project #6 – Optical failure localization Dataset

- 3-span optical line of an optical testbed
- 11 dB extra attenuation is inserted in one link at a time
- OSNR is monitored at each node every 1s for 6 hours




```
AUTHENTICATE CRAM-MD5.
ready
ready
*OPC is OK (1).
####    2019/07/26 18:47:19 558,1-PreAMP
 DATE   ,PK_WL[THZ],LEVEL[dBm],3.0dB WD[nm],CRT WL WL[NM],3.0dB PB[nm],RIPPLE[dB],CROSS TK[L][dB],CROSS TK[R][dB],OFFSET WL[n
m],OFFSET LEVEL[dBm],NOISE[dBm][NBW],OSNR[dB]
2019/07/26 18:47:19 558,194.800,-20.897,236702579,194800939,374929661,25.210, 8.649, 8.636, 0.000, 0.000,-46.876,25.978
2019/07/26 18:47:19 558,194.800,-20.906,241765735,194800940,373790320,25.125, 8.710, 8.682, 0.000, 0.000,-46.864,25.955
2019/07/26 18:47:19 558,194.800,-20.812,234171031,194800950,373663765,25.175, 8.756, 8.744, 0.000, 0.000,-46.864,25.997
2019/07/26 18:47:19 558,194.800,-20.862,235436805,194800948,374676501,25.234, 8.639, 8.592, 0.000, 0.000,-46.861,25.996
```

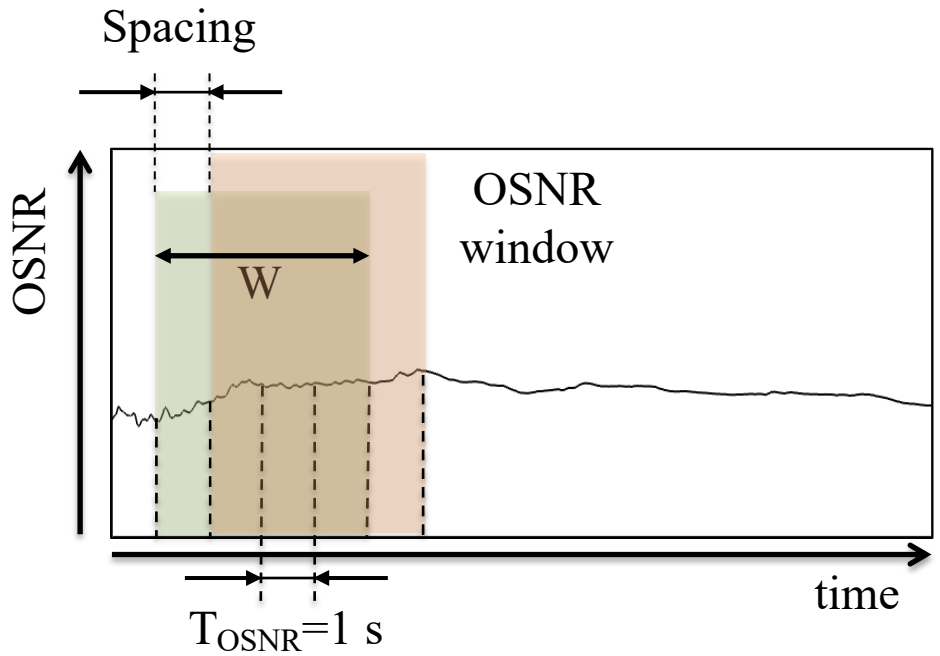
Data collected by OSNR monitor

Timestamp

OSNR, dB

# Project #6 – Optical failure localization
# Reminder on dataset preprocessing

1. Remove mean component from the OSNR signal, using normalization, standardization or differencing (one of your choice)
2. Form “windows” of duration W, including W consecutive OSNR samples
3. Compute the features of each window:
   - Mean
   - Standard deviation
   - Maximum value
   - Minimum value
   - Peak-to-peak $(max - min)$
   - Other features of your choice?



# Project #6 – Optical failure localization
# Assignment

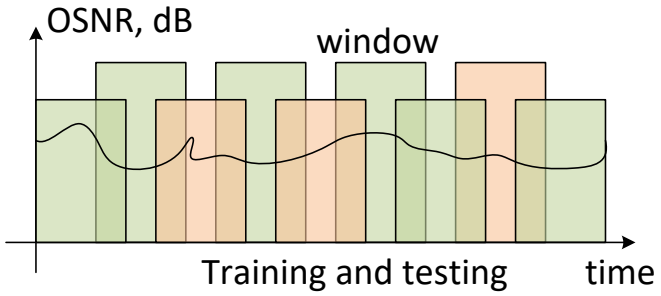
- **Brief summary**: given OSNR measurements as input, localize the fault: **multi-class classification**
- Use a tree classifier (e.g., Random Forest, XGBoost, LightGBM)

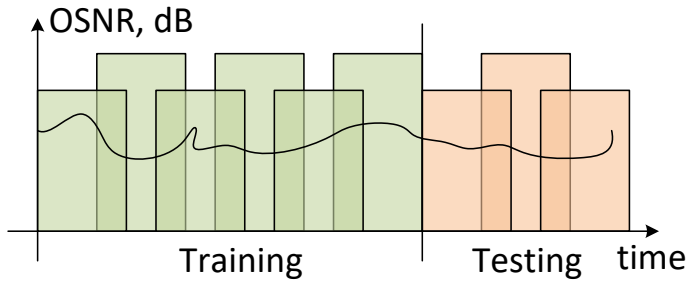
- We can use only OSNR at the receiver (single monitor) or OSNR from all monitors (multiple monitors)

| X | y |
|---|---|
| Features of OSNR windows measured at link 3 | Fault location |

| X | | | y |
|---|---|---|---|
| Features of OSNR windows measured at link 1 | Features of OSNR windows measured at link 2 | Features of OSNR windows measured at link 3 | Fault location |

- We can split data into training and testing datasets randomly (random split) or in time (temporal splits):




Compare localization accuracy in 4 scenarios:

| **Monitor/ Splitting** | Random | In time |
|---|---|---|
| Single | Scenario 1 | Scenario 2 |
| Multiple | Scenario 3 | Scenario 4 |

# Project #7-8 – Traffic forecasting
# Background

- Network traffic changes during the week and even the day
- Processing and transporting resources can be scaled
  - increase/decrease number of VMs
  - turn transceiver/antenna on/off
- This can be done statically or dynamically
  - **Statically**, based on peak traffic: over-dimensioning
  - **Statically,** based on average traffic: some users will have reduced service quality at peak times
  - **Dynamically**, if accurate traffic estimation is available

| Resources allocation based on… | Electricity consumption in 1 day | Service provisioning ratio |
|---|---|---|
| Peak traffic | $E_{peak} = 24 * 100 * P$ | $\cong 100\%$ |
| Average traffic | $E_{avg} = 24 * 10 * P$ | $<< 100\%$ |
| Traffic prediction | $E_{avg} \leq E_{pred} << E_{peak}$ | $\cong 100\%$ |

- **Can we use ML to predict traffic and reduce the number of active transceivers?**

# Project #7-8 – Traffic forecasting
# Dataset

- GÉANT is the research network that carries traffic between universities and research institutions in Europe
- GÉANT is composed of 23 routers connected with 38 links
- GÉANT uses SONET technology to multiplex traffic with different bitrates into one optical signal
- Channel with the smallest bitrate that can be created in SONET is 50 Mbit/s

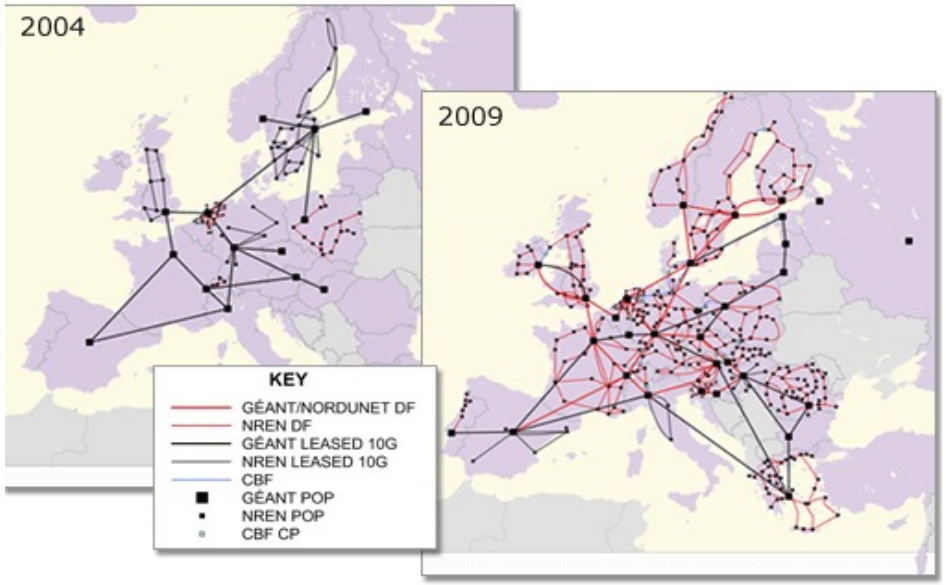
- Each file in the dataset describes total traffic in kbit/s between pairs of routers [1, 2]

Source node **11**

Data rate in **kbit/s**

```
<src id="11">
    <dst id="12">432392.0533</dst>
    <dst id="13">1623.2978</dst>
    <dst id="19">4221.3689</dst>
    <dst id="23">378.0622</dst>
```

Destination nodes **12, 13, 19, 23**




- Dataset includes 2941 files: traffic at 15 min intervals for 1 month

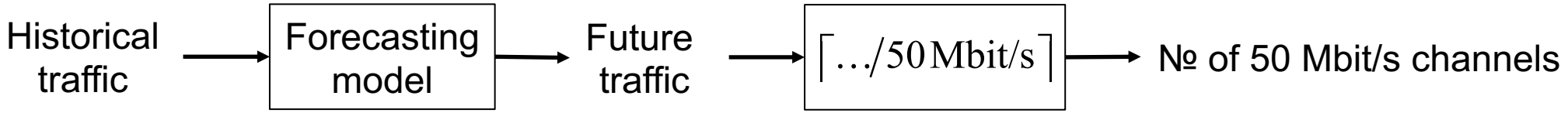

[1] https://totem.info.ucl.ac.be/dataset.html
[2] https://dl.acm.org/doi/10.1145/1111322.1111341

# Project #7 – Traffic forecasting
# Assignment

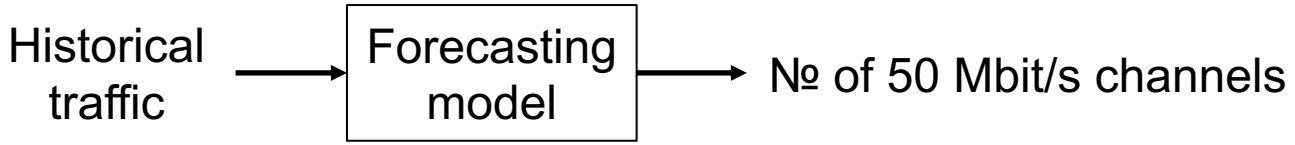
- **Brief summary**: Given traffic sequences as input, forecast number of 50 Mbit/s channels in the future
- Use LSTM network

- Choose a subset of source-destination pairs

1) Forecast traffic between two nodes, then map it into the number of 50 Mbit/s channels




2) Forecast the number 50 Mbit/s channels needed to carry traffic between 2 nodes




- **Compare the number of over/under-estimations of the number of 50 Mbit/s channels and model complexity**

# Project #8 – Traffic forecasting: local vs. global Assignment

- **Brief summary**: Given traffic sequences as input, forecast future traffic
- Use LSTM network

- Choose a subset of source-destination pairs

- Forecast traffic between two nodes in kbit/s
  - Use a local model for each source-destination pair
  - Use a single global model for all source-destination pairs
- **Compare the accuracy of the forecast (future traffic between nodes used for training) and model complexity**

Train [Model A-B] to forecast traffic between nodes A and B

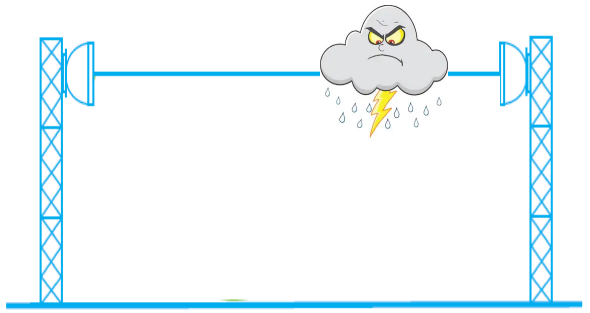
Train [Model C-D] to forecast traffic between nodes C and D

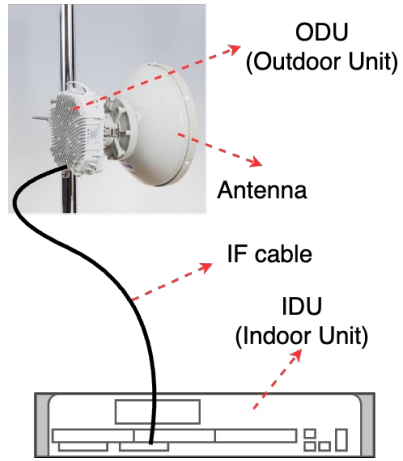
**vs.** Train [Model] to forecast traffic between nodes A and B, C and D

# Project #9 and #10 - Data-driven ML models: the more, the merrier?!

## **Background**: a microwave hardware failure use-case

- Failures in microwave networks
  1) Propagation failures
     - e.g., due to atmospheric changes
  2) **Hardware failures**
     - **e.g., due to transmission equipment failure**





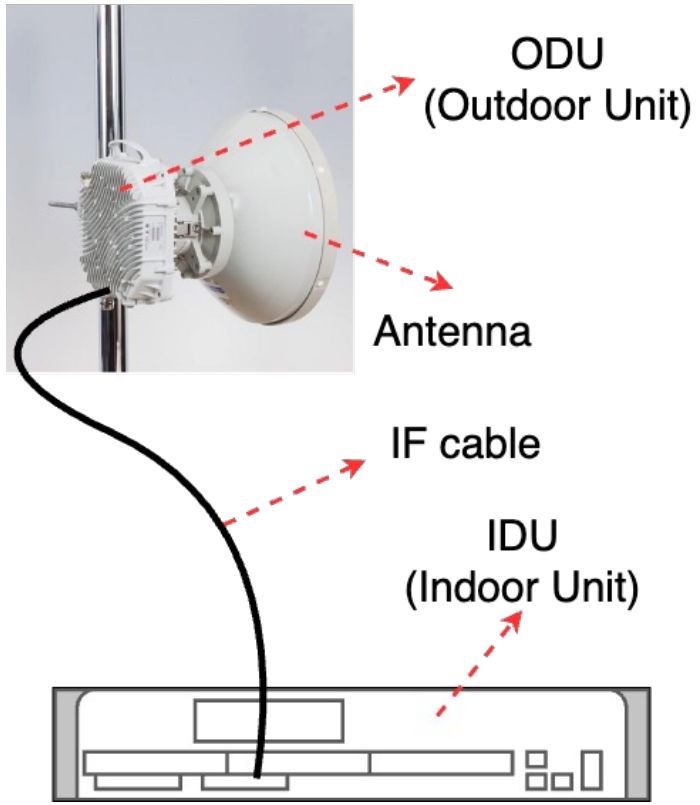

- How does a microwave engineer determine if there is a hardware failure?
  - For each microwave link, the Network Management System (NMS) records equipment alarms
    - For each link, for each 15-minute window, a value [0, 900] seconds is recorded for each alarm

| | alarm_0 | alarm_1 | alarm_2 | alarm_3 | alarm_4 | alarm_5 | alarm_6 | alarm_7 | alarm_8 | alarm_9 | ... | alarm_155 | alarm_156 | alarm_157 | alarm_158 | alarm_159 | alarm_160 | alarm_161 | alarm_162 | alarm_163 | label |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 1664 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 28 | 0 | 0 | 3 |
| 1665 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 1666 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 24 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1667 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 1668 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 388 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |

1669 rows × 165 columns

# Project #9 and #10 - Data-driven ML models: the more, the merrier?!
## **Dataset and assignment**: a microwave hardware failure use-case

- Four hardware failure types
  1. Failure class 0 (IDU failure): 515 observations (data points)
  2. Failure class 1 (ODU failure): 611 observations
  3. Failure class 2 (Cable failure): 207 observations
  4. Failure class 3 (Power failure): 336 observations




- What can we observe in the hardware failure dataset?
  1. The amount of labelled data is quite poor (only 1669 data points)
  2. In real-world datasets, the frequency of failures is not uniform

- Project questions:
  1. How can we use the data we have to generate synthetic data?
  2. Does more data mean better ML model performance?

# Project #9 and #10 - Data-driven ML models: the more, the merrier?! Assignment

- How can we address the project questions:
  1. Can we use the data we have to generate more data?
  2. Does more data mean better ML model performance?
- We define three approaches to generate more data
  1. SMOTE (Synthetic Minority Oversampling Technique) - #9 & #10
  2. VAE (Variational Autoencoder) - #9
  3. GAN (Generative Adversarial Networks) - #10

In both projects #9-#10:

- We define two scenarios of data generation
  1. DataGen-1: Rebalancing of the dataset
  2. DataGen-2: Dealing with extreme data imbalance
- We can use two feature sets
  1. Binary
  2. Categorical

Project #9: Team BLUE

| Project 9 | DataGen-1 | DataGen-2 |
|---|---|---|
| SMOTE | ✓ | ✓ |
| VAE | ✓ | ✓ |

Project #10: Team RED

| Project 10 | DataGen-1 | DataGen-2 |
|---|---|---|
| SMOTE | ✓ | ✓ |
| GAN | ✓ | ✓ |

# Project #9 and #10 - Data-driven ML models: the more, the merrier?!
## **Assignment:** Compare two feature sets

- Binary dataset
  - IF alarm value X > 0, then SET alarm value equal to X = 1
  - IF alarm value X = 0, then KEEP the alarm value as X = 0
- Categorical dataset
  - IF alarm value X = 0, then KEEP the alarm value as X = 0
  - IF alarm value is $0 < X \leq 45$, then SET the alarm value as X = 1
  - IF alarm value $45 < X \leq 450$, then SET the alarm value as X = 2
  - IF alarm value X > 450, then SET the alarm value as X = 3

# Project #9 and #10 - Data-driven ML models: the more, the merrier?!
## **Assignment:** Compare two data generation scenarios (DataGen1 vs DataGen2)

- **DataGen-1: Rebalancing the dataset**
  - Increase the number of data points for each class such that the dataset is balanced

1. Class 0 (IDU failure): 515 data points + 96 data points = 611 data points
2. Class 1 (ODU failure): 611 data points + 0 data points = 611 data points
3. Class 2 (Cable failure): 207 data points + 404 data points = 611 data points
4. Class 3 (Power failure): 336 data points + 275 data points = 611 data points

- **DataGen-2: Dealing with extreme data imbalance**
  - Remove X% of the data points of class Y from the training set
    - E.g., Remove 80% of data points from Class 2 from the training set
  - On the new reduced dataset, generate synthetic data (SMOTE/VAE/GAN)
    1. Rebalance the dataset (as in DataGen-1)
    2. Add a fixed number of datapoints per class (e.g., add 300 data points per each class)

# Project #9 and #10 - Data-driven ML models: the more, the merrier?!
## **Assignment:** Evaluation

1) Training and testing
   - Base model: train on real data, test on real data
   - Mixed model: train on mixed (real + synthetic) data, test on real data
2) Feature format
   - Binary dataset
   - Categorical dataset
3) Data generation scenario
   - DataGen-1: rebalancing the dataset
   - DataGen-2: dealing with extreme data imbalance
4) Evaluation metrics
   - F1-score: global and per-class
   - Accuracy: global and per-class
   - Any other metric learned during the course

Project #9: Team BLUE

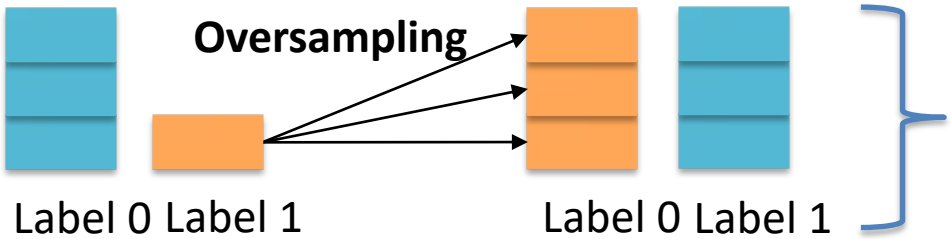
| Project X.1 | DataGen-1 | DataGen-2 |
|---|---|---|
| SMOTE | ✓ | ✓ |
| VAE | ✓ | ✓ |

Project #10: Team RED

| Project X.2 | DataGen-1 | DataGen-2 |
|---|---|---|
| SMOTE | ✓ | ✓ |
| GAN | ✓ | ✓ |

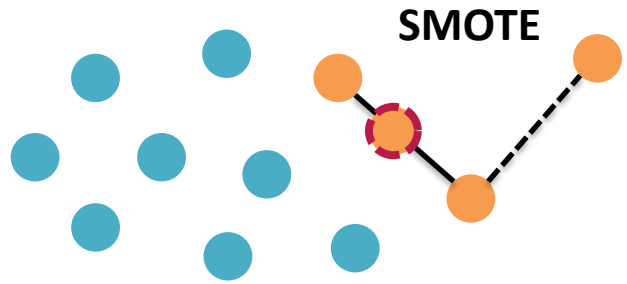
# Project #9 and #10 - Data-driven ML models: the more, the merrier?!
# SMOTE for data generation

- Oversampling: duplicate samples from the minority class in the training dataset




- Balanced class representation
- No additional information to the model

- SMOTE (Synthetic Minority Oversampling Technique )
  - Choose a random sample from the minority class
  - Find *k* nearest neighbors for that sample (typically *k*=5)
  - Choose a random neighbor out of *k*
  - Create a new sample at a random point between two samples

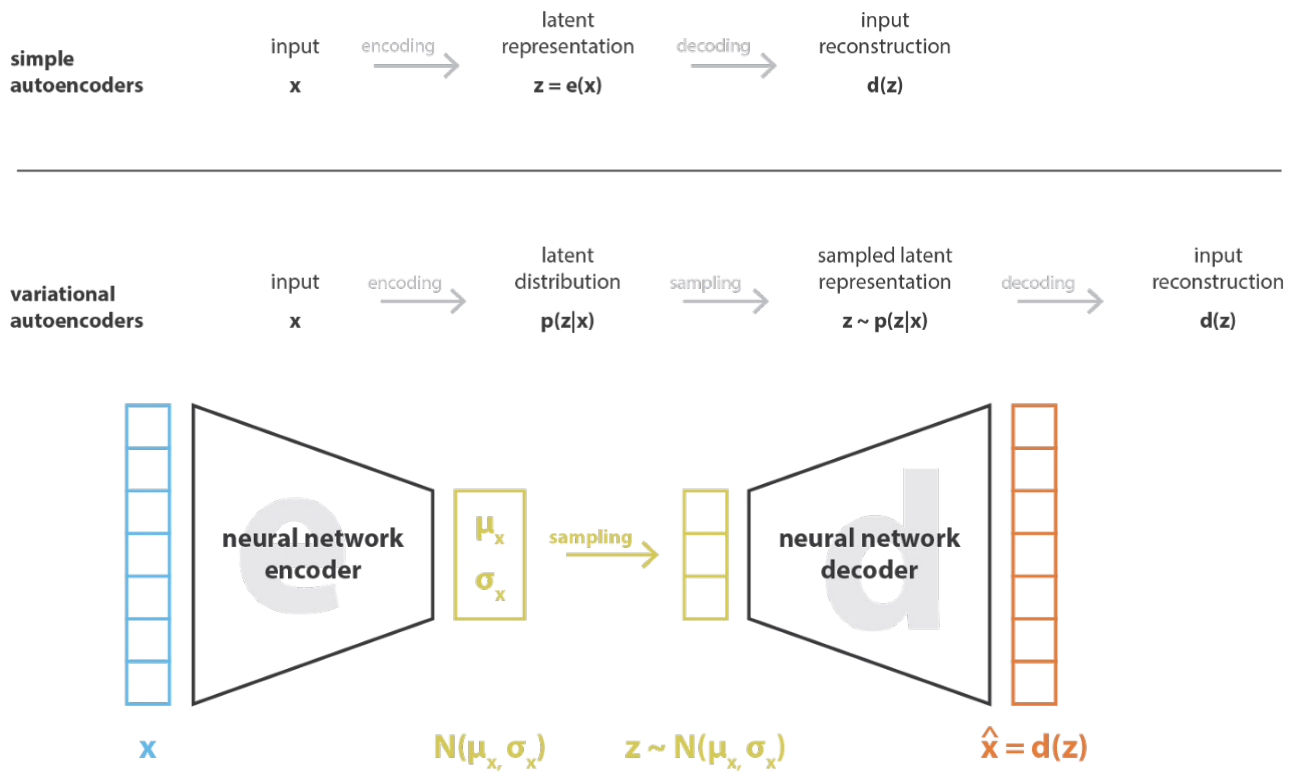



- Many variations of SMOTE, e.g., BorderlineSMOTE [2] or ADASYN [3]


[1] https://arxiv.org/abs/1106.1813
[2] https://link.springer.com/chapter/10.1007/11538059_91
[3] https://ieeexplore.ieee.org/document/4633969

# Project #9 - Data-driven ML models: the more, the merrier?!
# Variational Autoencoder (VAE)

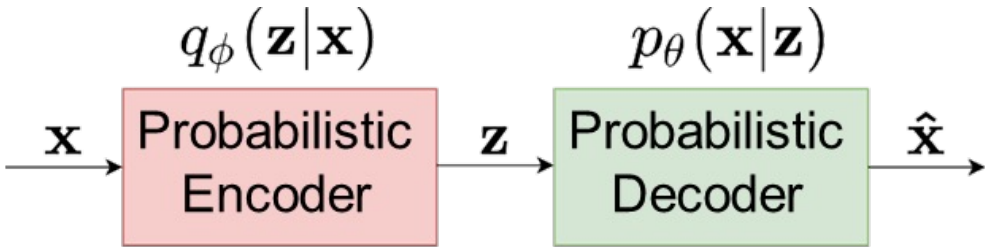
- **VAE** is an Autoencoder whose training is regularized and ensure that the latent space has good properties that **enable a generative process**



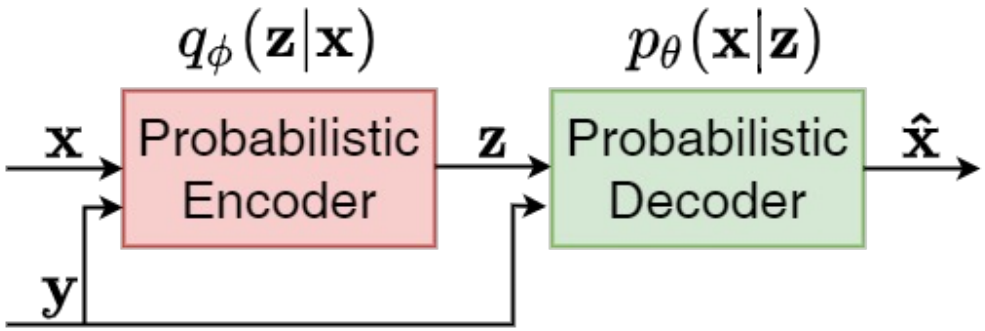
# Project #9 - Data-driven ML models: the more, the merrier?!
# Conditional Variational Autoencoder (CVAE)

- VAE learns a statistical model pθ (x|z) of the input data
- Conditional VAE (CVAE) allows to control the data generation process
  - CVAE models latent variables and data, conditioned to some random variable(s)

VAE



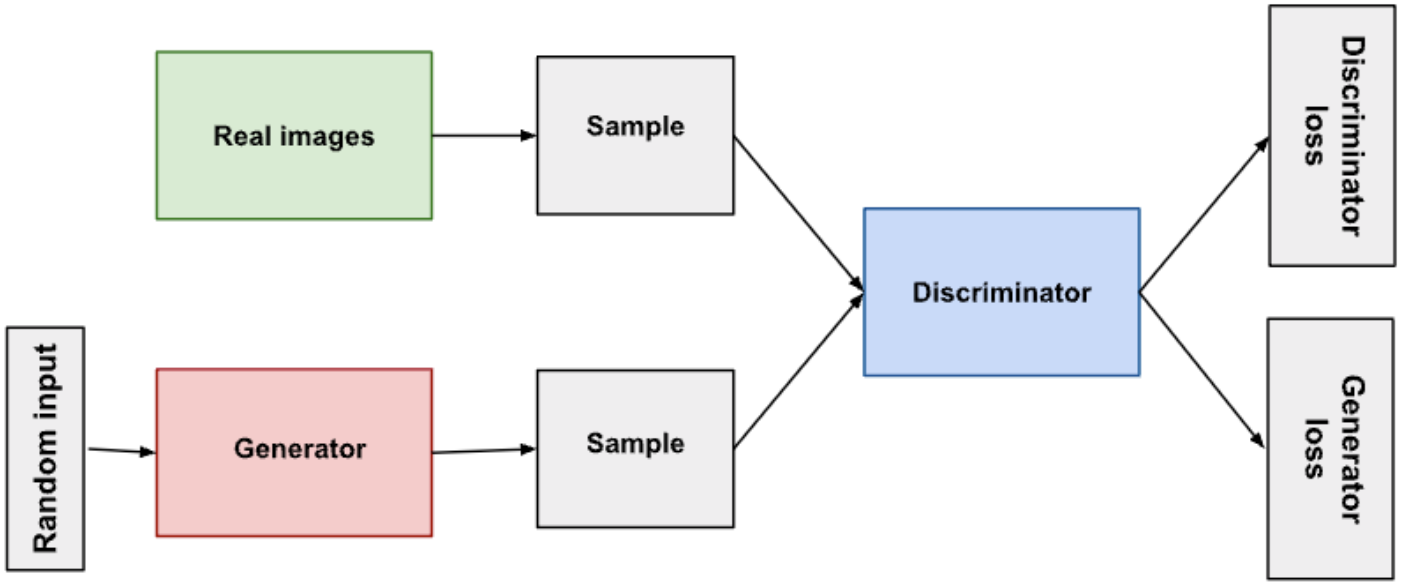

CVAE




References:


[1] https://arxiv.org/abs/1312.6114
[2] https://arxiv.org/abs/1906.02691
[3] GitHub link

# Project #10 - Data-driven ML models: the more, the merrier?!
# Generative Adversarial Networks (GAN)




- GAN structure is composed of **the generator** and **the discriminator**
  - Both are Neural Networks
  - The **generator** learns to generate plausible data
  - The **discriminator** learns to distinguish the generator's fake data from real data
    - Through backpropagation, the discriminator's classification provides a signal that the generator uses to update its weights
    - The discriminator in a GAN is simply a classifier: it aims to distinguish real data from the data created by the generator

# Project #10 - Data-driven ML models: the more, the merrier?! Generative Adversarial Networks (GAN)

- Various versions of GANs, depending on the application
- Conditional Tabular Adversarial Networks (CTGAN)
  - Explicitly designed for handing tabular data
  - A key feature of CTGAN is the capability to generate conditioned data
    - E.g., generate data on a specific class

- References:
  - [1] https://arxiv.org/abs/1406.2661
  - [2] https://arxiv.org/abs/1907.00503
  - [3] https://github.com/sdv-dev/CTGAN
  - [4] https://sdv.dev/SDV/user_guides/single_table/ctgan.html