



POLITECNICO
MILANO 1863



Network Data Analysis Laboratory

Proposed projects

Francesco Musumeci

Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB)

Politecnico di Milano, Milano, Italy

2024-2025

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, promising features, redundancy, ...)?
 2. Data preprocessing (e.g., generate «ordered» samples from raw data)
 3. ML models optimization and training
 - Hyperparameters tuning with cross-validation
 4. Testing the performance and evaluating different scenarios (**see specific projects assignment**)



General requirements (2/2) – Methodology

- All projects **must** include (for 12 points)
 - Specific project assignment (see later)
 - 2 or 3 ML algorithms (usually recommended in the assignment)
 - Different performance metrics: MSE, MAE, Accuracy, Precision, Recall, F-score, training duration, ...
- Additionally, projects **may** include (at least) one "advanced" task (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
 - Project-specific advanced task



Projects list

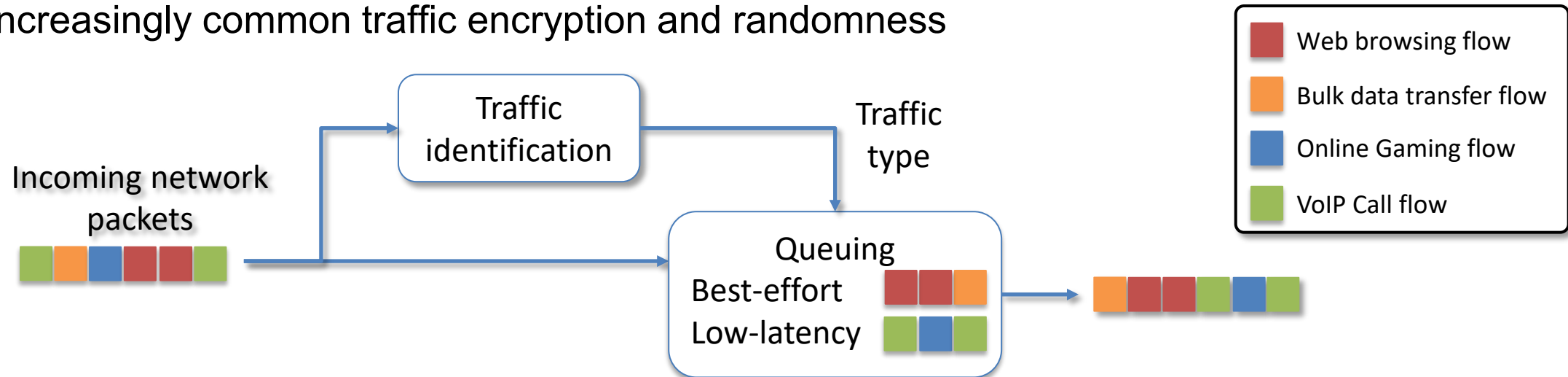
1. Network Traffic Analysis – Clustering
2. Network Traffic Analysis – Application Classification & Activity Classification with TL
3. Network Traffic Analysis – Activity Classification & Application Classification with FL
4. Packet Loss Event Classification with TL
5. Packet Loss Event Classification with FL
6. Weather-based performance prediction of satellite-internet provider
7. EDFA profile MIMO regression
8. EDFA profile MISO regression
9. EDFA profile multi-span regression
10. QoT estimation in optical networks
11. Optical failure localization: single-step classification
12. Optical failure localization: two-step classification



Project #1-2-3 – Network Traffic Analysis: overall goal

Motivation

- Home routers may use different priorities for different traffic types to intelligently queue the outbound Internet traffic. This can significantly improve the user's quality of experience of delay-sensitive application (e.g. online gaming or VoIP¹ traffic) that otherwise would struggle with in presence of high-capacity applications.
- Classifying the **application** or, more generally, the **activity** performed, is not trivial due to increasingly common traffic encryption and randomness



Can we infer “application” identity and “activity”, and understand their behaviour solely from analyzing encrypted network traffic patterns (packet sizes, timings, flow statistics) to categorize traffic?

¹VoIP: Voice over IP



Project #1-2-3 – Network Traffic Analysis

The Dataset [1] [2]

- **Apps:** Collection of smartphone traffic from 20 apps¹ over different 2244 sessions (≈ 490 hours)
- **Activities:** Each app was used to perform at least one activity (*Chat, Audio-call, Online Gaming, Video-call, Video-streaming*)
- **Sessions:** Each capture session (7~80 min) contains the traffic generated from a target app performing **one or more activities** sequentially
- **Dataset Structure:** Each session corresponds to a JSON file on the dataset, it contains three types of traffic data (Per-packet data, Per-flow features and Per-flow metadata) for each bidirectional sequence of packets sharing the L4 quintuple (IPs, L3 proto and L4 ports)

Statistical features extracted from the complete bidirectional flow (upstream, downstream, complete IP packet lengths and inter-arrival times)

¹ Clash Royale, Crunchyroll, Discord, Goto Meeting, Jitsi Meet, Kakao Talk, Line, Meet, Messenger, Omlet, Signal, Skype, Slack, Teams, Telegram, Trueconf, Twitch, Webex, WhatsApp, Zoom

[1] <https://ieeexplore.ieee.org/abstract/document/10770459>

[2] <https://traffic.comics.unina.it/mirage/mirage-2024.html>

Data	Name	Description
Per-packet Data	timestamp	Timestamp expressed as Unix Epoch time
	src_port	Source transport-layer port
	dst_port	Destination transport-layer port
	packet_dir	Packet direction (0 upstream, 1 downstream)
	IP_packet_bytes	Number of bytes in IP payload
	IP_header_bytes	Number of bytes in IP header
	L4_header_bytes	Number of bytes in L4 header
	L4_payload_bytes	Number of bytes in L4 payload
	iat	Inter-arrival time
	TCP_win_size	TCP window size (0 for UDP packets)
Per-flow Features	TCP_flags	TCP flags (empty for UDP packets)
	L4_raw_payload	Byte-wise raw L4 payload (integer $\in [0, 255]$)
	min	Minimum
	max	Maximum
	mean	Arithmetic mean
	std	Standard deviation
	var	Variance
	mad	Mean absolute deviation
	skew	Unbiased sample skewness
	kurtosis	Unbiased Fisher kurtosis
Per-flow Metadata	q_percentile	q^{th} percentile ($q \in [10 : 10 : 90]$)
	BF_device	MAC address of the mobile device
	BF_label	Android-package name
	BF_activity	Activity type
	BF_app_category	Android-package category name ‡
	BF_label_version_code	Android-package version code
	BF_label_version_name	Android-package version name
	BF_labeling_type	Exact or most-common labeling
	{BF, UF, DF}_num_packets	Number of packets
	{BF, UF, DF}_IP_packet_bytes	Total bytes in IP packets
	{BF, UF, DF}_L4_payload_bytes	Total bytes in L4 payloads
	{BF, UF, DF}_duration	(Bi)flow duration in seconds
	{UF, DF}_MSS	TCP Maximum Segment Size (0 for UDP flows)
	{UF, DF}_WS	TCP Window Scale factor (0 for UDP flows)

‡ Category provided by the Google Play Store.



Project #1 – Network Traffic Analysis

Clustering

- **Assignment**: Given packet and/or flow features as input, cluster the application and activities without considering the labels (**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?
 - Are apps performing the same activity clustered together?
 - Try to perform clustering only with sessions associated to a specific activity. Are apps grouped in different clusters? Does anything change if you change the activity used to filter?
- (3 points) Advanced task
 - Evaluate the effect of dimensionality reduction
 - Re-do clustering after performing dimensionality reduction and address the above questions again
 - Use at least 2 dimensionality reduction algorithms (e.g., PCA, t-SNE, PaCMAP)



Project #1 – Network Traffic Analysis

Clustering – Dimensionality Reduction Overview

Some pointers on
dimensionality reduction

- The dataset contains numerous statistical features per flow. High dimensionality can hinder clustering performance ("curse of dimensionality") and interpretation.
- The **goal is to reduce the number of features while preserving the most important information for distinguishing apps/activities.**
 - 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** [1] – transforms the data into uncorrelated principal components, which contain decreasing amounts of information as measured by variance
- **t-SNE** [2] – 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** [3] – 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 #2 – Network Traffic Analysis

Application Classification + Transfer Learning

- (12 points) Perform **Applications classification** based on packet and/or flow features
 - compare two ML models to perform multiclass application classification
 - Neural Network
 - Random Forest
 - evaluate the impact of training dataset (see next slide)
- (3 points) Advanced Task: Transfer Learning across different devices
 - Leverage knowledge gained from a subset of devices¹ (source domain) on classifying one set of **apps** to improve classification on one different device (target domain) having a limited amount of data that can be used for fine-tuning
 - Steps to follow:
 1. *Train a base model on a "source" task (app classification on device 1)*
 2. *Transfer the model to a "target" task (e.g., app classification on a limited dataset of device 2) **with fine-tuning***
 3. *Evaluate if transfer learning improves accuracy on the target task compared to training from scratch with the same limited target data*
 4. *Evaluate if transfer learning can approach the performance of an ideal model where lot of data is available for the target domain*

Resources for Transfer Learning

https://keras.io/guides/transfer_learning/

<https://neptune.ai/blog/transfer-learning-guide-examples-for-images-and-text-in-keras>

¹MAC address feature (BF_DEVICE) can be used to differentiate devices



Project #2 – Network Traffic Analysis

Application Classification + Transfer Learning

How to evaluate the impact of training dataset?

1. Select one or more applications from the dataset that performs two activities (Y and Z).
2. For each selected application, perform binary classification of type “App” or “Not App” maintaining the same train dataset for “Not App” label but varying the “App” label **in training set** as follows:
 - Case 1: Both activities (Y and Z)
 - Case 2: Only the activity Y
 - Case 3: Only the activity Z

Which is the activity that better generalize a specific application with respect to the other?

How does the training dataset affect generalization performance?

- For all experiments make sure that:
 - Training set is of the same size across various scenarios above
 - Test set includes samples for all the activities associated to the app under analysis

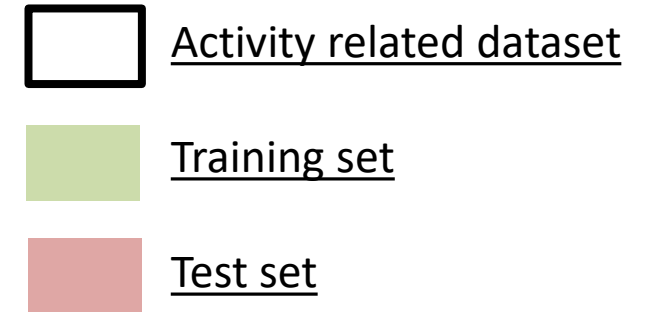
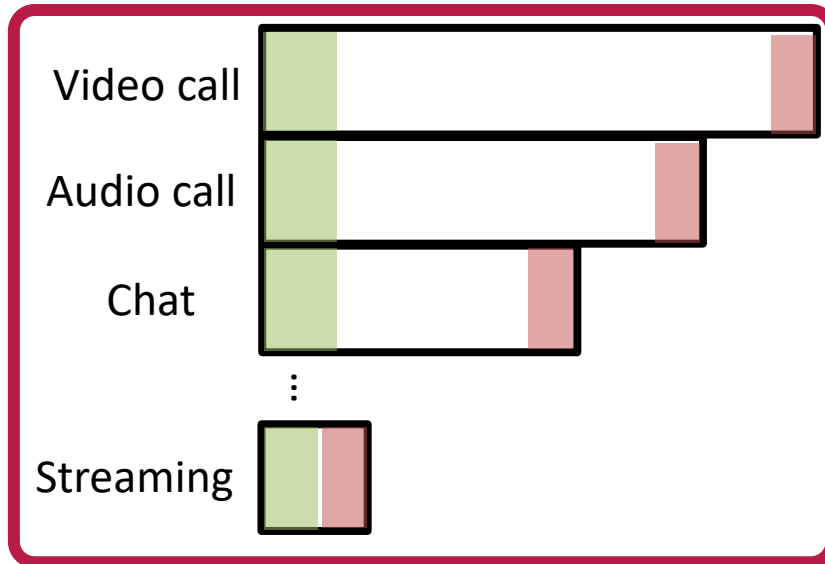


Project #2 – Network Traffic Analysis

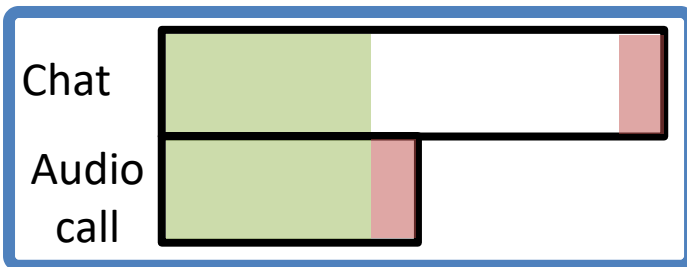
Application Classification + Transfer Learning

How to evaluate the impact of training dataset? - Visualized Example with “Telegram” application

Fixed “Not Telegram” Dataset

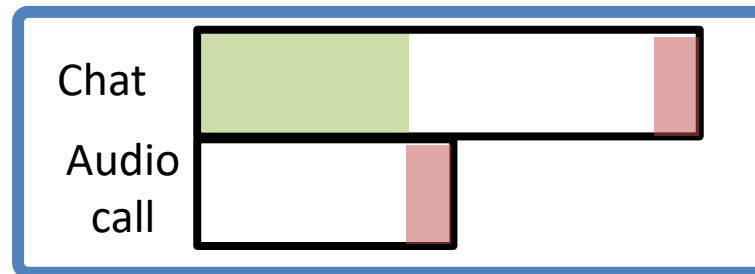


Case 1: Entire “Telegram” Dataset



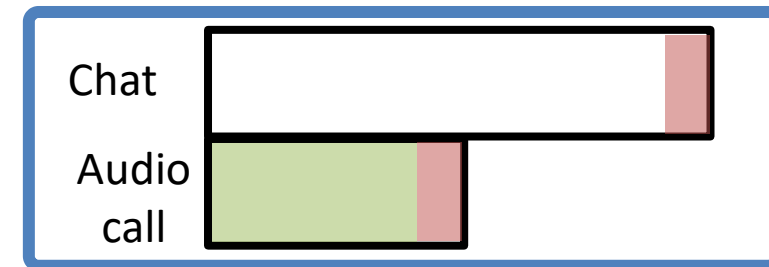
VS

Case 2: Only “Telegram” Chat Dataset



VS

Case 3: Only “Telegram” Audio Call Dataset



Project #3 – Network Traffic Analysis

Activity Classification + Federated Learning

- (12 points) Perform **Activity classification** based on packet and/or flow features
 - compare two ML models
 - Neural Network
 - Random Forest
 - evaluate the impact of training dataset (see next slide)
- (3 points) Advanced Task - Federated Learning for **Activity** Classification
 - Each smartphone (i.e., associated to a given MAC address¹) locally trains its own model with local data and periodically sends its weights to a centralized entity. A global classification model is obtained by aggregating the different local models
 - Analyses to perform
 - Show the performance trends over time (globally and/or locally)
 - Compare the performance with the same model but entirely trained on the locally generated
 - Reserve a MAC address in the dataset as holdout test

Resources for Federated Learning

<https://how.dev/answers/what-is-federated-averaging-fedavg>

¹MAC address feature (BF_DEVICE) can be used to differentiate devices



Project #3 – Network Traffic Analysis

Activity Classification + Federated Learning

How to evaluate the impact of training dataset?

1. Select one or more activities from the dataset that is performed by multiple apps (X, Y and Z)
2. For each selected activity, perform binary classification of type “Activity” or “Not Activity” maintaining the same train dataset for “Not Activity” label but varying the “Activity” label **in training set** as follows:
 - Case 1: All applications (X, Y and Z)
 - Case 2-3-4: Only the application {X, Y, Z}
 - Case 5-6-7: Only the applications {(X,Y), (X,Z), (Y,Z)}
- Is there an app/a subset of apps that allow better generalization of the specific activity with respect to the others?

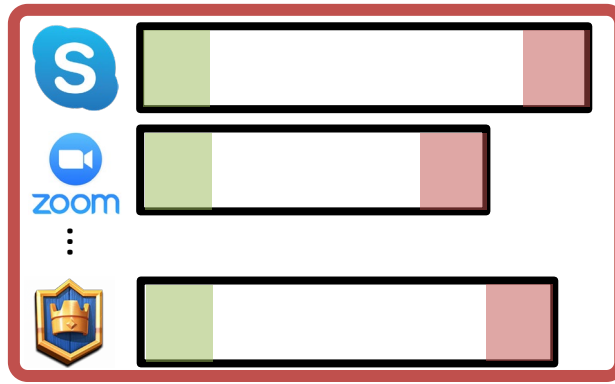


Project #3 – Network Traffic Analysis

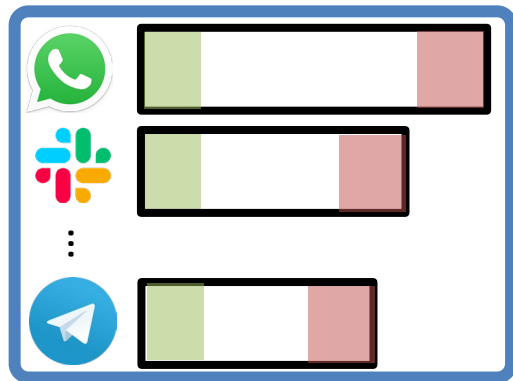
Activity Classification + Federated Learning

How to evaluate the impact of training dataset? - Visualized Example with “Chat” Activity

Fixed “Not Chat” Dataset

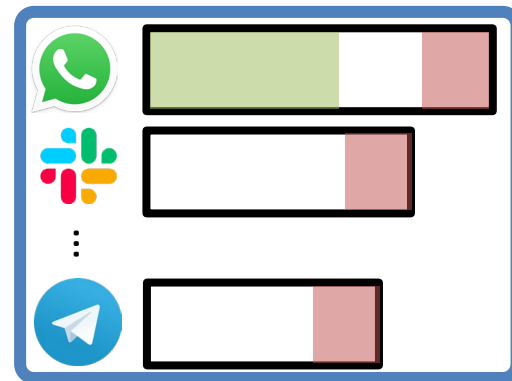


Case 1: Entire “Chat” Dataset



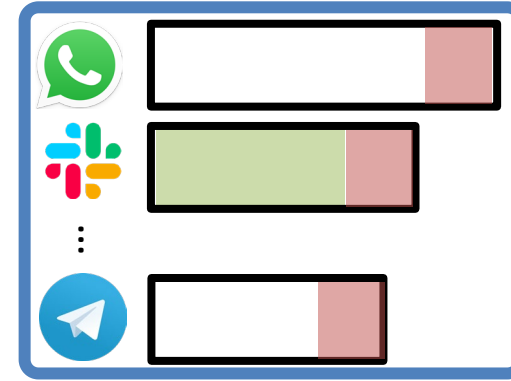
VS

Case 2: Only “Whatsapp” Dataset



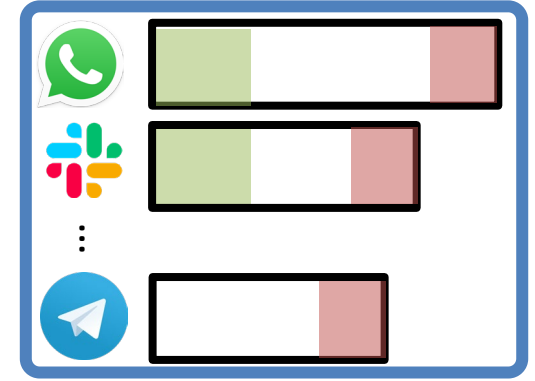
VS

Case 3: Only “Slack” Dataset



VS

Case 5: “Whatsapp” and “Slack” Dataset



Project #4-5 – Packet Loss Event Classification

Background

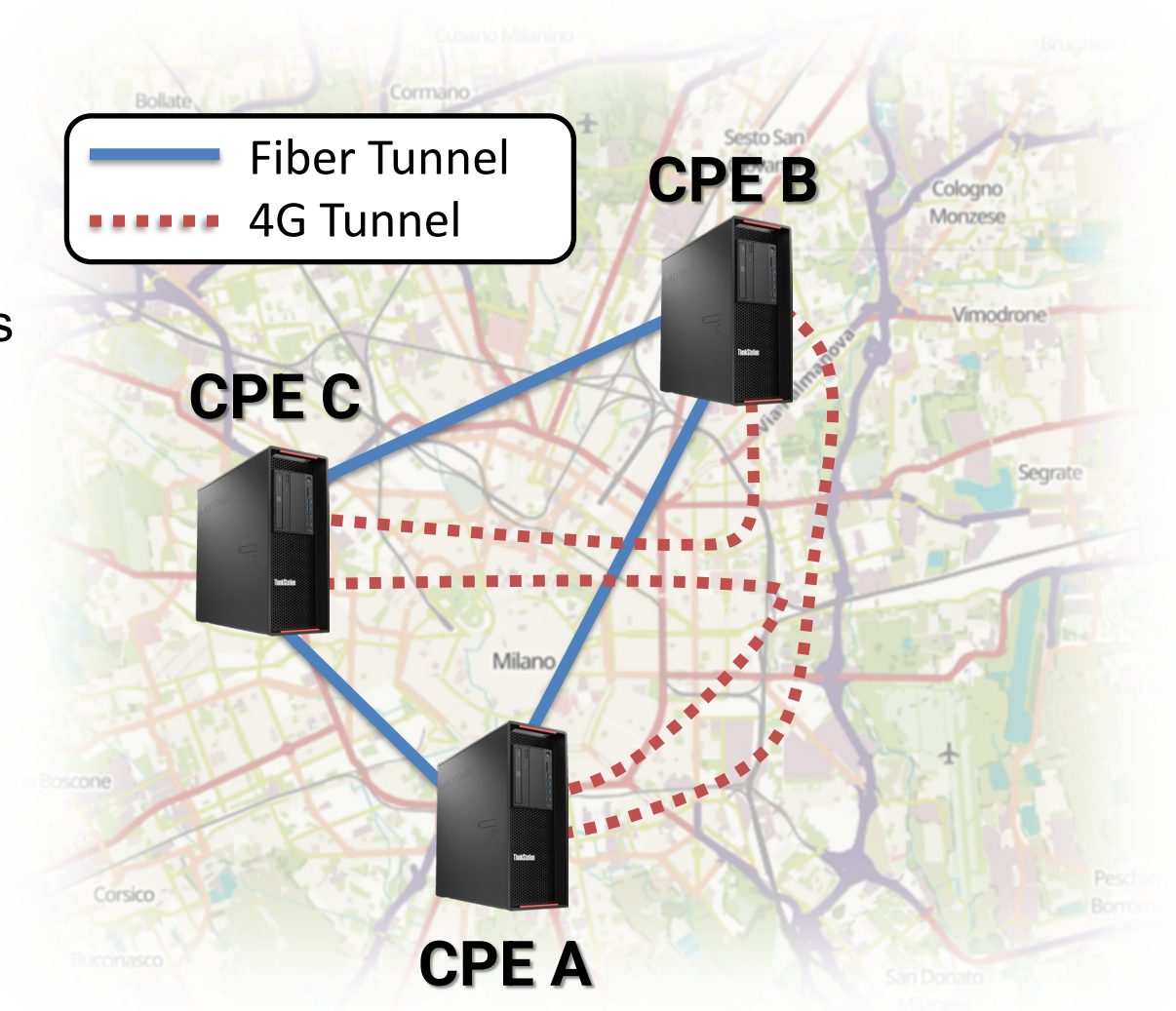
- SD-WAN (Software-Defined Wide Area Network) is an emerging technology for the interconnection of geographically distributed company branches
 - Uses **one or more public best-effort internet access** to interconnect the different sites through one CPE (Customer Premises Equipment) in each site. The interconnection is secure by conveying data through encrypted tunnels.
 - Each of these best-effort internet access corresponds to **different paths in the public Internet**
- **Problem:** Mission-critical inter-site traffic requires strict QoS (e.g., extremely low packet loss). Network inefficiencies (latency spikes, packet loss) on the paths can cause performance degradation and potential financial impact (e.g., in financial contexts).
- We would like to predict network inefficiencies (e.g. latency spikes or packet loss bursts) before they happen so as to redirect traffic on the best path
- Can we use ML to predict packet loss before it happens based on previous historical data?



Project #4-5 – Packet Loss Event Classification

Testbed setup

- Three geographically distributed SD-WAN CPEs across Milan
- Dual WAN connectivity per CPE:
 - Fiber (FTTC/FTTH) tunnel
 - 4G cellular network tunnel
- A latency measurement campaign has been performed on this setup



Project #4-5 – Packet Loss Event Classification Dataset

- For any ordered pair (CPE X, CPE Y), two measurement files (.csv) have been generated (one for the fiber tunnel and one for the 4G tunnel)
- Two measurement windows, respectively of duration ~15h and ~ 30h, with one second granularity

	time		delay_ms
23691	2025-03-05	19:05:22.567	13.2708
23692	2025-03-05	19:05:23.567	21.1522
23693	2025-03-05	19:05:24.566	14.8669
23694	2025-03-05	19:05:25.569	-1.0000
23695	2025-03-05	19:05:26.569	14.7499
23696	2025-03-05	19:05:27.570	13.8136
23697	2025-03-05	19:05:28.570	13.6749

Single file structure

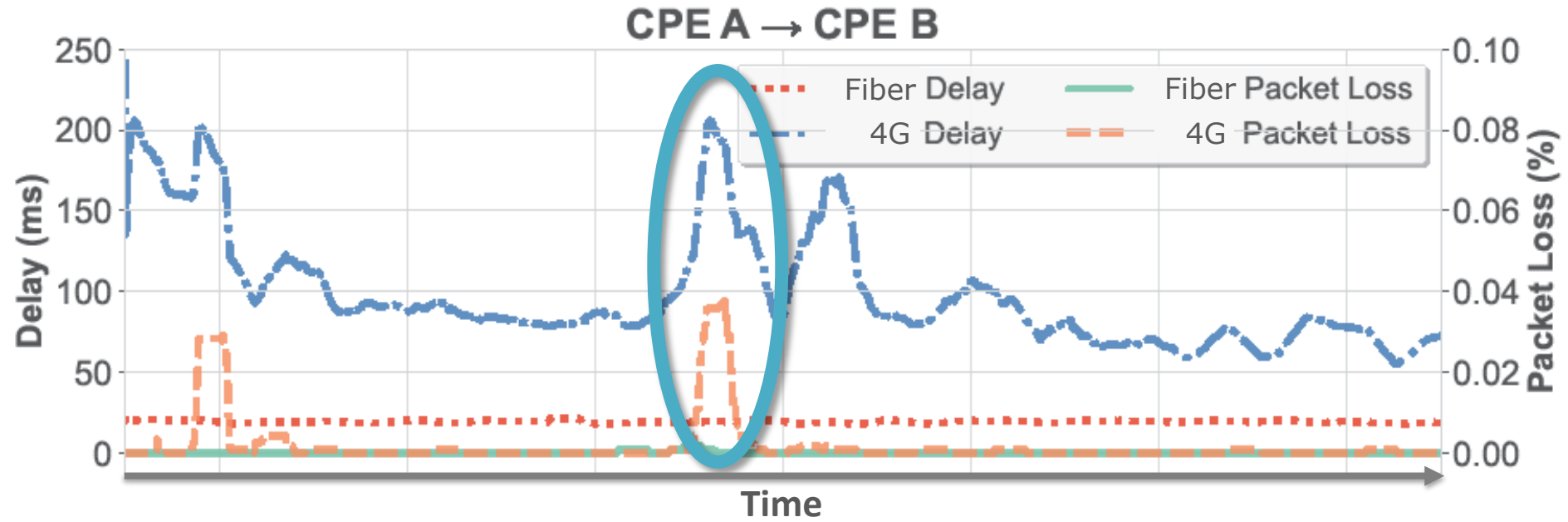
(e.g. fiber tunnel from CPE A to CPE B)

Packet losses are identified by the value -1 in the **delay_ms** column



Project #4-5 – Packet Loss Event Classification

The Dataset Visualized



Evident **correlation** between packet loss and delay increase

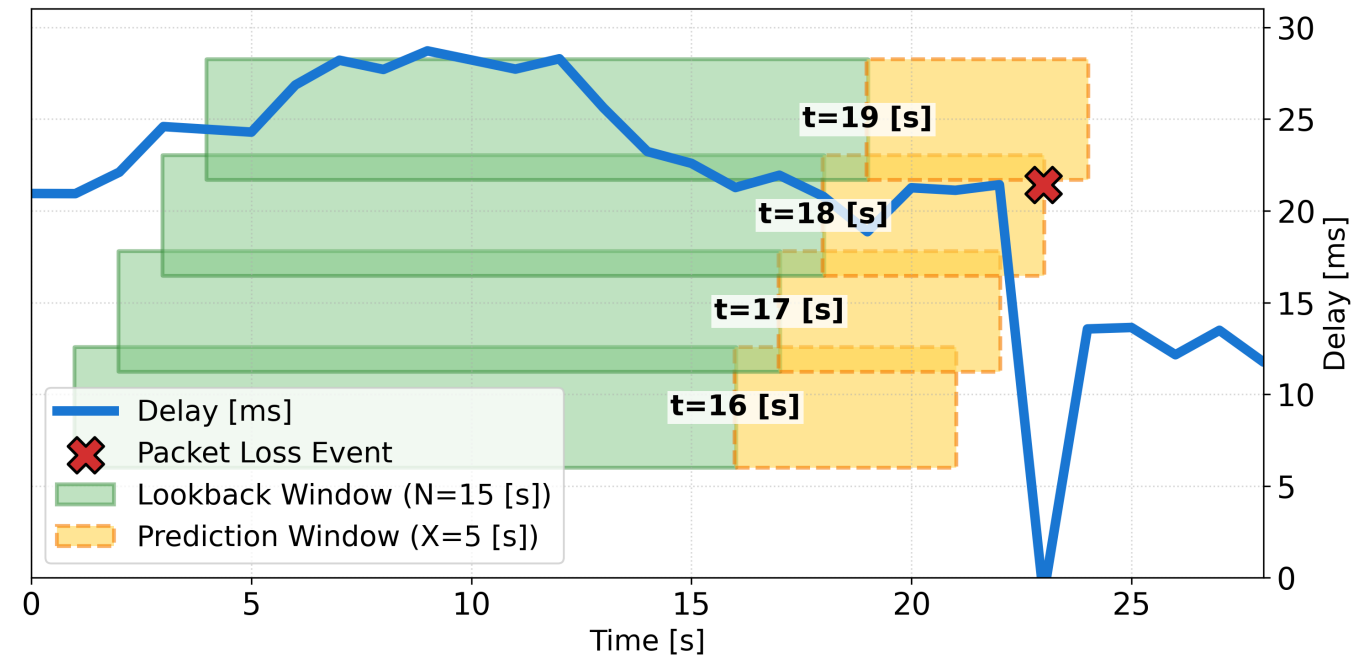
Can we predict packet loss event before it happens looking at historical delay values?



Project #4-5 – Packet Loss Event Classification

Hints

- **Data preprocessing**
 - The rows with `delay_ms` set to `-1` (packet loss event) must be pre-processed
- **Procedure**
 - Use a sliding window approach: take the last **N seconds** of measurements (*Lookback Window*) as input to predict the presence of a packet loss event in the following **X seconds** (*Pred. Window*)
 - **Experiment with different N and X values**
 - For each Lookback Window statistical features can be used



Project #4 – Packet Loss Event Classification

Packet Loss Event Classification with RF & Transfer Learning

- (12 points) Perform classification on packet loss event using two different models
 - Neural Network
 - Random Forest
 - Evaluate feature importance
 - Experiment with different N and X values (see previous slide)
- (3 points) Advanced Task: Transfer Learning
 - Build a model trained on a single direction (e.g., CPE A \rightarrow CPE B) and fine-tune the model with data on the opposite direction (e.g., CPE A \leftarrow CPE B)
 - Assess the performance of the fine-tuned model
 - Compare the accuracy between the fine-tuned model and individually computed models entirely trained on local dataset
 - Keep the last hours of each measurement windows as hold-out test set



Project #5 – Packet Loss Event Classification

Packet Loss Event Classification with XGBoost & Federated Learning

- (12 points) Perform classification on packet loss event using two different models
 - Neural Network
 - XGBoost
 - Evaluate feature importance
 - Experiment with different N and X values (see previous slide)
- (3 points) Advanced Task: Federated Learning
 - Assume a centralized entity (e.g., SD-WAN Controller) periodically collects local models (e.g., trained with data available at individual CPEs), aggregates them and redistribute the information to build a generalized model
 - Compare the performance between the general (i.e., federated) and the CPE-specific (local) models




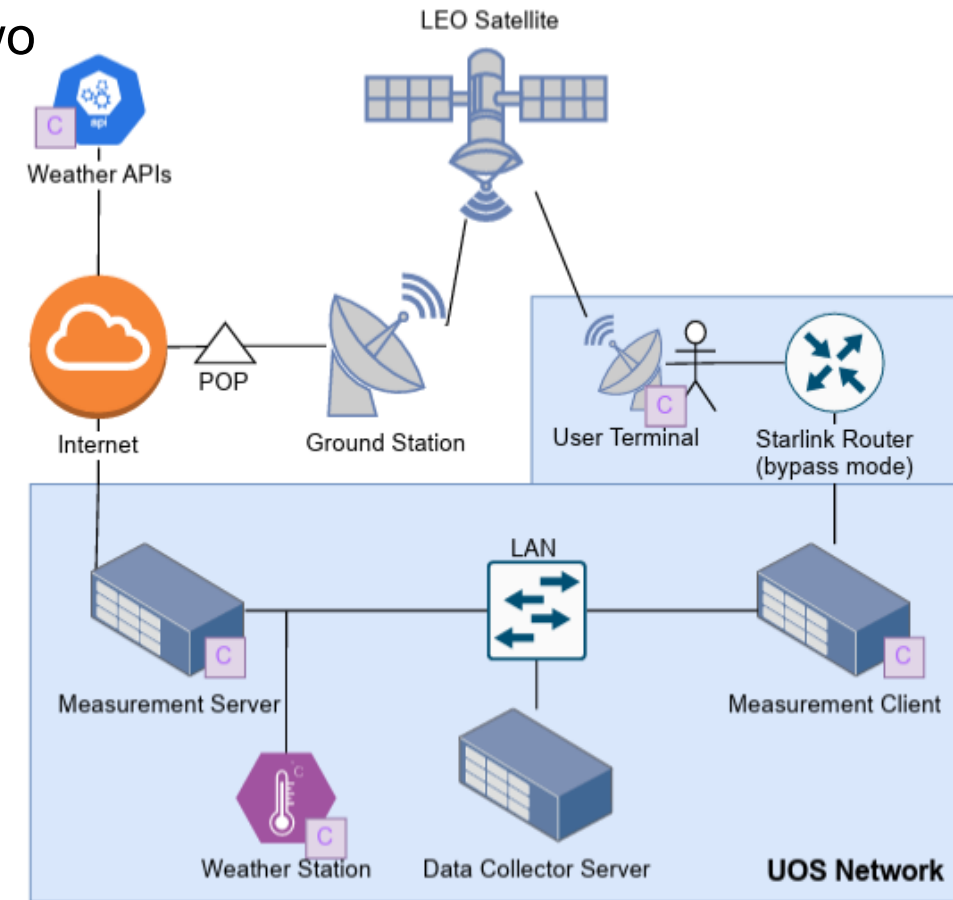
Project #6 – Weather-based performance prediction of satellite-internet provider: Background

- Low-Earth Orbit (LEO) satellite-based internet service providers (e.g., Starlink) offer high-speed internet globally, but performance can vary a lot over time
 - Atmospheric conditions (rain, fog, water absorption and droplet scattering), affect the available bandwidth
- Can we leverage weather forecasts to predict bandwidth degradation before it occurs, allowing users or systems to anticipate performance?



Project #6 – Weather-based performance prediction of satellite-internet provider: Dataset [1][2]

- 140k measurements taken over a period of 6 months from two different European cities:
 - Osnabrück (DE) → 6.5 months, 80k measurements
 - Enschede (NL) → 5.5 months, 60k measurements
- The raw dataset consists of various .csv files, containing measurements collected from a data point  (Fig.1).
- The following shall be considered:
 - `starlink.csv`: Data collected from the Starlink dishes (User Terminal in Fig. 1) with information about **uplink and downlink throughput, latency** and other statistics
 - `froggit.csv`: Data from weather stations located near the dishes (rain, humidity, temperature ...)



[1] Laniewski, Dominic, et al. "Wetlinks: a large-scale longitudinal starlink dataset with contiguous weather data." *2024 8th Network Traffic Measurement and Analysis Conference (TMA)*. IEEE, 2024.

[2] <https://github.com/sys-uos/WetLinks>



Project #6 – Weather-based performance prediction of satellite-internet provider: Dataset – “Visual” Considerations

- A correlation between hour of the day, weather condition and the download (and upload) throughput can be seen
- We want to use this correlation to forecast the available bandwidth in advance

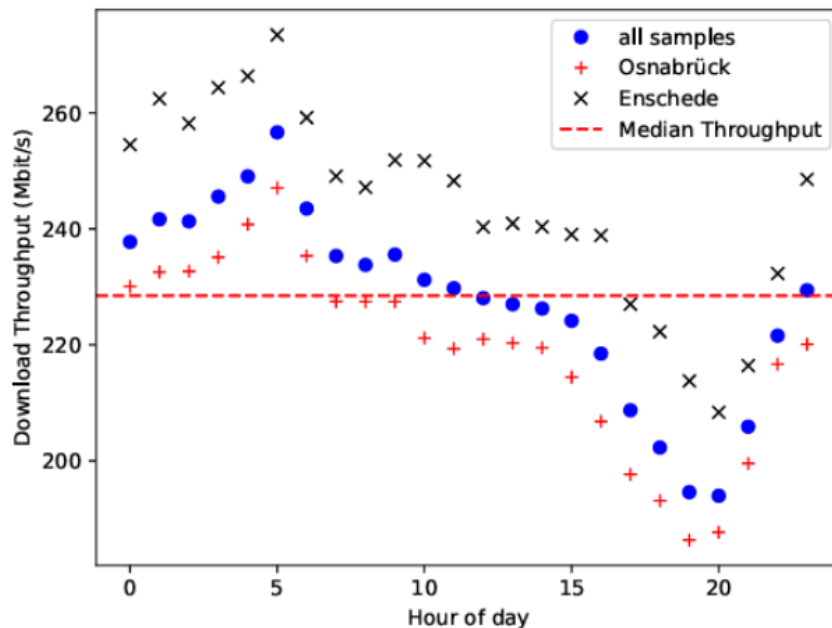


Fig. 10: Median download throughput rate per hour

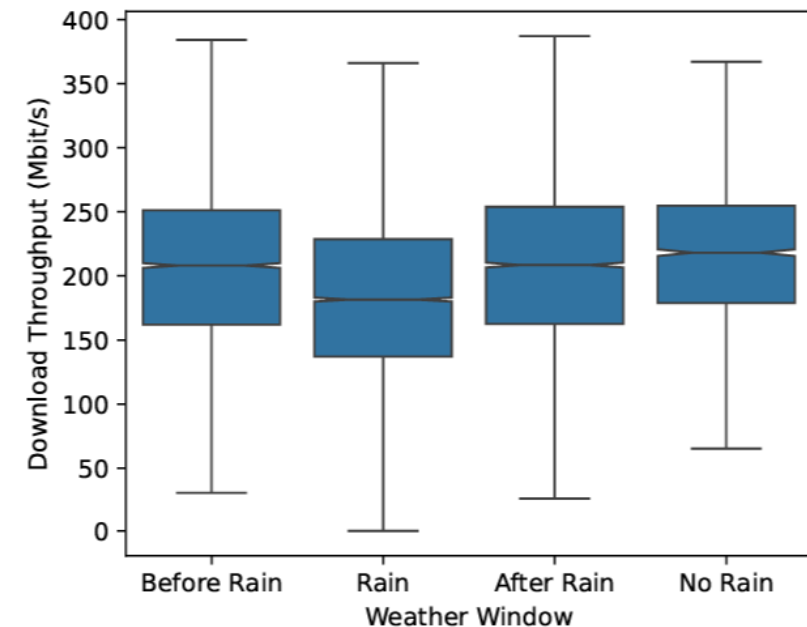


Fig. 12: Boxplots for different weather conditions. For each rain window, 4 samples before and after the rain are used. We assume cloudy conditions before and after rain. In the no rain case, only samples with high solar radiation are used, to ensure that there are no clouds.



Project #6 – Weather-based performance prediction of satellite-internet provider: Assignment

- (12 points) Basic assignment
 - Build a model to predict the bandwidth (download and upload) given the time of the day and the weather conditions
 - Assume we have an ideal weather forecaster (100% accuracy and perfectly synchronized with satellite network data). How does this impact the prediction of future available bandwidth?
 - Compare two different ML models: LSTM and Neural Network
 - Evaluate feature importance
 - Experiment on the amount of historical samples (weather, previous bandwidth and hour of the day)
- (3 points) Advanced Task: Transfer Learning
 - Train a model from a site (e.g. Osnabrück (DE)) and try to apply it with fine-tuning to another one (e.g. Enschede(NL)). Does this bring any improvements?
 - Use the data from last seven days of each month of the target dataset as hold-out test



Project #7-8-9 – EDFA profile regression

Background

- Optical amplifiers (Erbium-Doped Fiber Amplifier, EDFA) compensate power attenuation in optical fibers and guarantee sufficient power at the receiver
- Gain is not the same across the wavelengths
 - Input power profile: $P_{in}(\Lambda) = \{P_{in}(\lambda_1), P_{in}(\lambda_2), \dots, P_{in}(\lambda_N)\}$
 - Output power profile: $P_{out}(\Lambda) = \{P_{out}(\lambda_1), P_{out}(\lambda_2), \dots, P_{out}(\lambda_N)\}$
- Complex transfer function: $P_{out}(\Lambda) = f(P_{in}(\Lambda))$
- 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?**

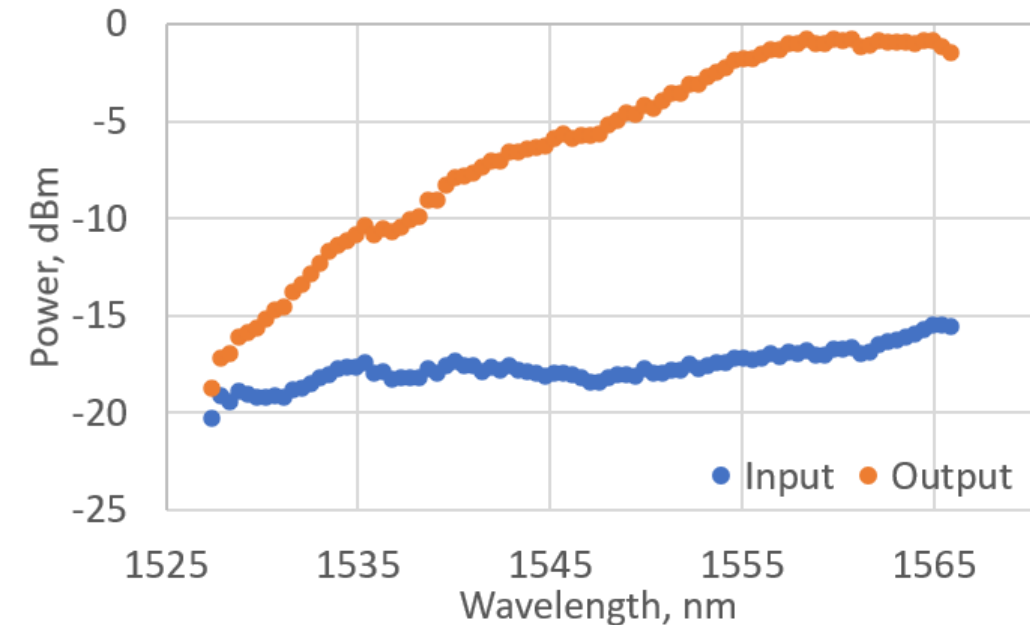
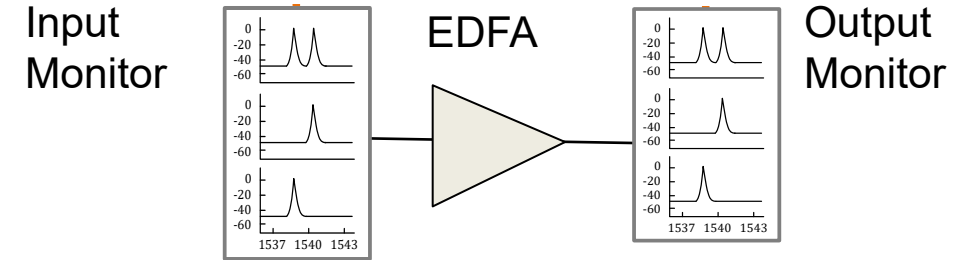


Project #7-8-9 – EDFA profile regression

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 $N=84$ channels
 - 16497 entries with different power profiles (subsets of active 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



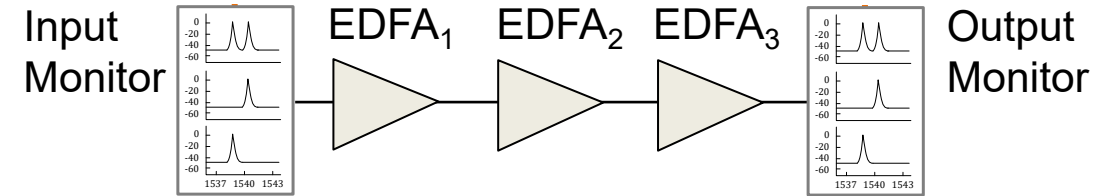
[1] https://data.dtu.dk/articles/dataset/Input-output_power_spectral_densities_for_three_C-band_EDFAs_and_four_multispan_inline_EDFAd_fiber_optic_systems_of_different_lengths/13135754/1

[2] <https://ieeexplore.ieee.org/document/9333297>

Project #7-8-9 – 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 $N=84$ channels
 - 2500 entries with different power profiles



- 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>

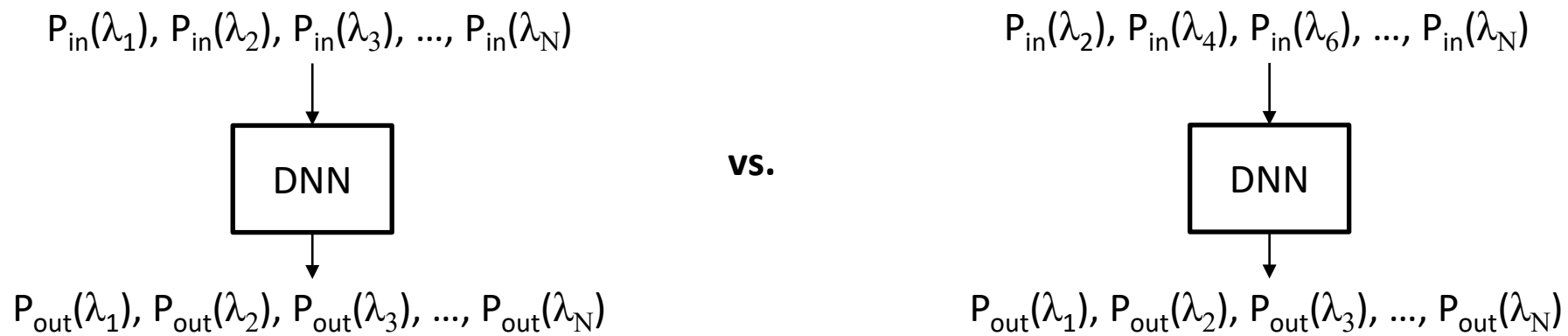


Project #7 – EDFA profile MIMO regression Assignment

(3 points) Advanced Task:
Free Choice (TL, FL, XAI)

(12 points) Basic assignment:

- Given input power profile, predict **FULL output power profile**: regression with multiple inputs and multiple outputs
 - Use Deep Neural Network regression
 - Use only 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 performance of output power profile prediction and model complexity**

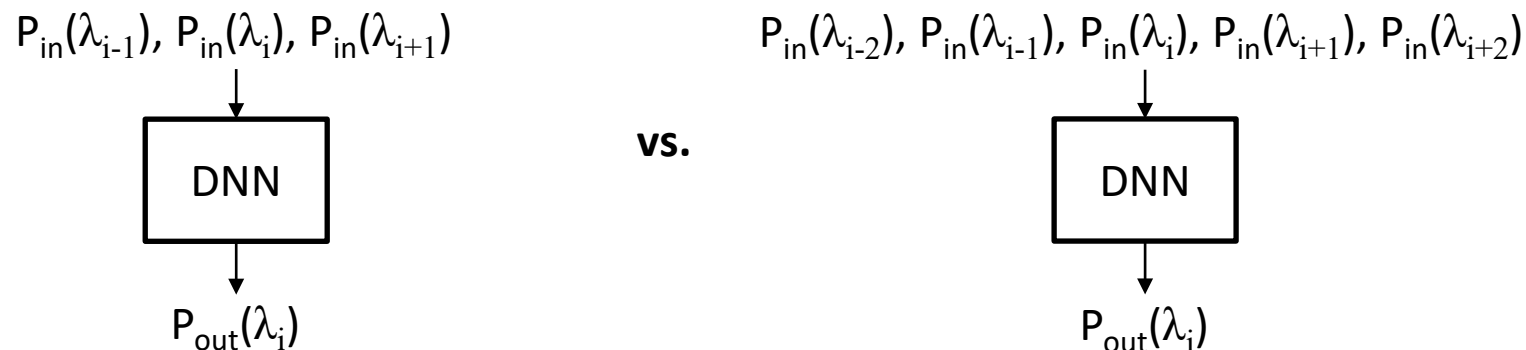


Project #8 – EDFA profile MISO regression Assignment

(3 points) Advanced Task:
Free Choice (TL, FL, XAI)

(12 points) Basic assignment:

- Given input power profile, predict **output power for one channel**: **regression with multiple inputs and single output**
 - Use Deep Neural Network regression
 - Use only 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 near the CUT
 - Train the regressor with different input dimensions to predict **power of CUT**
- **Compare the accuracy of output power prediction and model complexity**

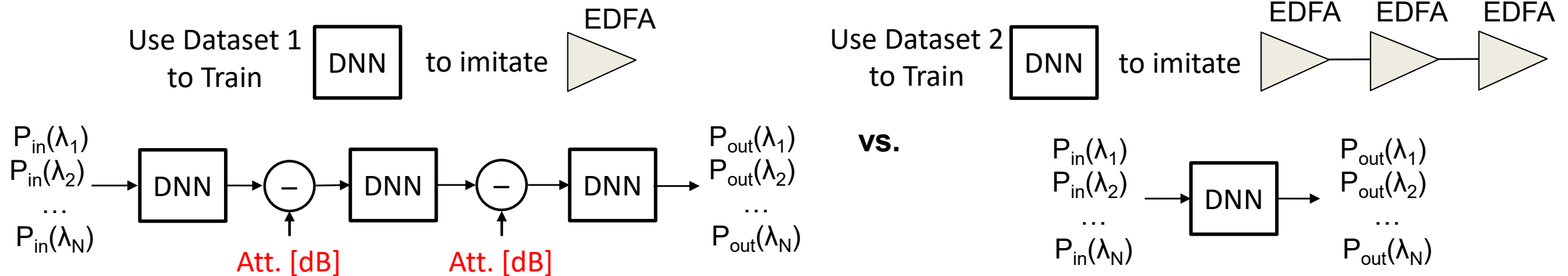


Project #9 – EDFA profile multi-span prediction Assignment

(3 points) Advanced Task:
Free Choice (TL, FL, XAI)

(12 points) Basic assignment:

- Given input power profile, predict **FULL 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

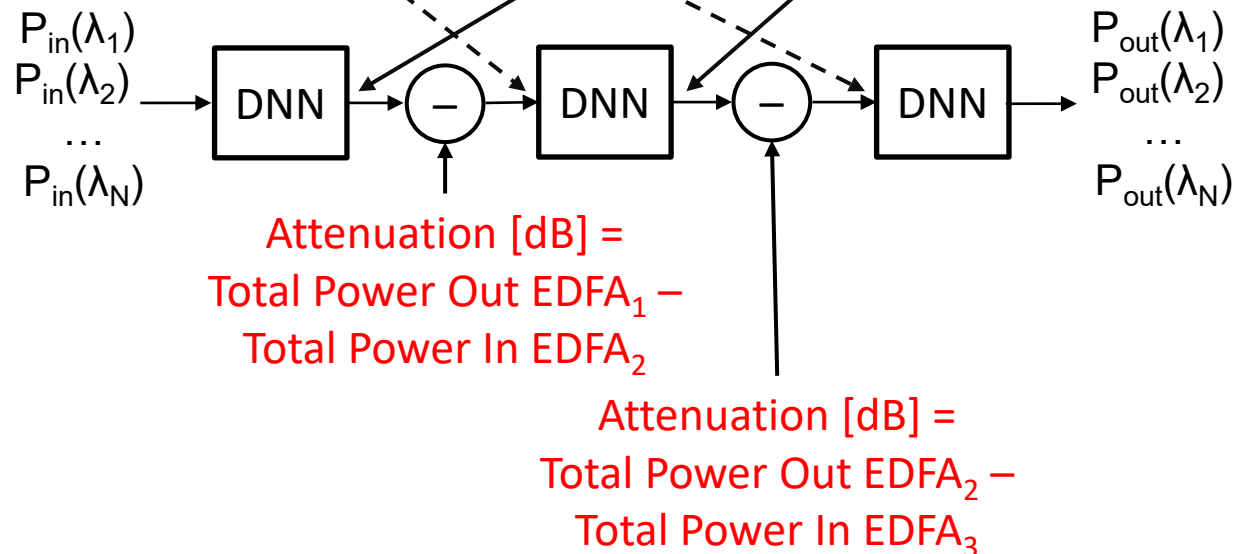


Project #9 – EDFA profile multi-span prediction

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 #10 – 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** 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?**

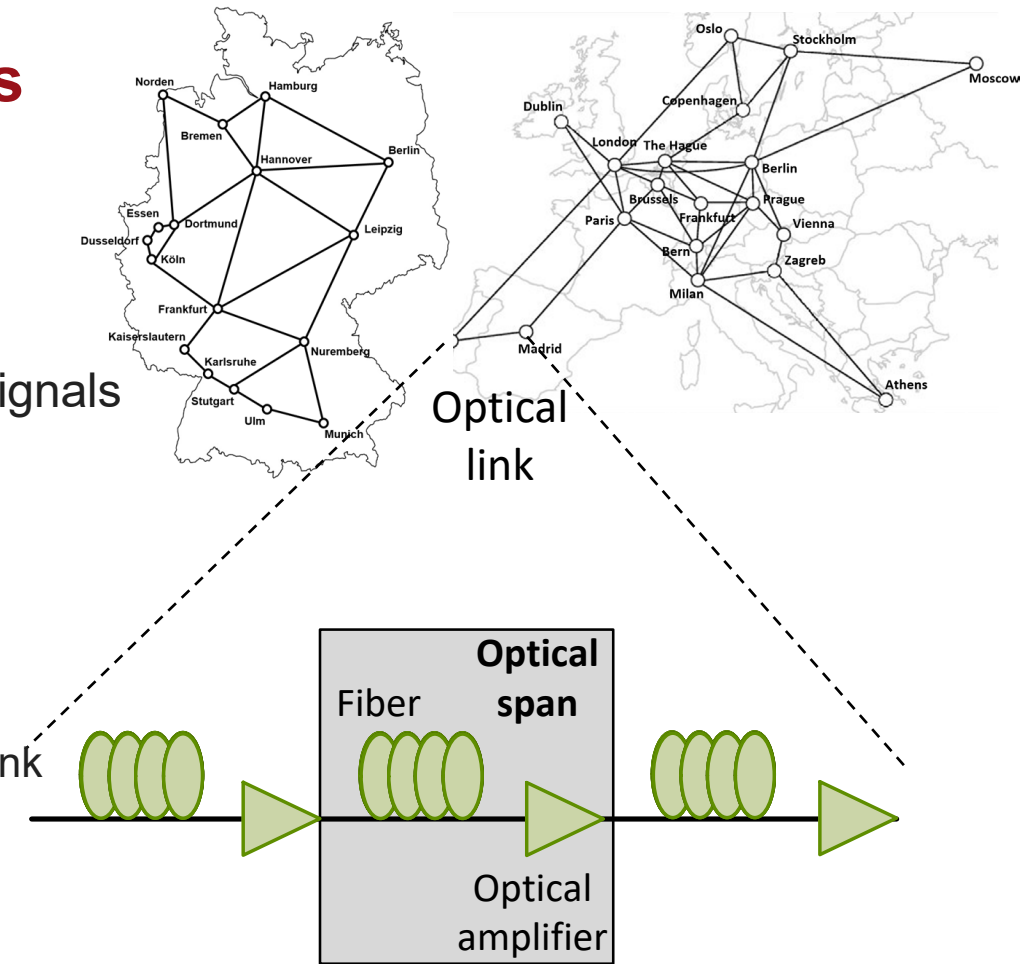
SNR prediction	Accuracy with imprecisely known path parameters	Margins (underutilization of network resources)	Must be trained on SNR measurements from the network
Analytical model	Low	High	No
ML estimator	High	Low	Yes



Project #10 – 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
 - 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



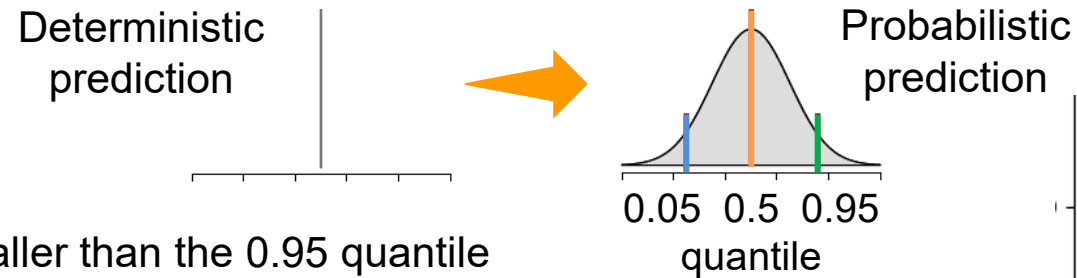
Path features			Interference features			Label
Length (in km) of fiber in span 1	...	Length (in km) of fiber in span N	Number of channels in link 1	...	Number of channels in link M	SNR, dB



Project #10 – 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 predictionOR
 - penalize overestimations – low quantiles – more conservative prediction

- Implemented in scikit-learn for GradientBoostingRegressor:

```
loss, default='squared_error'
```

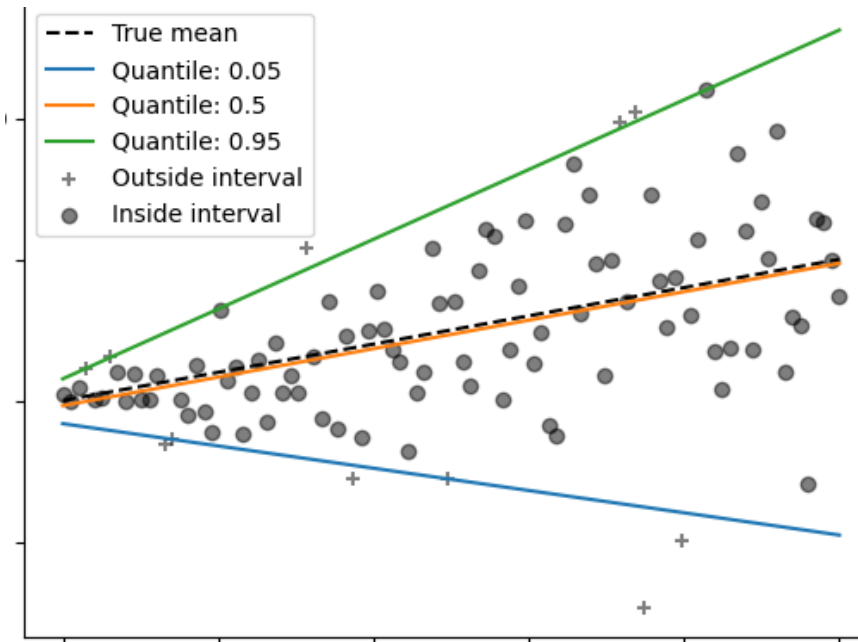
```
'quantile' allows quantile regression (alpha specifies the quantile)
```

```
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 #10 – QoT prediction in optical networks

Assignment

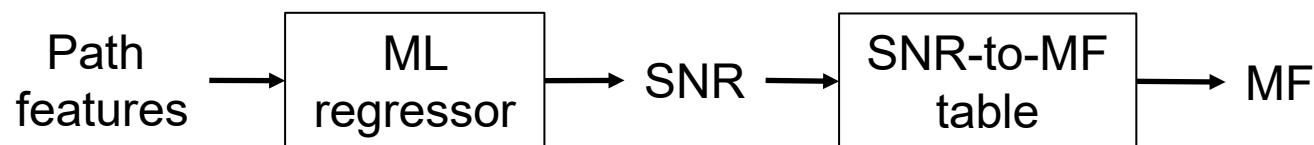
(3 points) Advanced Task:

Free Choice (TL, FL, XAI)

(12 points) Basic assignment:

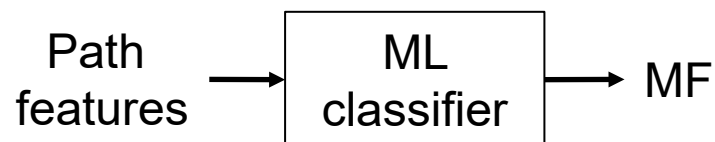
- Given path features as input, choose MF: **regression or multi-class classification**
- Use a tree model (e.g., GBT, LightGBM) as both regressor and classifier

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 #11 – 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 #11 – Optical failure localization

Background

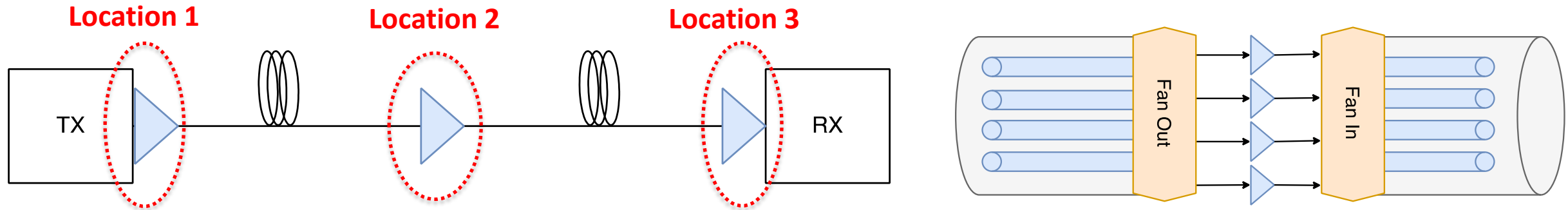
1. **(Early) Detection** (Whether or not?)
 - Predict/assess if OSNR/BER is/will be intolerable
 - Allows early/quick activation of proactive procedures
2. **Identification** (Which cause?)
 - e.g., filter misalignment, laser drift, fiber bending, amplifier malfunctioning ..
 - Reduced Mean Time To Repair (MTTR)
3. **Localization** of soft-failures (Where?)
 - e.g., which node/link along the path?
4. **Magnitude estimation** (How much?)
 - Triggers the proper reaction(e.g., device restart/reconfiguration, lightpath re-routuing, in-field reparation...)

Project's focus



Project #11 – Failure Management in Optical Networks

Testbed setup



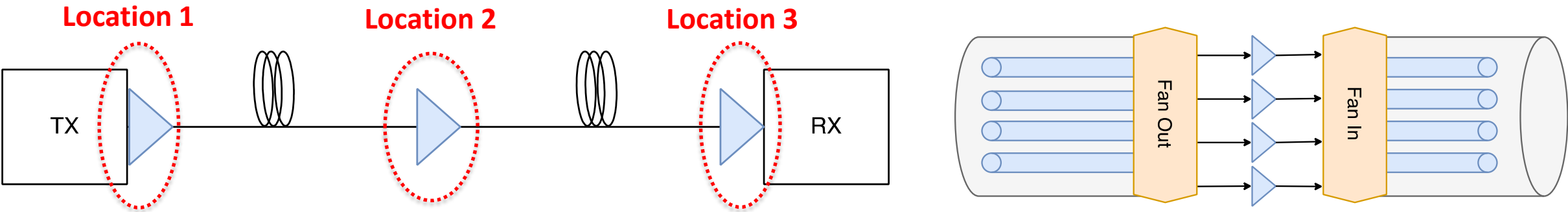
- Failure injection: introducing additional cross-talk in a multi-core optical line system
- Failure location: @TX (Location 1); @fan-in/fan-out (Location 2); @RX (Location 3)
- Three modulation formats: QPSK, 16-QAM, 64-QAM
- QoT metrics (features): BER, OSNR, EVM, GMI, THR
 - [BER \(Bit Error Rate\)](#); [OSNR \(Optical Signal to Noise Ratio\)](#); [EVM \(Error Vector Magnitude\)](#); [GMI \(Generalized Mutual Information\)](#); [THR \(Throughput\)](#)
- Dataset 1: class 0 (no failure); class 1 (failure @TX); class 2 (failure @fan-in/fan-out); class 3 (failure @RX)
- Dataset 2: class 1 (failure @TX); class 2 (failure @fan-in/fan-out); class 3 (failure @RX)

Can we use ML to localize cross-talk in the optical line based on statistical variations of QoT metrics?



Project #11 – Failure Management in Optical Networks

Dataset



- Dataset 1: class 0 (no failure); class 1 (failure @TX); class 2 (failure @fan-in/fan-out); class 3 (failure @RX)
- Dataset 2: class 1 (failure @TX); class 2 (failure @fan-in/fan-out); class 3 (failure @RX)

start date	stop date	acquisition time	scenario	XT	modulation	BER	Q	EVM	GMI	thr	OSNR
2025-01-15 10:51:20	2025-01-15 10:51:23	2.0648	0	NaN	64	0.031331	5.3977	0.11357	10.601	212.02	25.501
2025-01-15 10:51:44	2025-01-15 10:51:46	1.9795	0	NaN	64	0.031530	5.3845	0.11371	10.599	211.98	25.485
2025-01-15 10:55:44	2025-01-15 10:55:46	2.0468	0	NaN	64	0.031776	5.3683	0.11385	10.593	211.86	25.482
2025-01-15 11:36:48	2025-01-15 11:36:50	1.9725	0	NaN	64	0.033470	5.2589	0.11459	10.515	210.29	25.545
2025-01-15 11:36:51	2025-01-15 11:36:53	1.9905	0	NaN	64	0.033505	5.2567	0.11430	10.510	210.19	25.547

Timestamp

Label

Mod. Format

QoT metrics

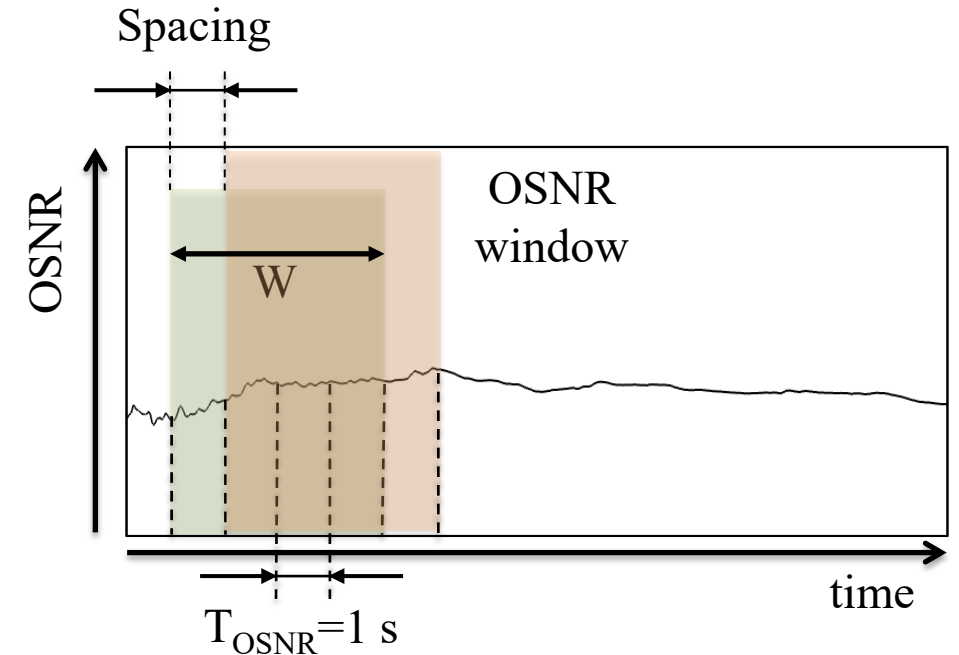
Project #11 – Optical failure localization

Hint on dataset preprocessing

1. Remove mean component from the QoT metric (BER, OSNR, EVM, GMI, THR), using normalization, standardization (one of your choice)*
2. Form “windows” of duration W , including W consecutive QoT 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?

*Data normalization (remove mean component)

- Per failure location and per modulation format
 - Gather data for each failure location and for each modulation format, and then concatenate the data while keeping the temporal order



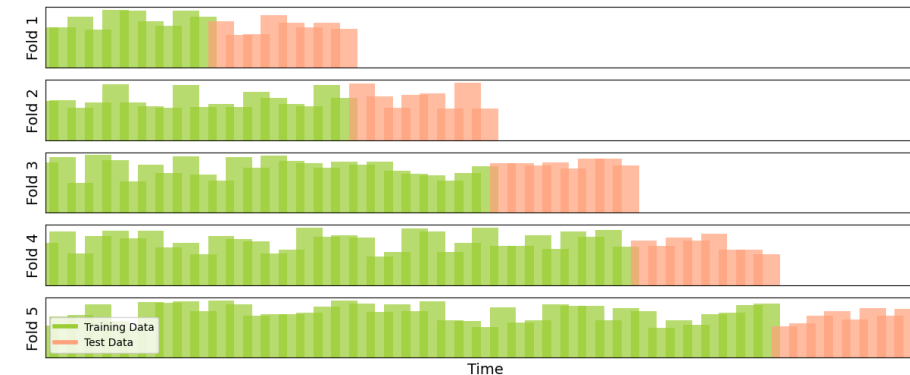
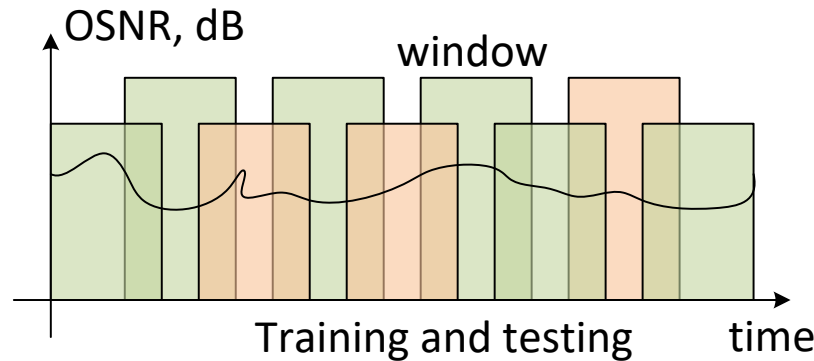
Project #11 – Optical failure localization: single-step multi-class classification Assignment

(3 points) Advanced Task:

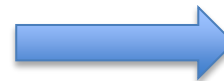
Free Choice (TL, FL, XAI)

(12 points) Basic assignment:

- Given QoT metrics measurements as input, localize the fault: **multi-class classification**
- Use a tree classifier (e.g., Random Forest, XGBoost, LightGBM)
- We can split data into training and testing datasets randomly (random split) or in time (temporal splits)



- Perform parametric analysis on the following
 - Feature selection
 - Window size
 - Train/Test split size (Random vs In time)



Dataset/ Splitting	Random	In time
Dataset 1	Scenario 1	Scenario 2
Dataset 2	Scenario 3	Scenario 4

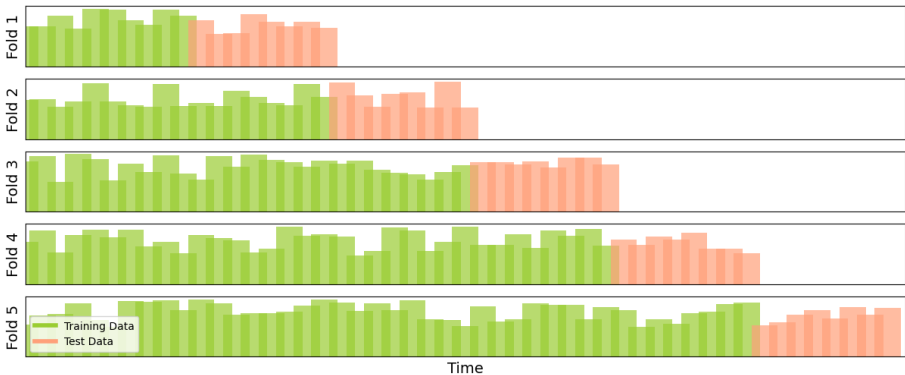
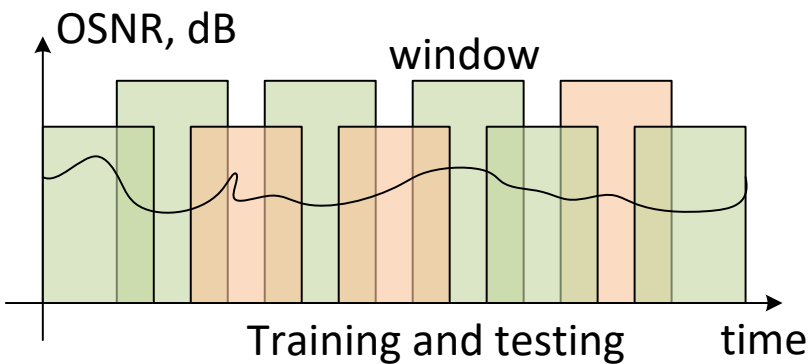


Project #12 – Optical failure localization: two-step multi-class classification Assignment

(3 points) Advanced Task:
Free Choice (TL, FL, XAI)

(12 points) Basic assignment:

- Given QoT metrics measurements as input, localize the fault: **multi-class classification**
- Use a tree classifier (e.g., Random Forest, XGBoost, LightGBM)
- We can split data into training and testing datasets randomly (random split) or in time (temporal splits)



- One- vs Two-step classification (using only Dataset 1)
 - One-step classification
 - Multi-class failure localization (four classes)
 - Two-step classification=Classifier 1 + Classifier 2
 - Classifier 1: binary classification (2 classes: fail/no-fail)
 - Classifier 2: multi-class classification (3 classes: failure localization)

Classifier/ Splitting	Random	In time
One-step	Scenario 1	Scenario 2
Two-step	Scenario 3	Scenario 4

