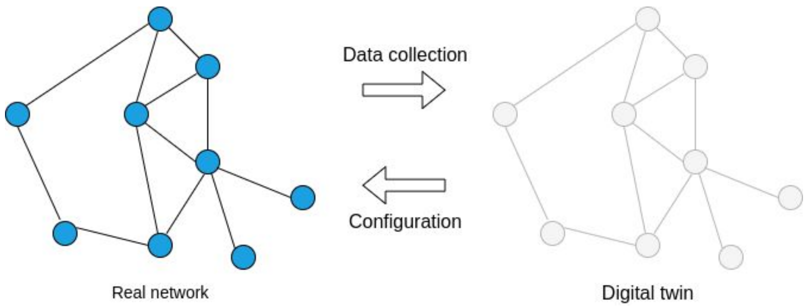
Graph Neural Networks to evaluate KPIs

Miquel Farreras

Institute of Informatics and Applications, Universitat de Girona, Girona, Spain
Fifth Delft - Girona Workshop on Robustness of Complex Networks
7th July 2022

Introduction

- **Digital Twin** → virtual representation of a real object. Explore configurations to analyze impact. Typical use cases:
  - **Engineering of physical objects** → test any design reducing development costs.
  - **Operations management** → networks, logistics, maintenance, business process optimization…
- Use case on communication networks, **5G and B5G**:
  - Complex, strict Service Level Agreement (SLA) services e.g., AR, V2X, IoT.
  - **Key Performance Indicators (KPIs)** → delay, jitter, loss, throughput in real time.
  - Network Slicing management (admission control, orchestration) with low response times.



Tools to build a Digital Twin Network

Solutions for predicting network properties:

	Advantages	**Disadvantages**
Analytical modeling	Fast predictions	Non-realistic and static network properties, poor predictions
Packet simulators	High precision	High computational complexity and execution time
Traditional AI/ML models	Easy model update/retrain, fast KPIs prediction	Not designed for graph data, poor predictions
Graph Neural Networks (GNNs)	ML models optimized for graphs. Fast and reliable KPIs prediction	Complex to develop, generalization problems with different size networks

How do GNNs work? (I)

Intuition: Nodes **aggregate** information from their **neighbors** using neural networks.

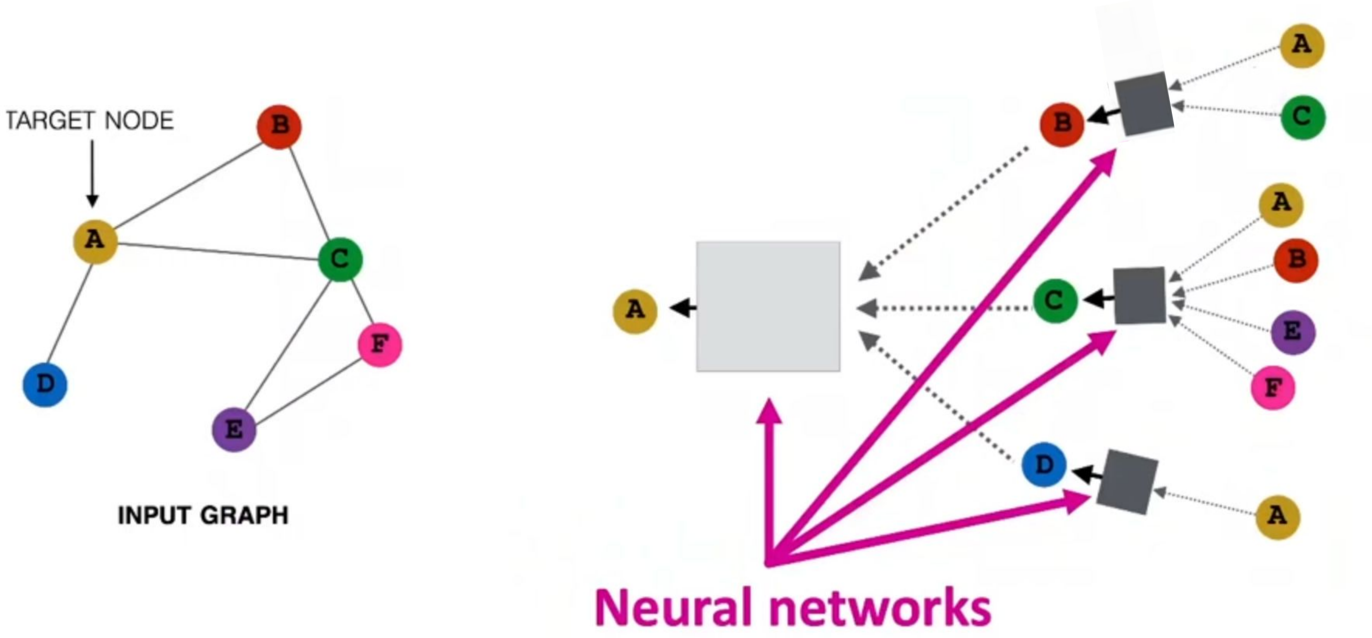
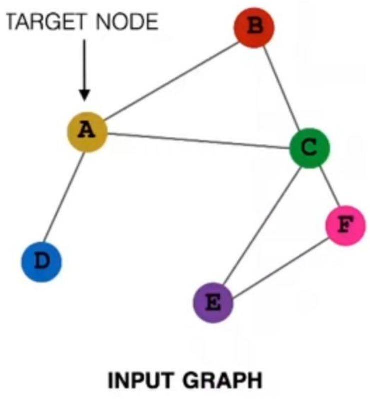


Image sources: CS224W course. Machine Learning with Graphs. Jure Leskovec, University of Stanford.

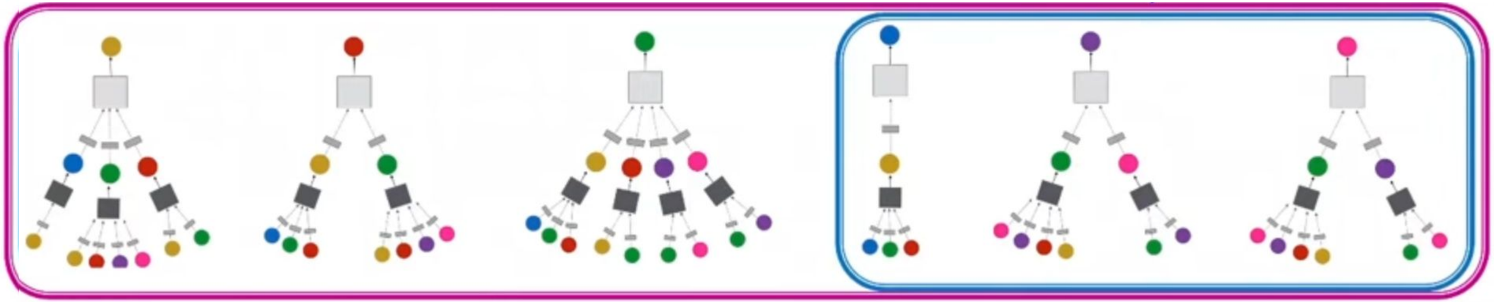
How do GNNs work? (II)

Intuition: Nodes **aggregate** information from their **neighbors** using neural networks.



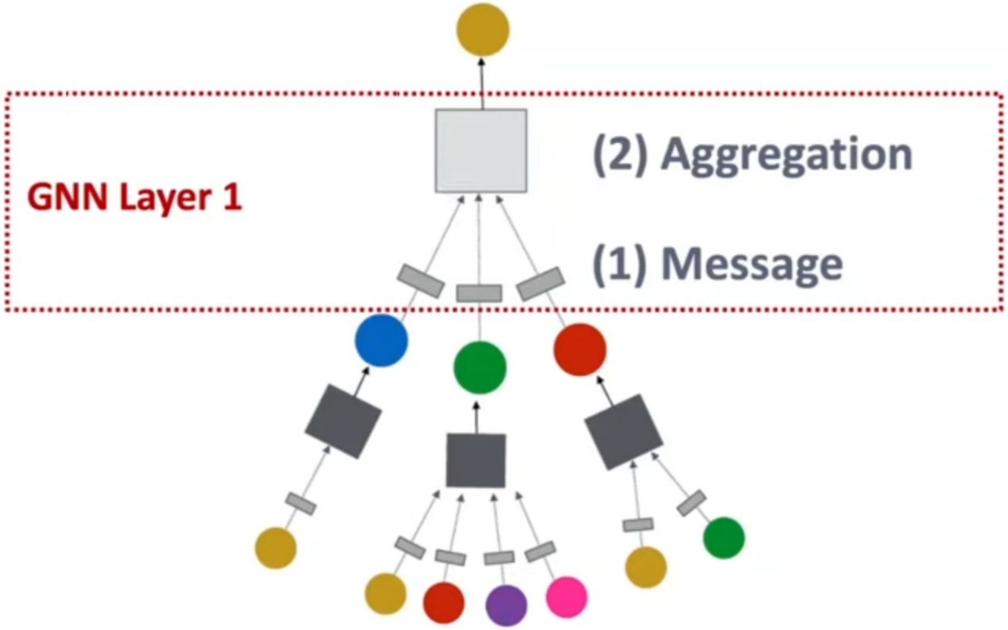
Generate embeddings for the nodes of the input graph

Also for nodes and graphs it was never trained on, using the same neural networks



How do GNNs work? (III)

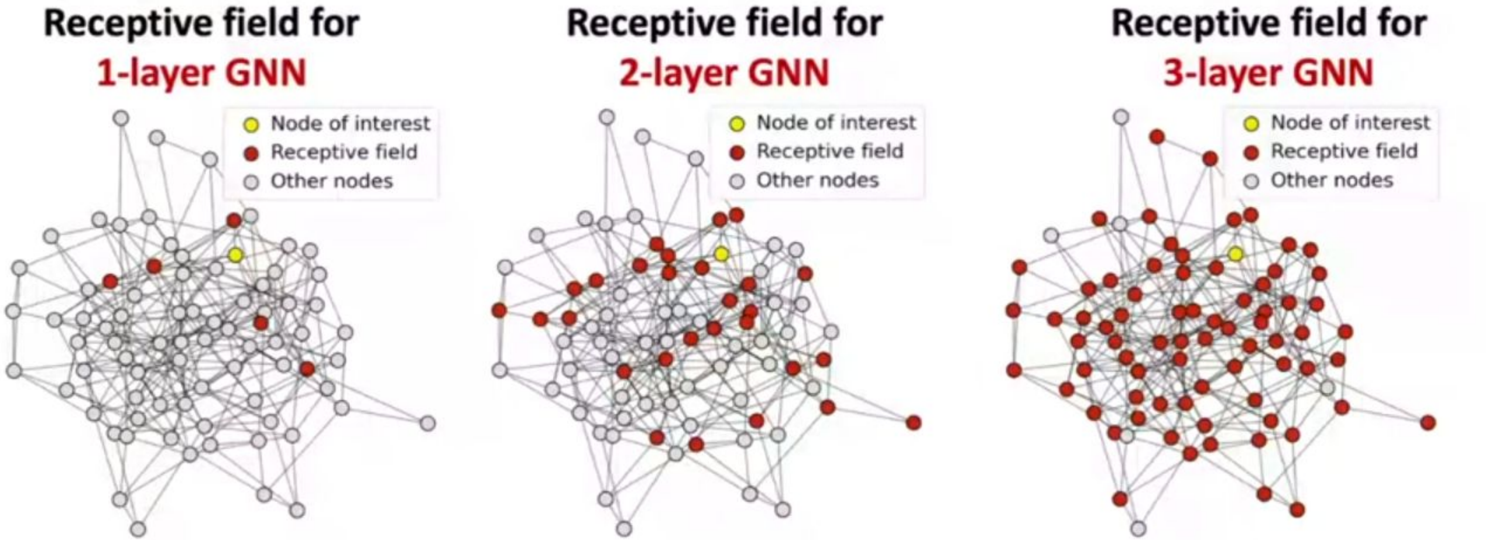
GNN Layer = Message + Aggregation



	Aggregation functions (neural networks). Can be the same or different for each layer.
	Message transformation function (optional). Simple operations, gates (GRU, LSTM)...

How do GNNs work? (IV)

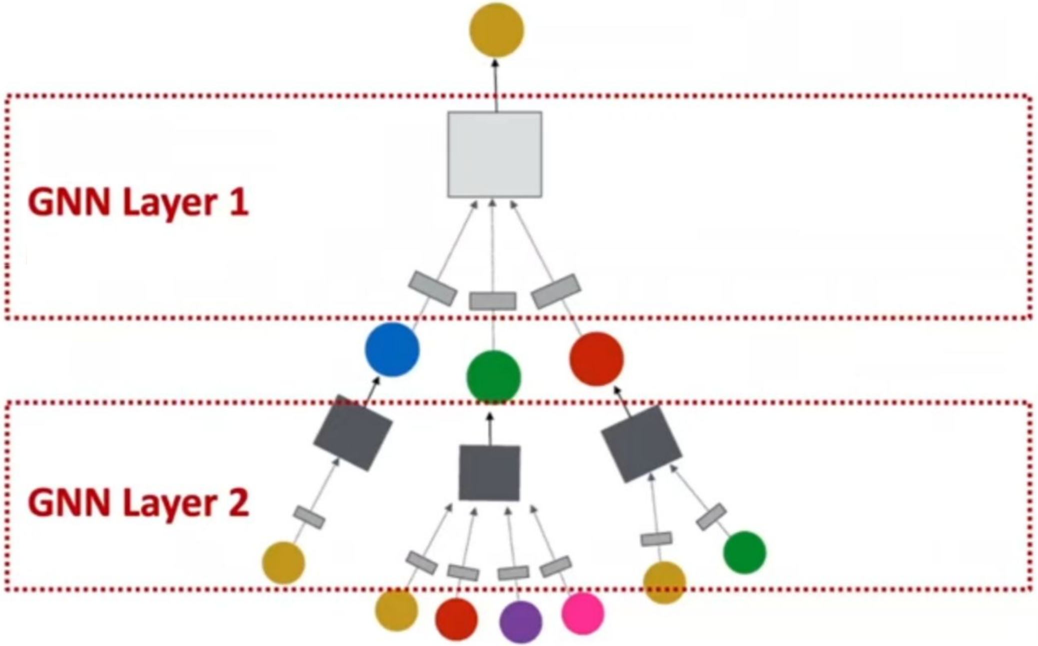
Problem! Oversmoothing: in the 3-layer GNN all node values converge to the same value.

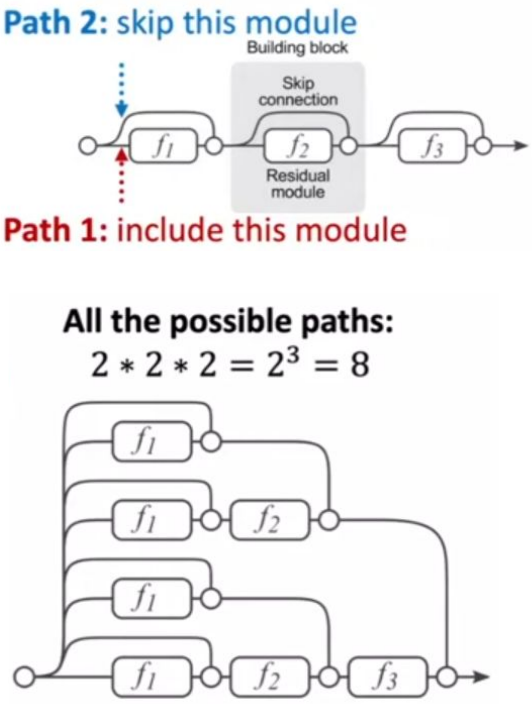


How do GNNs work? (V)

Connect GNN layers:

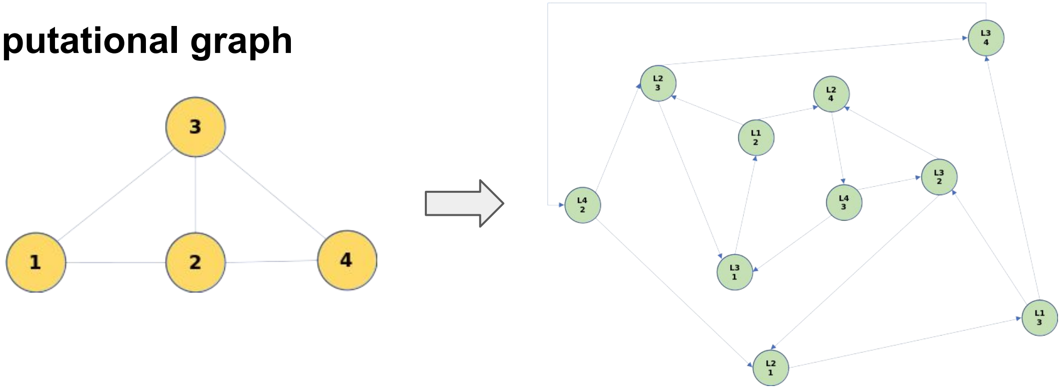
- Stack layers sequentially.
- Ways of adding skip connections to improve generalization.





How do GNNs work? (VI)

Raw input graph ≠ computational graph



Techniques to adapt the data to a GNN, helping with generalization:

- Graph **feature augmentation**: new features from the existing ones.
- Graph **structure manipulation**: new nodes or edges, or completely rebuilt graph based on original data.

GNN training: supervised/unsupervised.
GNN predictions: **node/edge/graph** level.

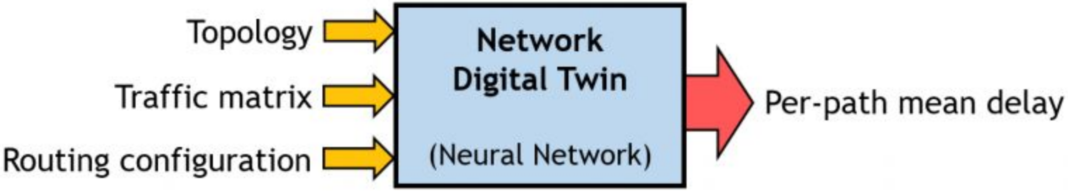
GNN use case

Performance prediction in larger unseen networks

- **GNNs generalization problem** → GNN complexity and errors increase with network size.
- **This work started with our participation in the AI/ML in 5G ITU Graph Neural Networking Challenge 2021.** Creating a Scalable Network Digital Twin. Further work after the competition was realized.
- **Joint work with the University of Antwerp, resulting in GAIN team** → Girona & Antwerp Intelligence for Networking.
- Conference publication: Farreras, M., Soto, P., Camelo, M., Fàbrega, L., Vilà, P. (2022). *Predicting network performance using GNNs: generalization to larger unseen networks*. https://doi.org/10.1109/NOMS54207.2022.9789766
- Journal publication: in preparation

Problem description

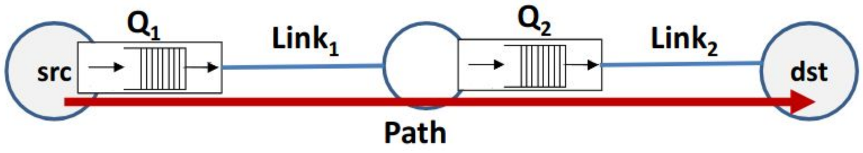
- GNN baseline, called RouteNet, provided by the challenge organizers (Barcelona Neural Networking Center, UPC).
- RouteNet predicts per-path mean delay, jitter, loss; here the focus is on per-path mean delay.
- Efficient to generalize with topologies, routings and traffic not seen before.
- **But poor generalization** when graph size and features are bigger than training samples.
- **Link and path features** available.
- The prediction error was measured using the Mean Average Percentage Error (MAPE).
- RouteNet baseline achieved 187% MAPE.
- Dataset: small networks for training and larger networks for validation/testing (higher number of nodes, higher link capacities, longer paths)



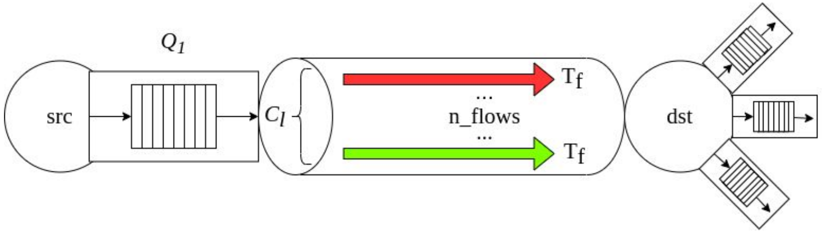
$$MAPE = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{\hat{y}_l - y_i}{y_i} \right|$$

Our improvements

- Use **link features** instead of path features, postprocessing of the GNN result:



- **Min-max normalization** of predictor features: analysis of train, validation and test datasets.
- **Feature selection** based on correlation tests.
- **Feature augmentation**: creation of a feature called offered traffic intensity of a link. Sum of flows traversing a link in a [0..1] range:



- **Hyper-parameter optimization**: less training time and GPU/power consumption.

Results

TEST DATASET MAPE (%) RESULTS AND PERFORMANCE FOR BASELINE AND EACH GAIN SOLUTION

	Full testing dataset	S_1	S_1 **50 nodes**	S_1 **300 nodes**	S_2	S_2 **50 nodes**	S_2 **300 nodes**	S_3	S_3 **50 nodes**	S_3 **300 nodes**	**Train time**
Baseline	187.28	79.145	68.481	92.979	253.075	68.481	345.135	247.217	44.669	368.019	12h 15m
GAIN 1	44.73	13.074	11.705	13.108	54.318	16.214	42.374	67.44	70.994	90.739	9h 45m
GAIN 2	28.739	11.719	11.026	11.864	35.067	17.092	39.773	31.754	17.581	31.353	9h 48m
GAIN 3	18.471	9.436	6.893	12.468	26.897	22.106	30.067	18.143	12.569	21.862	9h 40m
GAIN 4	2.612	2.652	1.539	3.687	2.492	1.386	2.567	2.584	1.607	2.363	3h 25m
GAIN 5	**1.838**	**1.407**	**1.111**	**1.808**	**1.929**	**1.573**	**1.535**	**1.756**	**1.388**	**1.462**	**2h 20m**

GAIN 1: Inferring per-path delay from predicted queue occupancy
GAIN 2: Normalization of predictor features
GAIN 3: Feature selection
GAIN 4: Feature augmentation: Offered Traffic Intensity
GAIN 5: Hyper-parameters optimization

Setting 1 (S_1): Longer paths.
Setting 2 (S_2): Increased link capacity.
Setting 3 (S_3): Both properties mixed.

Future work

- Improve the previous work to achieve greater performance and evaluate the implications.
- Closed-loop control context for SDN-NFV.
- Investigate other GNN architectures: different input graphs and newer GNN implementations.
- Generation of network slicing datasets.
- Adaptation of a GNN model to orchestrate a network slicing infrastructure.

Any questions?