Benchmarking Graph Neural Networks for Internet Routing Data

Dimitrios P. Giakatos, Sofia Kostoglou,

Pavlos Sermpezis, Athena Vakali

Data and Web Science Lab

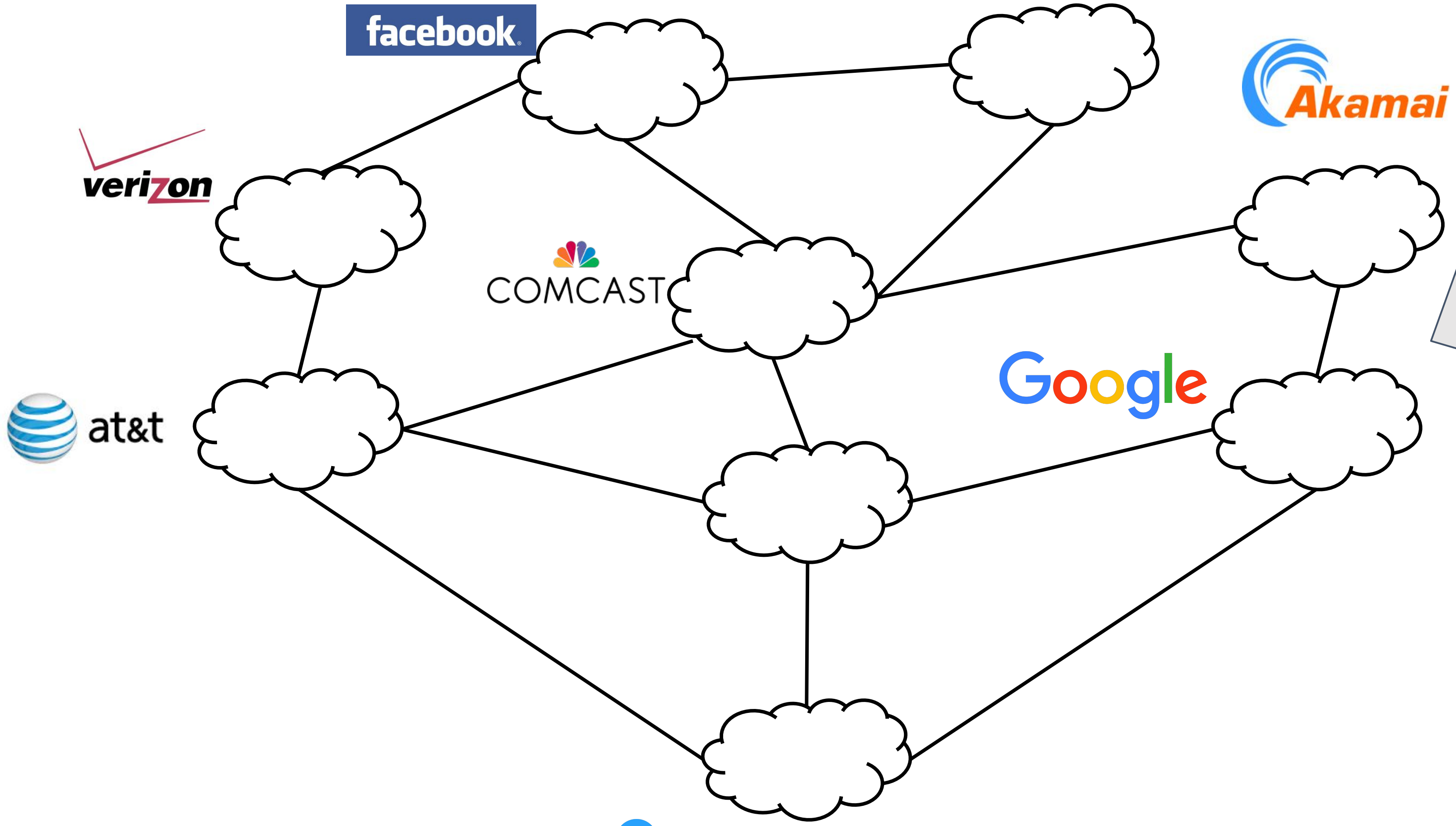
(https://datalab.csd.auth.gr/)

Aristotle University of Thessaloniki, Greece

DATA&WEB SCIENCE LABORATORY

ARISTOTLE UNIVERSITY OF THESSALONIKI

The Internet (as a graph)



The Internet is a network of networks (also called *Autonomous Systems*, or *ASes*)

- > 70k ASes (i.e., ***nodes***)
- > 400k links (i.e., ***edges***)

There exists a lot of information for each network (i.e. ***node features***) in public datasets

GNNs for Internet routing data: state of the art

The are no works on GNNs for Internet routing data !

GraphML for Internet routing data: state of the art

- methods using simple graph properties, e.g., [*Sanchez et. al., BigDAMA (CoNEXT), 2019*]
    - using, e.g., centrality, degree, clustering coef., etc., as node features

- **bgp2vec** [*Shapira and Shavitt, IEEE TNSM, 2022*]
    - mechanism similar to the ***word2vec*** model
    - uses *BGP messages* (publicly available) → i.e., uses *AS paths* as lists of node sequences

GNNs for Internet data: Mind the gap!

- Lack of GNN methods for Internet routing data → Why?

- Probable reason [*disclaimer: our conjecture*]
  - Need for expertise in both domains: Internet routing & GNNs

  - GNN libraries are open and well-documented, but …
    → *are Internet researchers aware of all practical intricacies of this (relatively) new technology?*

  - Internet data are public, but …
    → *do GNN researchers know where to find them & how to collect/process them?*

Contributions

- Benchmark dataset
  - Collect multiple data: graph (edgelist) & node attributes
  - Preprocessing → a “ready to use” dataset for GNN models

- GNN benchmarking & initial insights
  - Test GNNs on Internet routing data
  - Do baseline GNN models perform well (vs. state-of-the-art)?

Benchmark dataset

Internet routing: data sources

- CAIDA AS-rank
  - Network information: location, network size, topology, etc.
- CAIDA AS-relationships
  - Graph information: edgelist (i.e., peering links)
- Peering DB
  - Network information: connectivity, network type, traffic, etc.
- AS hegemony
  - Network information: size, topology
- Country-level Transit Influence (CTI)
  - Network information: size, topology
- ASDB
  - Network information: network types

Graph
(~74k nodes,
~462k edges)
&
19 node features
(12 numerical,
7 categorical)

The compiled dataset

← **19 node features** (12 numerical, 7 categorical)→

↓ **74k networks (nodes)** ↑

ASN	Location-related information		Network-size related information			Topology-related information		IXP-related information		Network type-related information		
	RIR Region	Continent	Customer cone (in #ASNs)	AS hegemony	...	#neighbors (in #ASNs)	...	#IXPs connected to	...	Net. type (PeeringDB)	Net. type (ASDB)	...
174	ARIN	North America	32457	0.09	...	6614	...	0	...	NSP	ICT	...
1299	RIPE	Europe	37162	0.10		2328		0		NSP	ICT	
2497	APNIC	Asia	507	0.01		338		16		NSP	NaN	
3320	RIPE	Europe	3015	0.01		667		5		NSP	ICT	
3333	RIPE	Europe	3	0.00		320		1		Non-profit	ICT	
5470	RIPE	Europe	1	0.00		1		NaN		NaN	Education & Research	
15169	ARIN	North America	12	0.01		366		214		Content	ICT	
...	...	...	...	...		...		...		...		...

The "problem" for GNNs & the preprocessing

- Categorical features
  → one hot encoding
- Numerical features
  - large heterogeneity
  - heavy tail distributions

  → log(x+1) & MinMaxScaler
- Graph
  - many leaf nodes (~40%)
  - density < 0.01%

  → remove leaf nodes

RIR Region
ARIN
RIPE
APNIC
RIPE
RIPE
RIPE
ARIN
...

RIR Region			
1	0	...	0
0	1	...	0
0	0	...	1
0	1	...	0
0	1	...	0
0	1	...	0
1	0	...	0
...			

Network-size related information		
Customer cone (in #ASNs)	AS hegemony	...
32457	0.09	...
37162	0.10	
507	0.01	
3015	0.01	
3	0.00	
1	0.00	
12	0.01	
...	...	

Network-size related information		
Customer cone	AS hegemony	...
0.90	0.09	...
0.80	0.10	
0.01	0.01	
0.01	0.01	
0.05	0.00	
0.01	0.00	
0.20	0.01	
...	...	

74k nodes
460k edges

46k nodes
430k edges

The **benchmark dataset** (i.e., after preprocessing)

← **72 node features** (all numerical and in range [0,1])→

← 46k networks (nodes) →

ASN	Location-related information							Network-size related information			Topology-related information		IXP-related information		Network type-related information				
	RIR Region				Continent			Customer cone	AS hegemony	...	#neighbors	...	#IXPs	...	Net. type				...
174	1	0	...	0	0	...	0	0.90	0.09	...	0.95	...	0	...	1	0	...	0	...
1299	0	1	...	0	1	...	0	0.80	0.10		0.73		0		0	1	...	0	
2497	0	0	...	1	0	...	1	0.01	0.01		0.30		0.15		0	0	...	1	
3320	0	1	...	0	1	...	0	0.01	0.01		0.60		0.08		0	1	...	0	
3333	0	1	...	0	1	...	0	0.05	0.00		0.11		0.01		0	1	...	0	
5470	0	1	...	0	1	...	0	0.01	0.00		0.01		0		0	1	...	0	
15169	1	0	...	0	0	...	0	0.20	0.01		0.13		0.57		1	0	...	0	
...	...							...	...		...		...		...				...

Benchmarking GNNs

Benchmarking methodology

- Learning tasks
  - Link prediction (why? → view of AS graph is incomplete)
  - Node classification (why? → PeeringDB is incomplete)
    - 4 classification tasks: traffic ratio, network scope, network type, peering policy

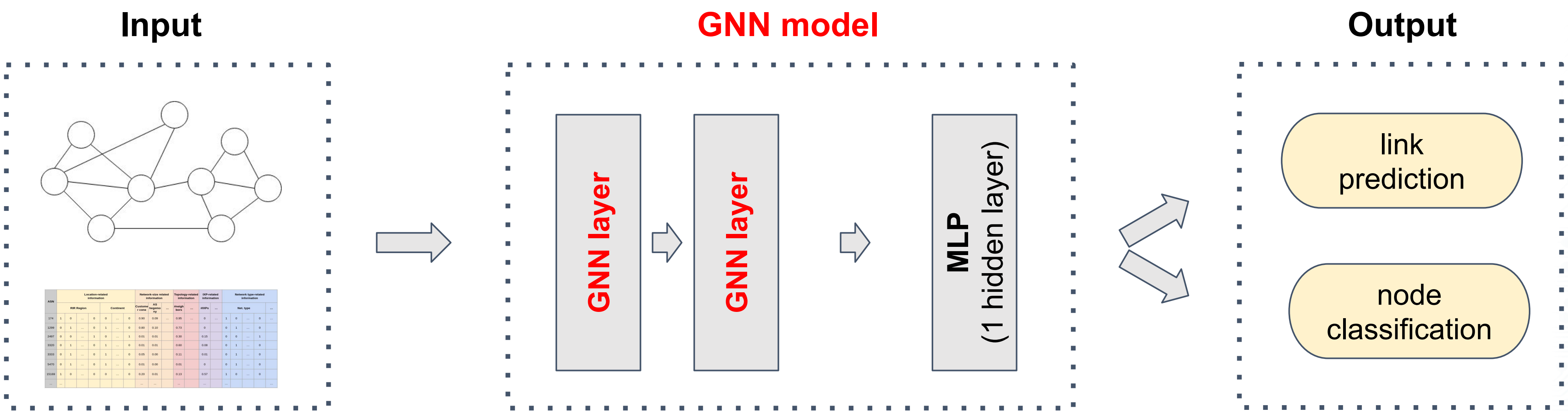
- ML models
  - GraphSage
  - GCN
  - GAT
  - node2vec
  - bgp2vec
  - random forest

GNNs: GraphSage, GCN, GAT

graph embedding models (no node features): node2vec, bgp2vec

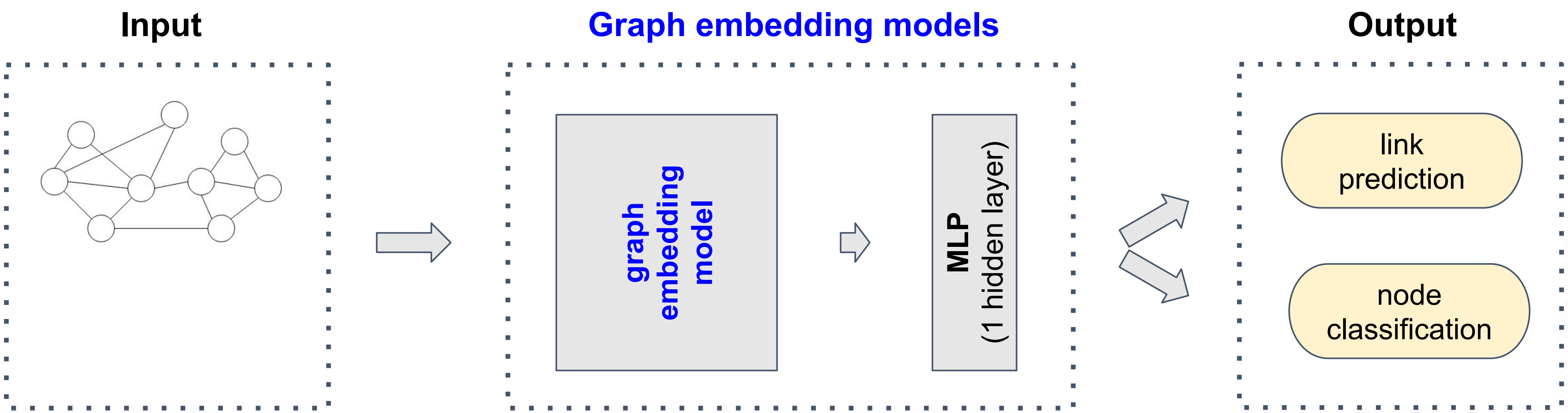
Benchmarking methodology: GNN models

- 2 GNN layers (GraphSage, GCN, or GAT) & MLP (1 hidden layer)
- 32-dimensional node embeddings
- cross-entropy
- light tuning (no dropout, learning rate 0.01, few 100s epochs)



Benchmarking methodology: Graph embedding models

- node2vec & bgp2vec → input: only graph information, no node features
  - random walks on AS-graph [node2vec]
  - BGP messages from route collectors [bgp2vec]
- 16-dimensional node embeddings



Benchmarking methodology: Random Forest

- input: only node features, no graph information

Results: Link prediction

Table 3: Results for the link prediction task: average AUC, Recall, and Precision metrics over 10 runs per model.

Model	AUC	Recall ($\frac{\#TP}{\#P}$)	Precision ($\frac{\#TP}{\#TP+\#FP}$)
GraphSAGE	94.7%	82.7%	86.8%
GCN	95.3%	85.5%	95.9%
GAT	64.4%	24.2%	28.4%
node2vec	86.5%	82.7%	95.5%
bgp2vec	93.0%	85.5%	91.5%
Rnd. forest	96.2%	24.2%	96.3%

GraphSAGE & **GCN** are very efficient for link prediction

bgp2vec is also efficient for link prediction

RF does not mispredict links (high AUC), but only predicts ¼ of the actual links (low Recall)

Key insights:

- the graph information (see bgp2vec) is good enough for link prediction
- GNNs with light tuning can be equally good or better than state-of-the-art
  → potential for improvement

Results: Link prediction

Table 4: Detailed link prediction results (Recall / Precision) for node pairs of different size of neighborhoods.

	Low	Medium	High
Low	31.6% / 89.8%	80.7% / 98.2%	67.2% / 90.5%
Medium		94.4% / 98.1%	94.4% / 95.4%
High			99.5% / 99.0%

Low: networks with < 10 neighbors
Medium: >10 and < 20 neighbors
High: >20 neighbors

80% of networks are in the **low** category (10% **medium**, 10% **high**)

Predicting links between nodes of medium/high degrees is easy

Predicting links between nodes of low degrees is a more difficult task
→ why?

Key insights:

- need for GNN models that are efficient for all parts of the graph
- How to design them? Is it even feasible?

Results: Node classification

Table 5: Results for the node classification tasks: average accuracy (ACC) and F1 score metrics over 10 runs per model.

Model	Traffic ratio (PDB)		Scope (PDB)		Network type (PDB)		Peering policy (PDB)	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1
GraphSAGE	44.8%	35.9%	49.2%	47.2%	54.7%	53.1%	34.9%	30.6%
GCN	38.7%	30.5%	40.1%	37.1%	46.6%	44.6%	37.0%	31.1%
GAT	38.0%	31.3%	41.8%	38.4%	49.9%	47.2%	32.2%	28.8%
node2vec	20.9%	19.4%	16.3%	15.7%	19.0%	18.6%	31.1%	27.0%
bgp2vec	14.8%	13.4%	14.8%	13.0%	19.9%	19.1%	29.4%	26.8%
Rnd. Forest	51.1%	35.8%	36.6%	33.7%	49.8%	42.9%	54.6%	34.8%

multi-class classification (3 to 7 classes)

GraphSAGE still performs well;
GAT performs quite well now
(vs. link prediction)

Models that use only graph information (and no node features) underperform in the node classification tasks

RF performs better than GNNs in predicting some features
→ *are these features not strongly related to graph structure?*

Key insights:

- only graph information (state-of-the-art) is not enough for node classification tasks in Internet data
- there is still a lot of room for improvement (see low accuracy) & research on GNN for Internet data

Summarizing…

https://github.com/dpgiakatos/gnn-internet-data/

- **Benchmark dataset** for Internet routing data
  - large dataset: 74k nodes, 460k edges, 19 node features
  - “ready-to-use”: no need for collection/processing → easier for researchers
  - use as a common “standard” dataset to compare methods

- **Benchmarking GNNs** on Internet routing data tasks
  - First test of GNNs on Internet data
  - Even simple GNNs can outperform state-of-the-art methods (bgp2vec)
  - Still a lot of room for improvement & research
    - e.g,. how to design GNNs for the special structure of the Internet AS-graph?
    - e.g., how to treat the non randomly missing information (i.e., bias) of the Internet data?

- What about a GNNet challenge for Internet routing data?
  - feedback & how-to