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:

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

Intuition: Nodes aggregate information from their neighbors using neural networks.
How do GNNs work? (II)

Intuition: Nodes **aggregate** information from their **neighbors** using 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.
- 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)

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

<p>| Test dataset MAPE (%) results and performance for baseline and each GAIN solution |
|---------------------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|</p>
<table>
<thead>
<tr>
<th></th>
<th>Full testing dataset</th>
<th>$S_1$</th>
<th>$S_1$ 50 nodes</th>
<th>$S_1$ 300 nodes</th>
<th>$S_2$</th>
<th>$S_2$ 50 nodes</th>
<th>$S_2$ 300 nodes</th>
<th>$S_3$</th>
<th>$S_3$ 50 nodes</th>
<th>$S_3$ 300 nodes</th>
<th>Train time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>187.28</td>
<td>79.145</td>
<td>68.481</td>
<td>92.979</td>
<td>253.075</td>
<td>68.481</td>
<td>345.135</td>
<td>247.217</td>
<td>44.669</td>
<td>368.019</td>
<td>12h 15m</td>
</tr>
<tr>
<td>GAIN 1</td>
<td>44.73</td>
<td>13.074</td>
<td>11.705</td>
<td>13.108</td>
<td>54.318</td>
<td>16.214</td>
<td>42.374</td>
<td>67.44</td>
<td>70.994</td>
<td>90.739</td>
<td>9h 45m</td>
</tr>
<tr>
<td>GAIN 4</td>
<td>2.612</td>
<td>2.652</td>
<td>1.539</td>
<td>3.687</td>
<td>2.492</td>
<td>1.386</td>
<td>2.567</td>
<td>2.584</td>
<td>1.607</td>
<td>2.363</td>
<td>3h 25m</td>
</tr>
<tr>
<td>GAIN 5</td>
<td>1.838</td>
<td>1.407</td>
<td>1.111</td>
<td>1.808</td>
<td>1.929</td>
<td>1.573</td>
<td>1.535</td>
<td>1.756</td>
<td>1.388</td>
<td>1.462</td>
<td>2h 20m</td>
</tr>
</tbody>
</table>

**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?