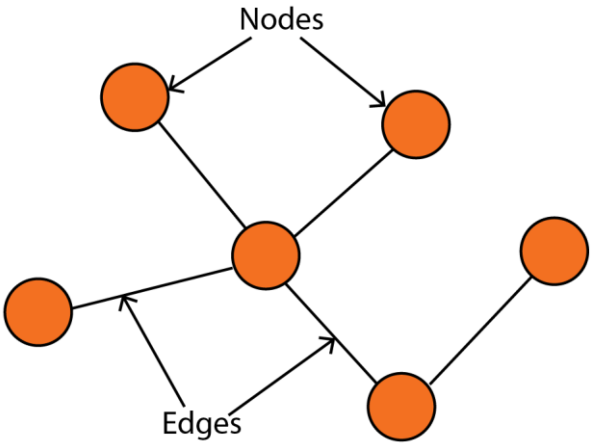
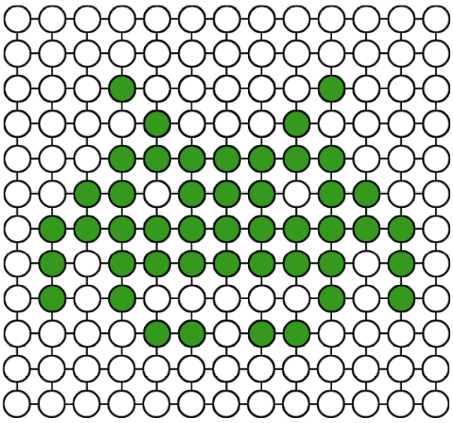
# Chapter 1: Getting Started with Graph Learning

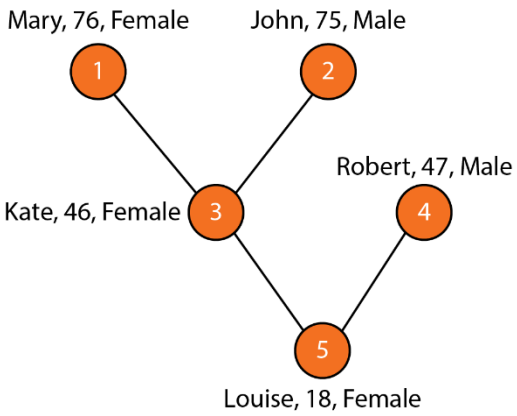




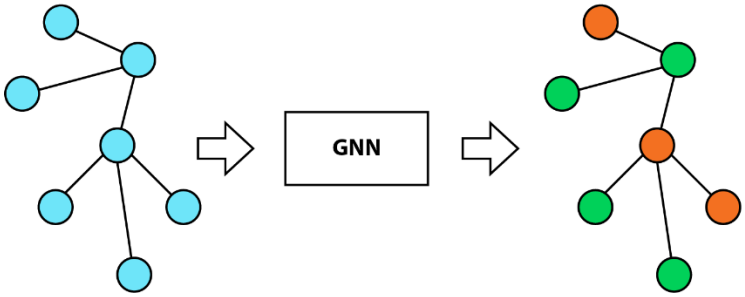
**Tabular dataset**

| ID | Name | Age | Gender |
|---|---|---|---|
| 1 | Mary | 76 | Female |
| 2 | John | 75 | Male |
| 3 | Kate | 46 | Female |
| 4 | Robert | 47 | Male |
| 5 | Loise | 18 | Female |

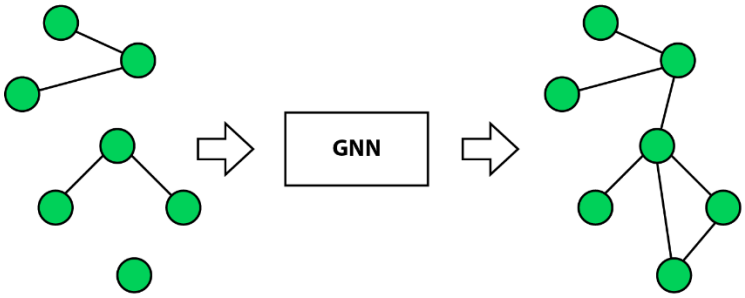
**Graph dataset**



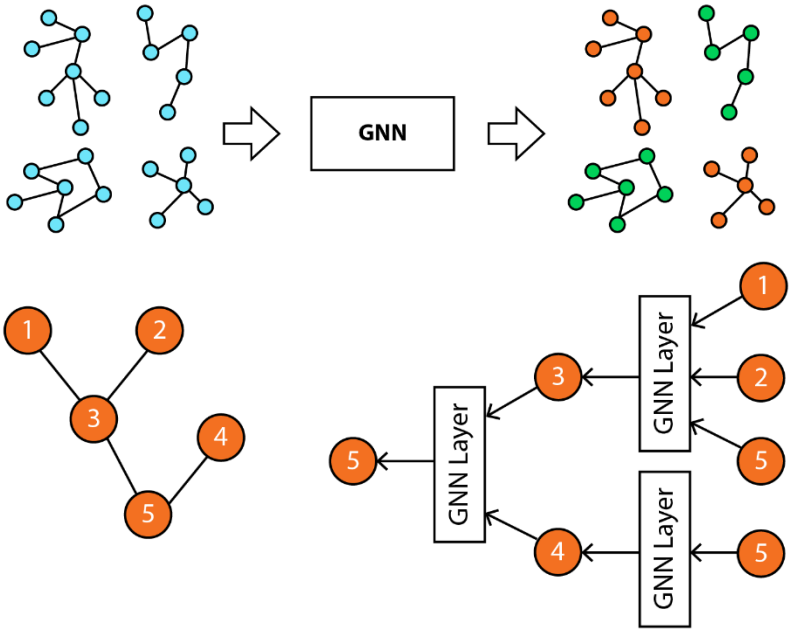
a. Node classification



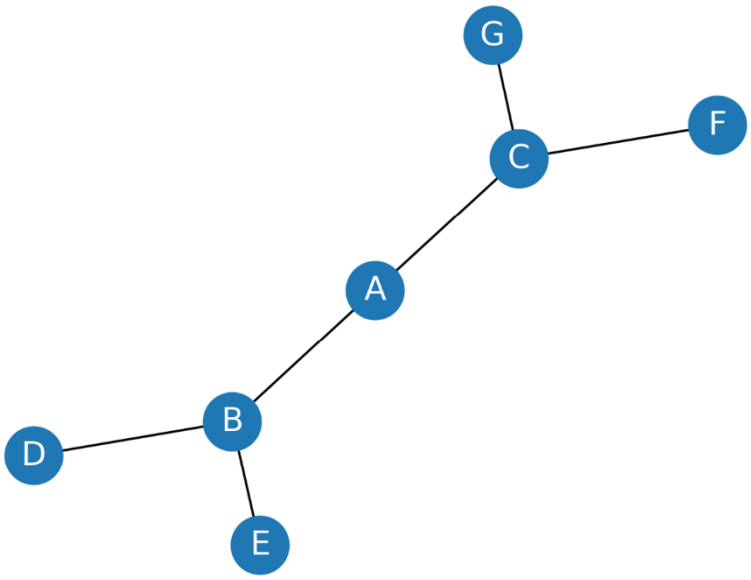

b. Link prediction

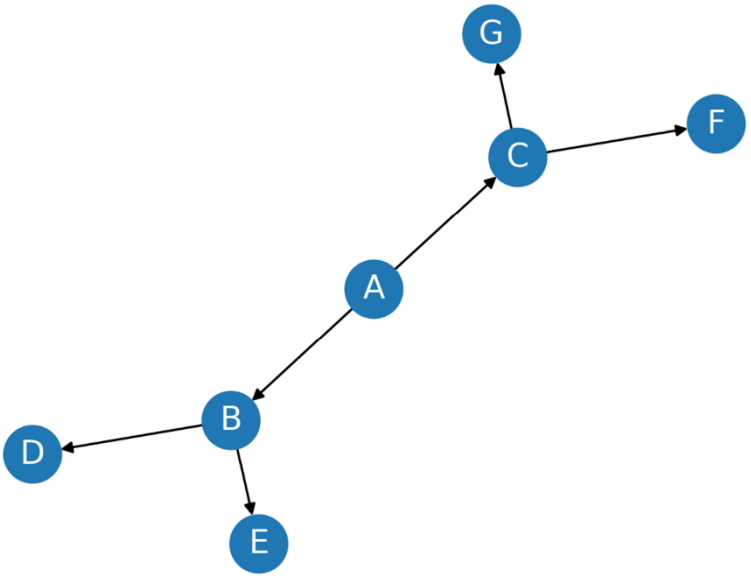


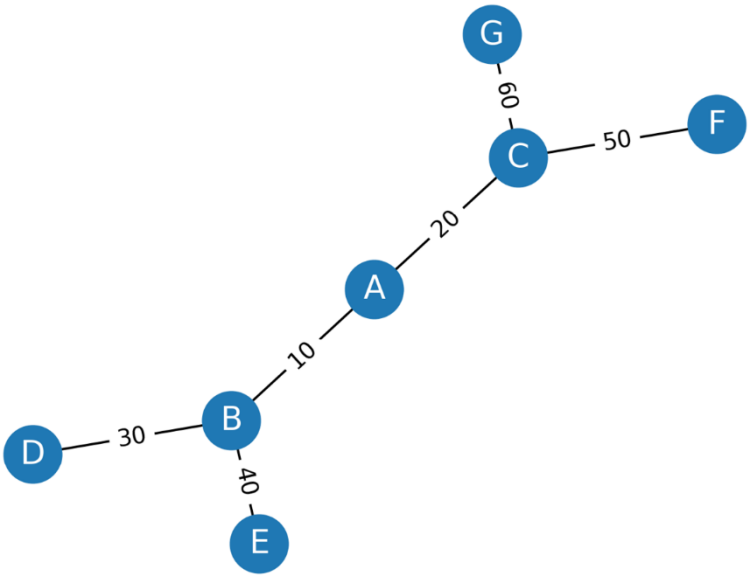

c. Graph classification

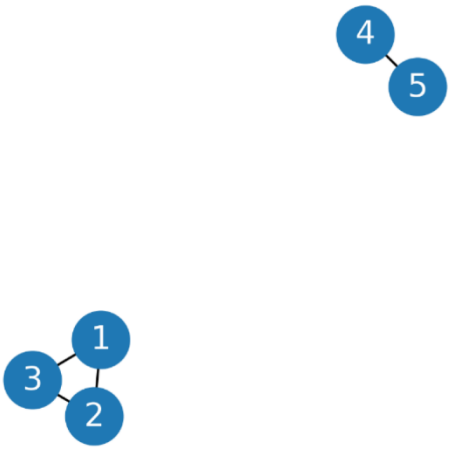


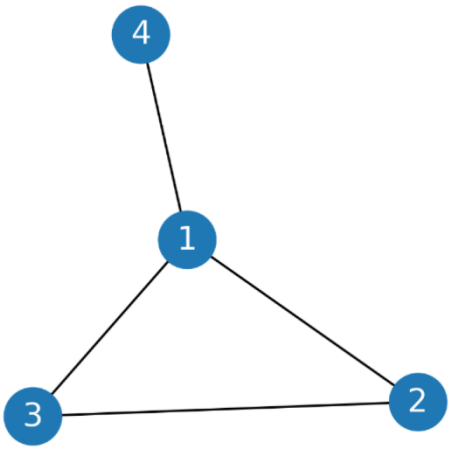
# Chapter 2: Graph Theory for Graph Neural Networks



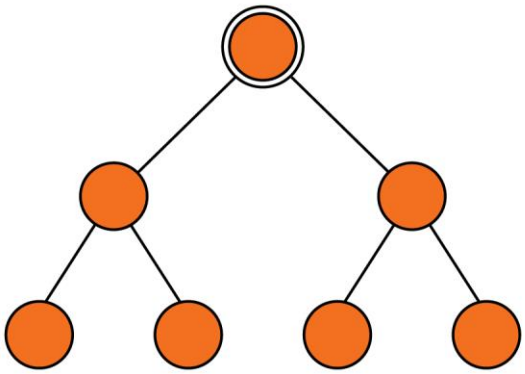



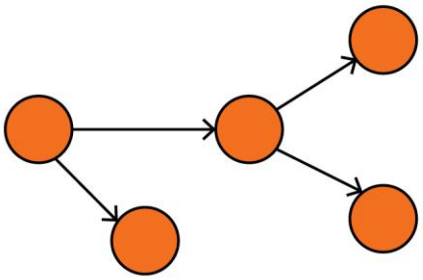



Rooted tree/tree

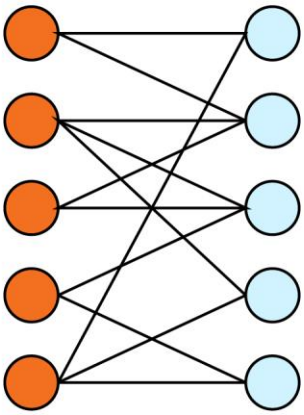
Directed acyclic graph

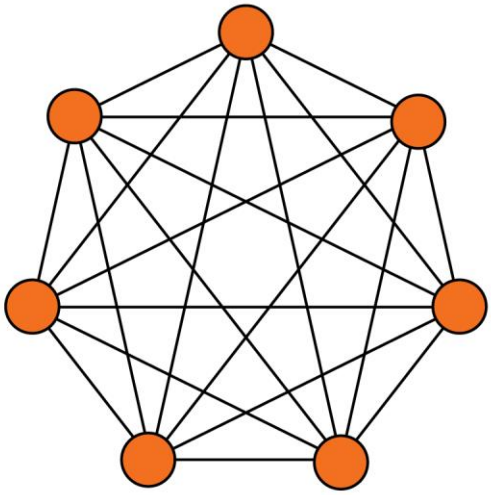


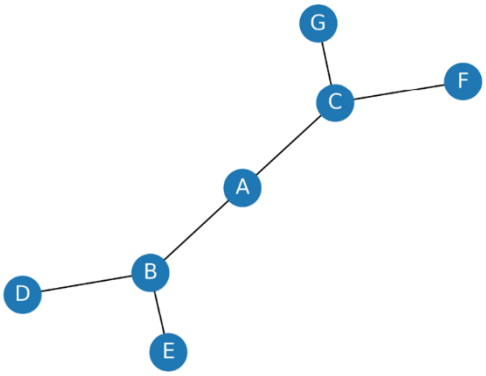


Bipartite graph

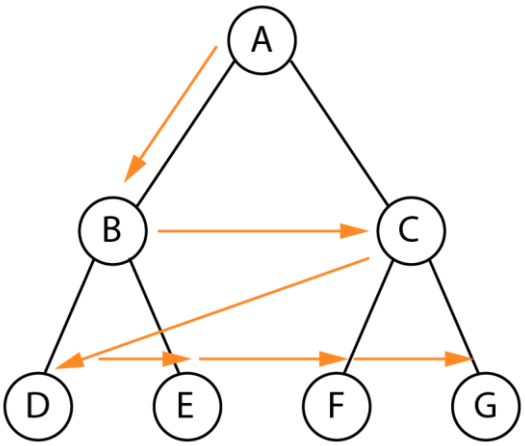
Complete graph

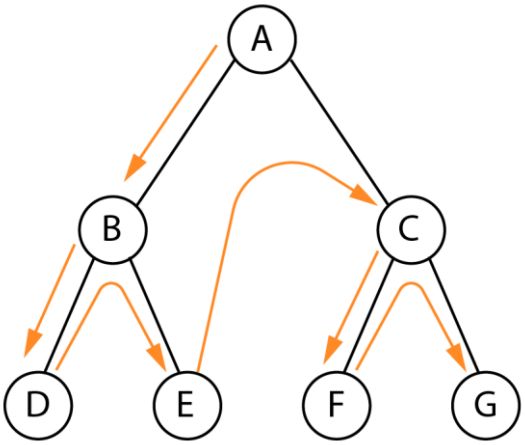




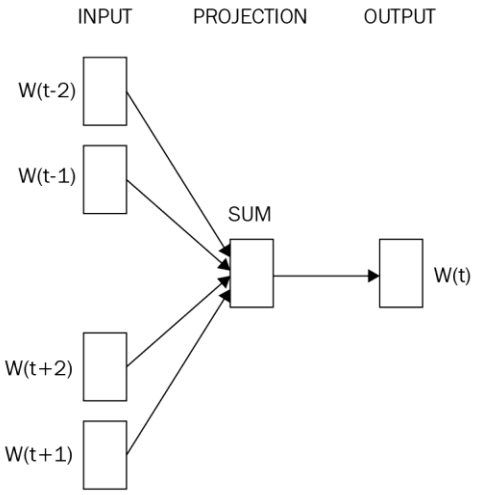



$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

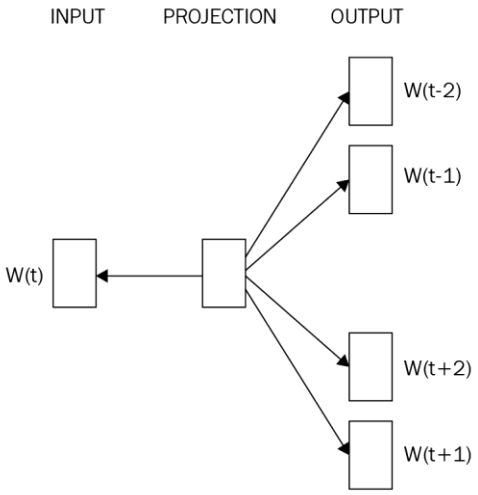


# Chapter 3: Creating Node Representations with DeepWalk

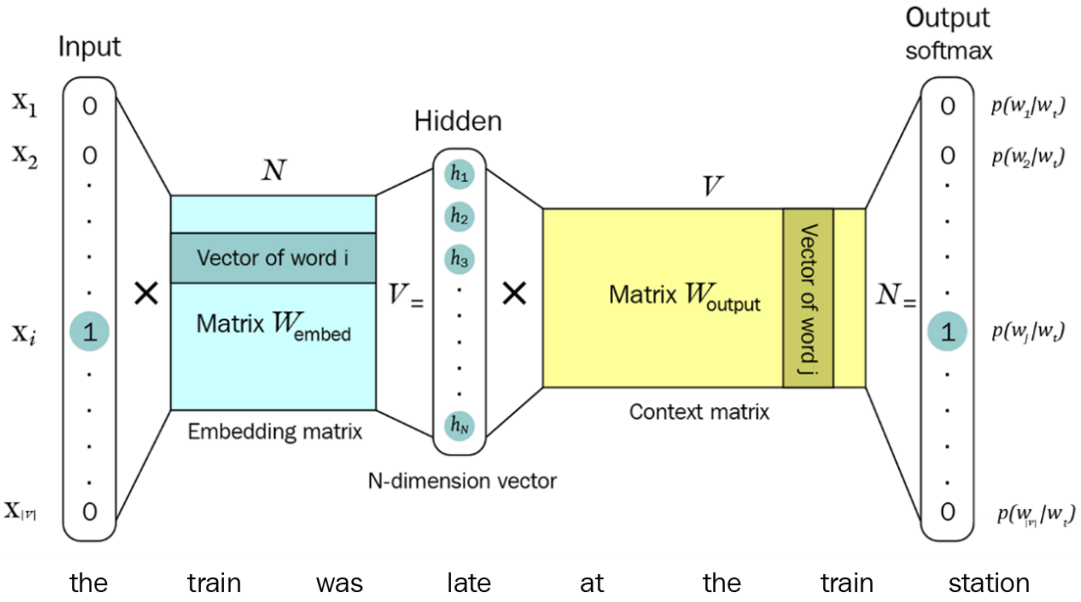


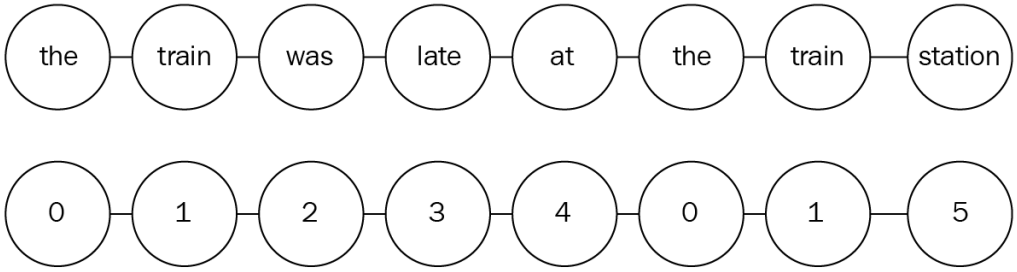

**CBOW**

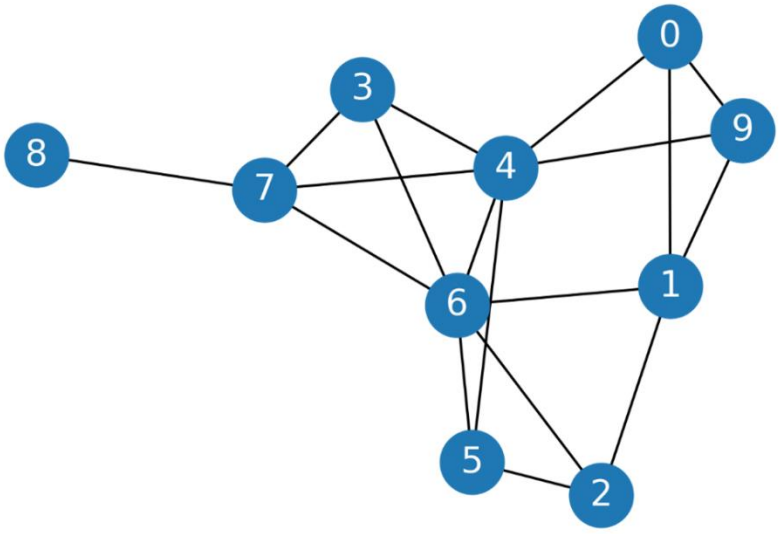


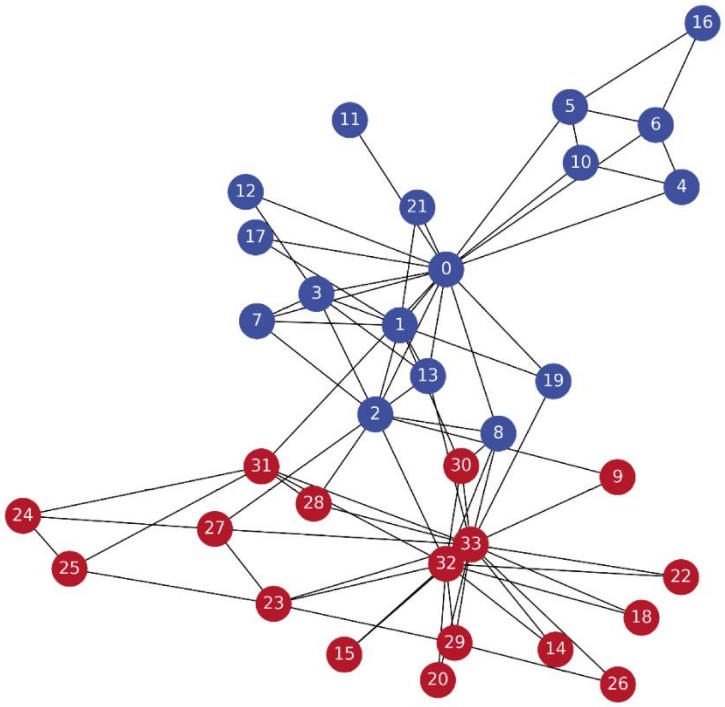

**Skip-gram**

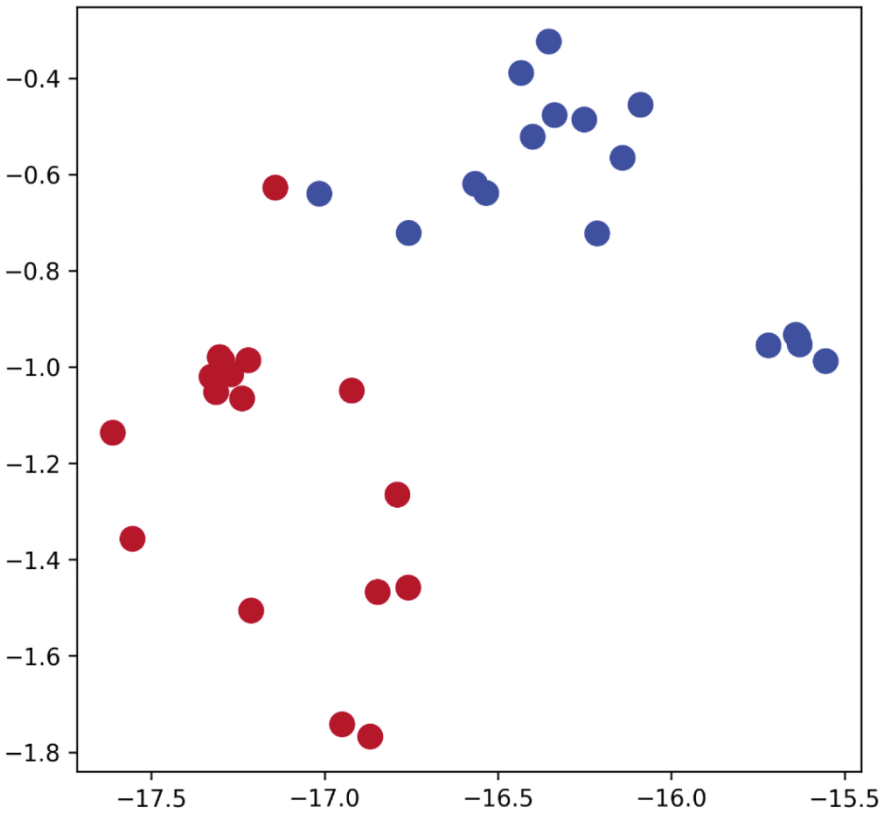
| Context Size | Text | Skip-grams |
|---|---|---|
| 1 | **the *train*** was late. | (’the’, ‘train’) |
| | ***the* train *was*** late | (‘train’, ’the’)<br>(‘train’, ’was’) |
| | the ***train* was *late*** | (’was’, ‘train’)<br>(’was’, ’late’) |
| | the train ***was* late** | (’late’, ‘was’) |
| 2 | **the *train was*** late | (’the’, ‘train’)<br>(’the’, ’was’) |
| | ***the* train *was late*** | (‘train’, ’the’)<br>(‘train’, ’was’)<br>(‘train’, ’late’) |
| | ***the train* was *late*** | (’was’, ’the’)<br>(’was’, ‘train’)<br>(’was’, ’late’) |
| | the ***train was* late** | (’late’, ‘train’)<br>(’late’, ‘was’) |


Input
Output
softmax
Hidden
N
V
Vector of word i
Matrix $W_{embed}$
Embedding matrix
Matrix $W_{output}$
Vector of word j
Context matrix
N-dimension vector
$x_1$
$x_2$
$x_i$
$x_{|V|}$
$h_1$
$h_2$
$h_3$
$h_N$
$p(w_1|w_t)$
$p(w_2|w_t)$
$p(w_j|w_t)$
$p(w_{|V|}|w_t)$

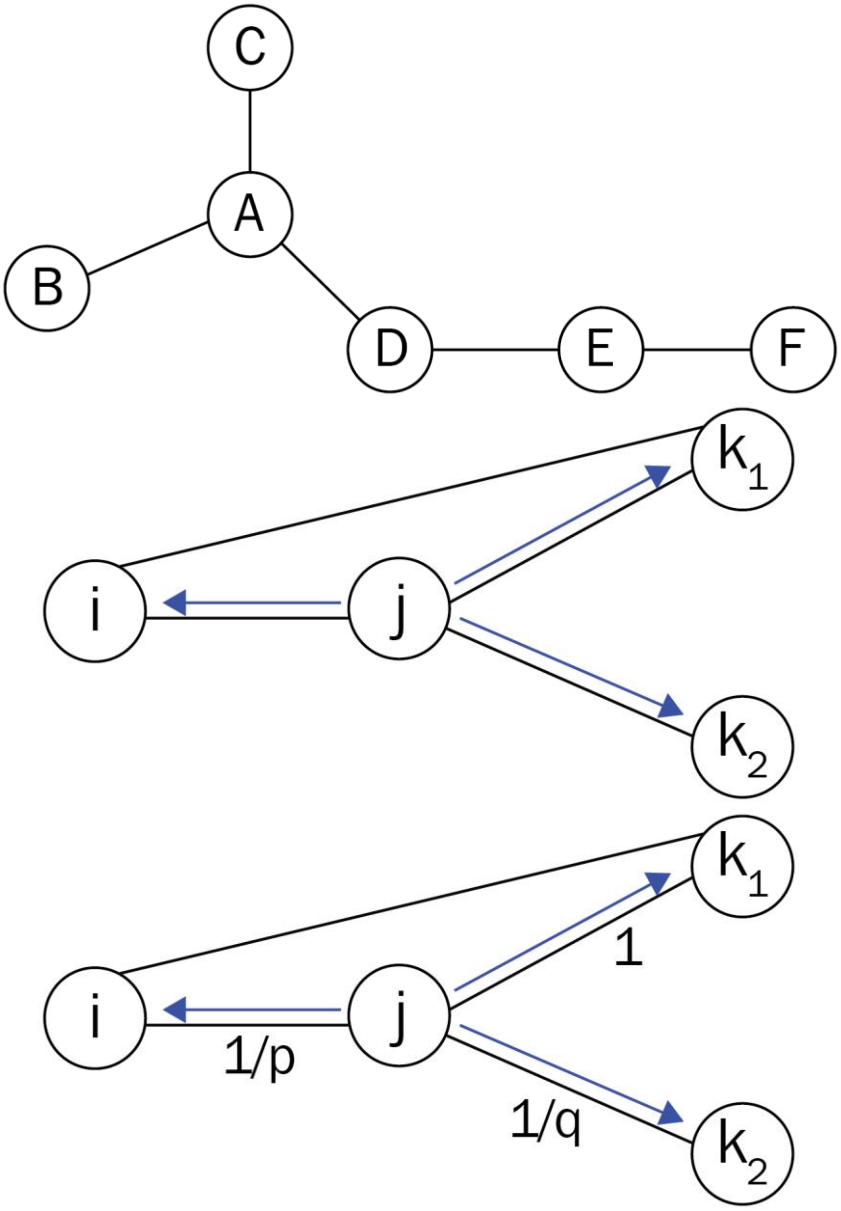

the
train
was
late
at
the
train
station
0
1
2
3
4
0
1
5

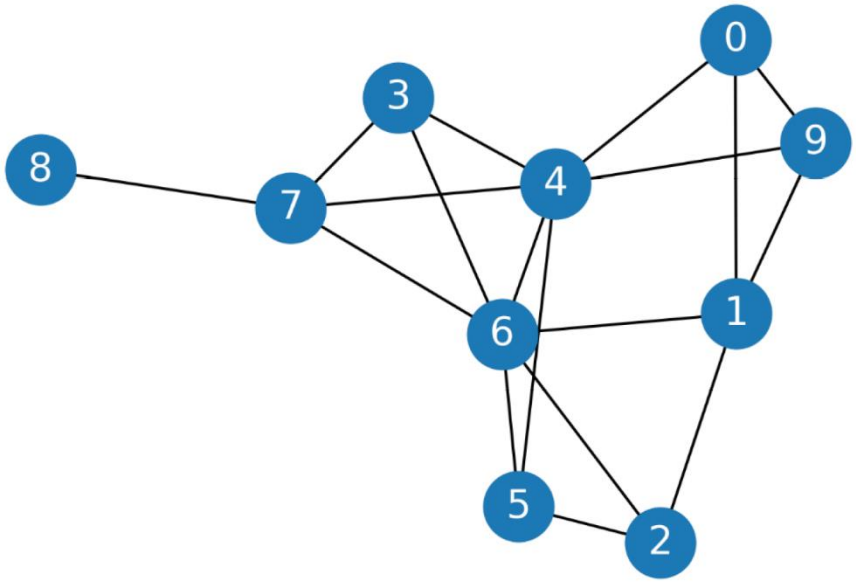

0
1
2
3
4
5
6
7
8
9


16
5
6
11
10
4
12
21
17
0
3
7
1
13
19
2
8
31
30
9
28
24
27
33
32
22
25
23
18
29
15
14
20
26


−0.4
−0.6
−0.8
−1.0
−1.2
−1.4
−1.6
−1.8
−17.5
−17.0
−16.5
−16.0
−15.5

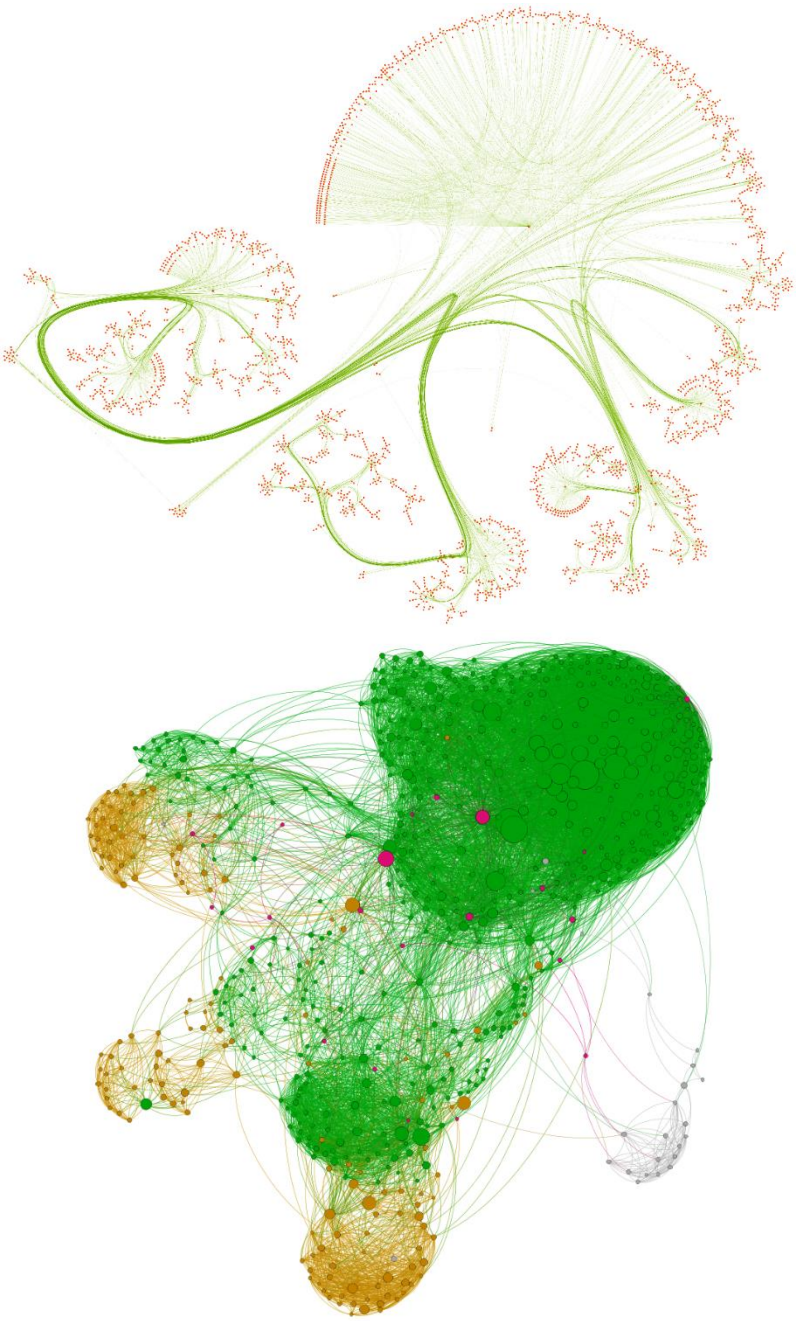
## Chapter 4: Improving Embeddings with Biased Random Walks in Node2Vec



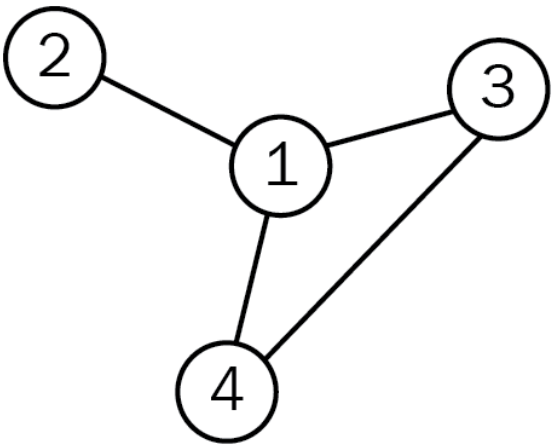


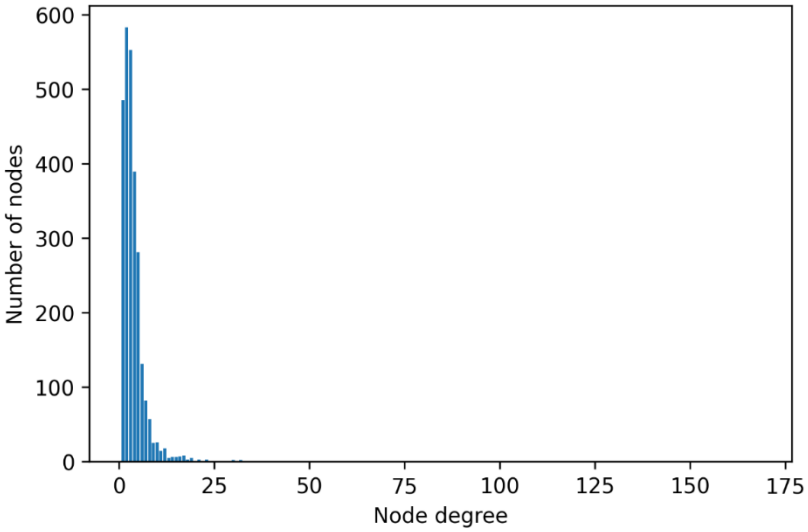

| | p=1 | p=2 | p=3 | p=4 | p=5 | p=6 | p=7 |
|---|---|---|---|---|---|---|---|
| q=1 | 92.95% (± 4.61%) | 94.45% (± 4.19%) | 96.36% (± 4.69%) | 95.41% (± 4.14%) | 95.59% (± 4.30%) | 95.82% (± 4.67%) | 95.41% (± 3.94%) |
| q=2 | 93.64% (± 4.36%) | 93.95% (± 3.97%) | 95.09% (± 4.34%) | 95.55% (± 3.80%) | 96.27% (± 3.82%) | 96.18% (± 3.90%) | 97.45% (± 3.60%) |
| q=3 | 93.45% (± 3.82%) | 94.41% (± 4.11%) | 95.77% (± 3.59%) | 95.27% (± 3.63%) | 96.68% (± 3.90%) | 95.64% (± 3.69%) | 96.00% (± 3.82%) |
| q=4 | 94.14% (± 3.93%) | 94.14% (± 3.93%) | 95.45% (± 3.40%) | 95.05% (± 3.58%) | 95.95% (± 3.46%) | 96.41% (± 3.71%) | 95.59% (± 3.31%) |
| q=5 | 94.41% (± 3.68%) | 94.18% (± 3.64%) | 94.68% (± 3.58%) | 95.36% (± 3.75%) | 95.64% (± 3.34%) | 95.55% (± 3.58%) | 95.27% (± 4.01%) |
| q=6 | 94.91% (± 3.71%) | 94.55% (± 3.08%) | 94.59% (± 3.13%) | 95.05% (± 3.86%) | 95.77% (± 3.23%) | 94.55% (± 4.17%) | 95.05% (± 3.75%) |
| q=7 | 94.64% (± 4.03%) | 95.00% (± 3.78%) | 93.59% (± 3.97%) | 94.86% (± 3.67%) | 94.14% (± 3.87%) | 95.27% (± 3.74%) | 95.82% (± 3.38%) |

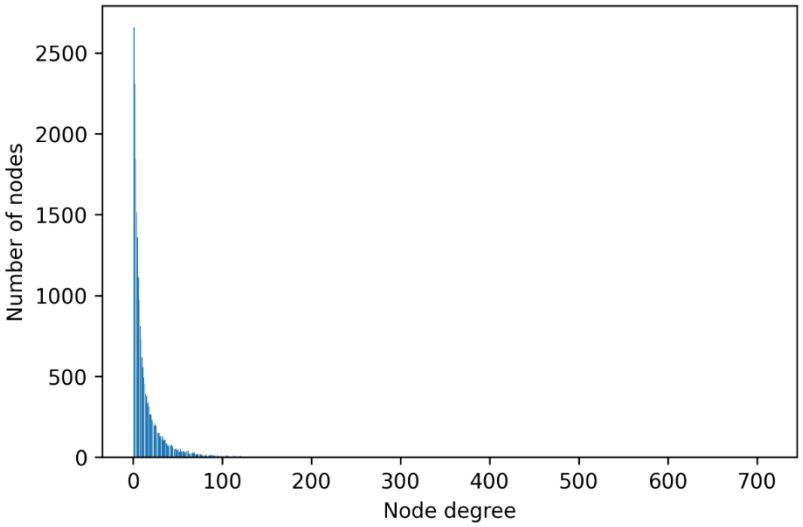
# Chapter 5: Including Node Features with Vanilla Neural Networks

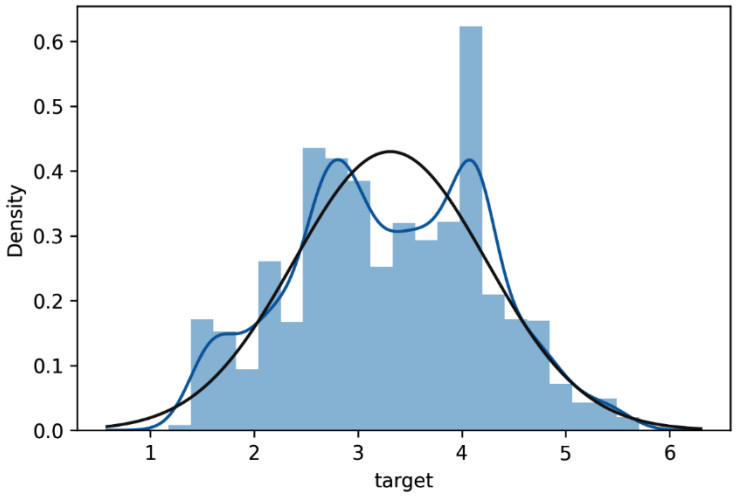


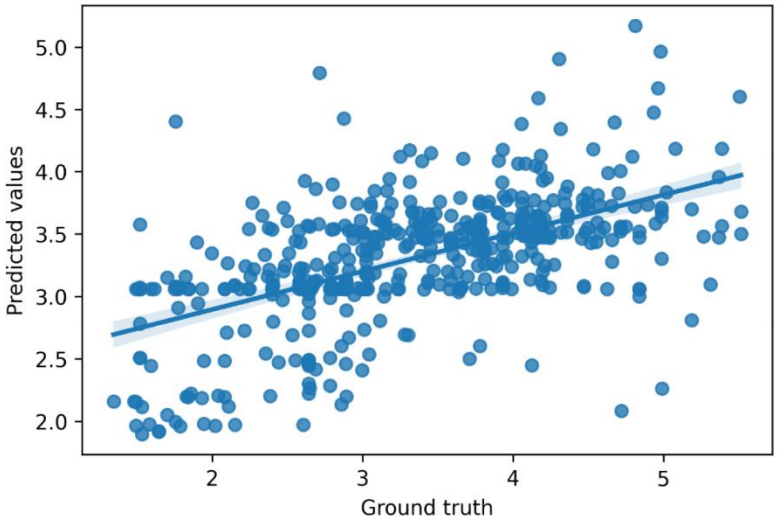
# Chapter 6: Introducing Graph Convolutional Networks

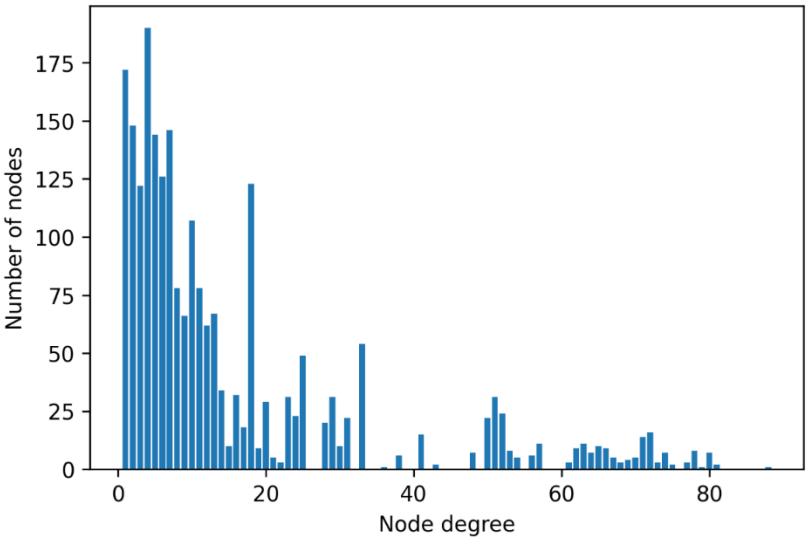


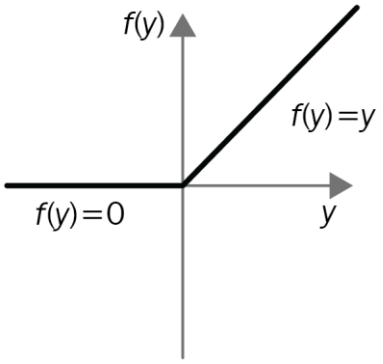




Density
target

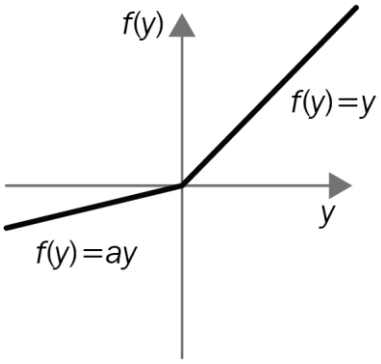

Predicted values
Ground truth


Number of nodes
Node degree

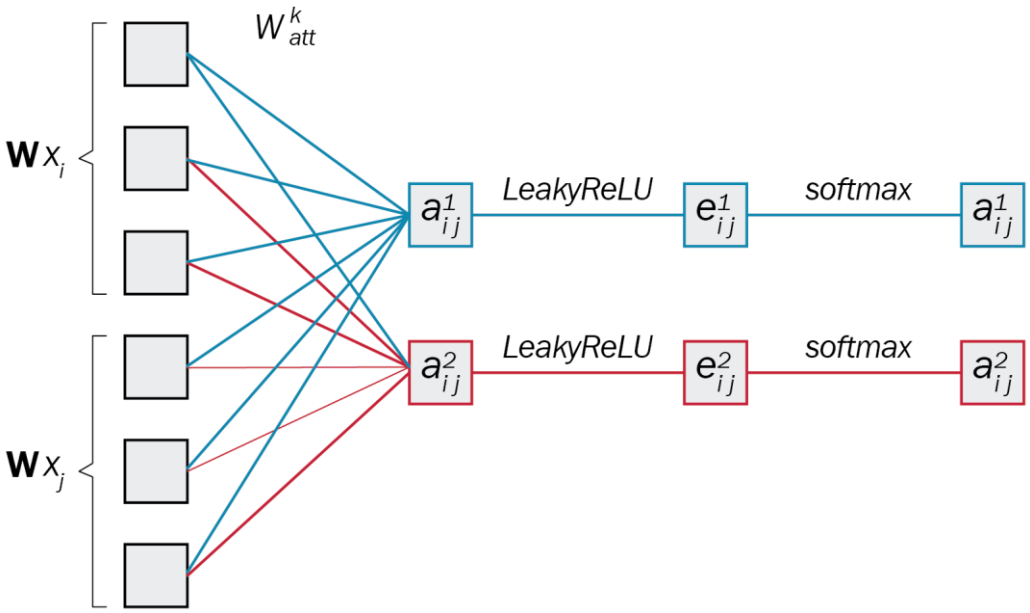
# Chapter 7: Graph Attention Networks

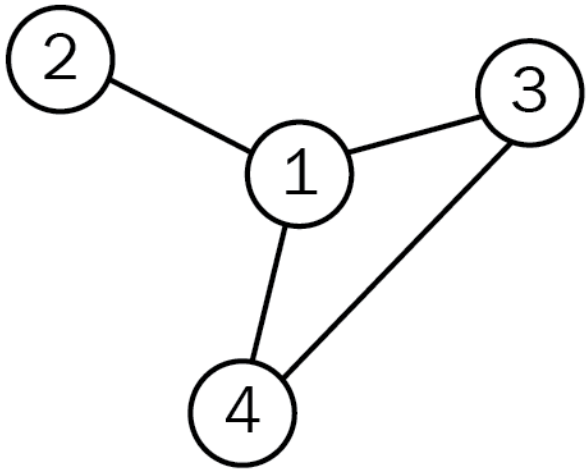


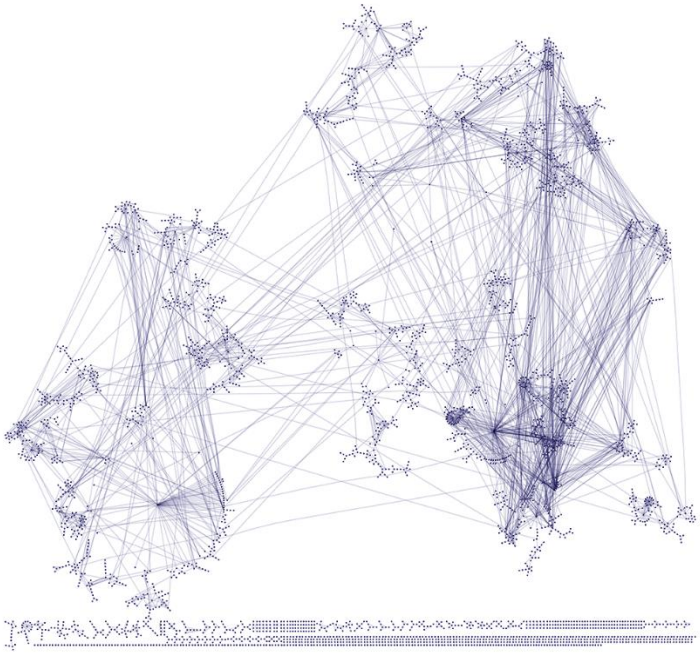

ReLU

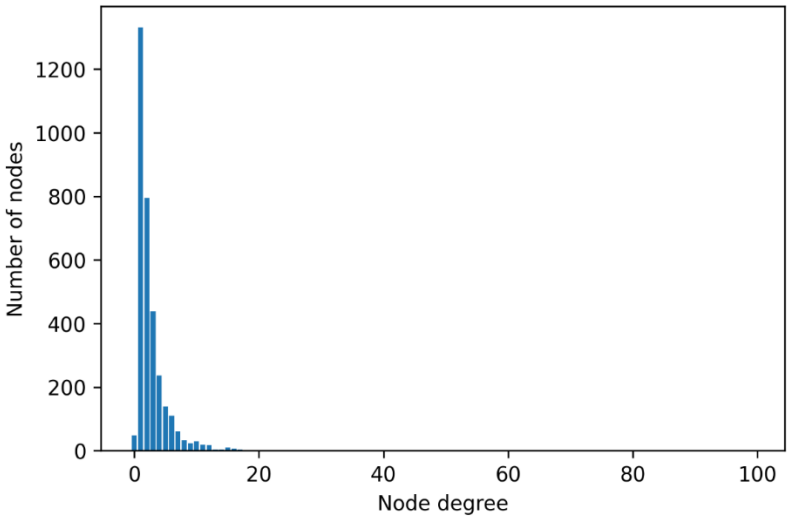


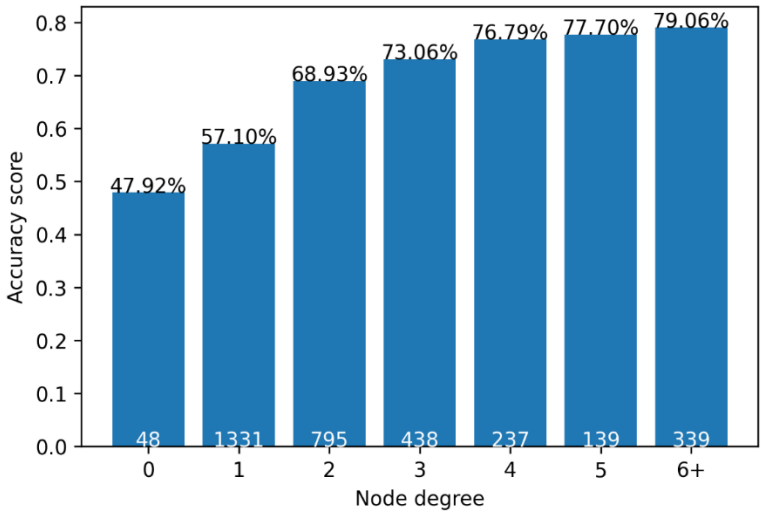

Leaky ReLU



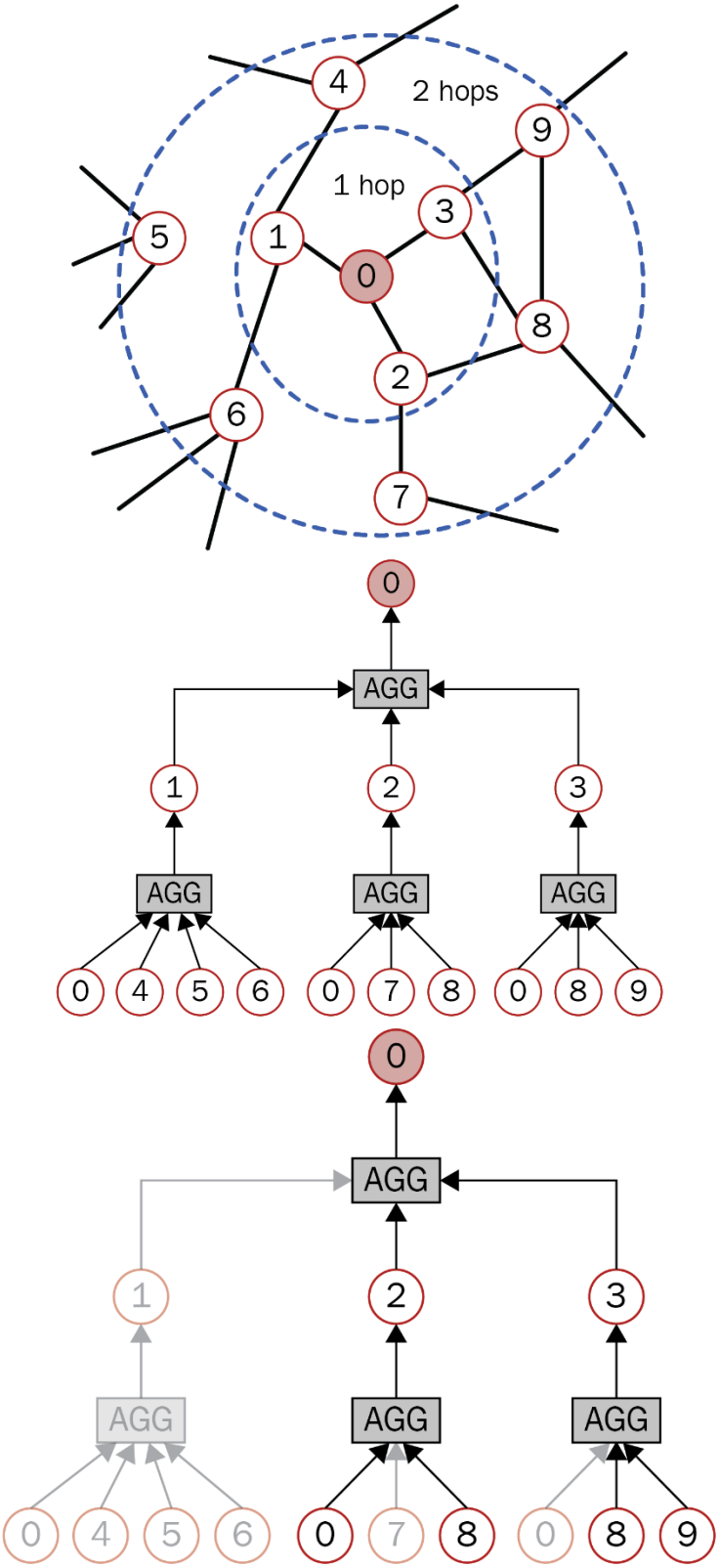



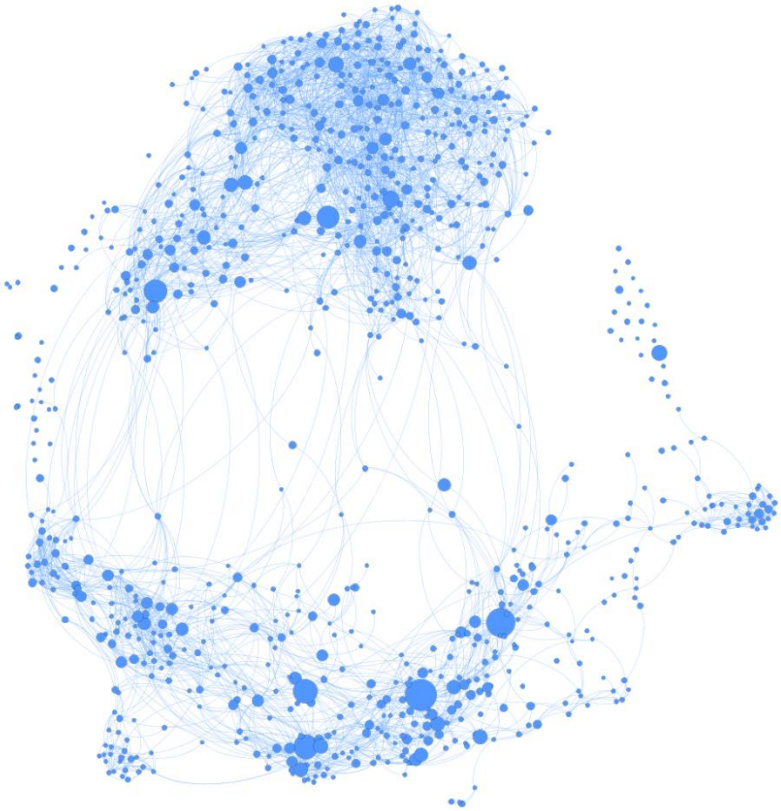



Number of nodes
1200
1000
800
600
400
200
0
0
20
40
60
80
100
Node degree

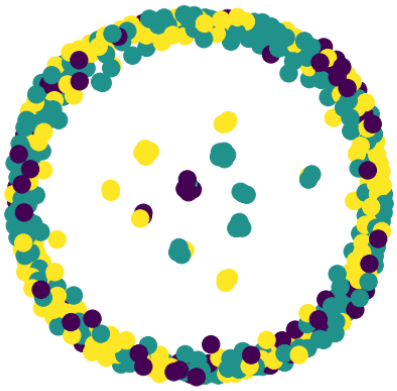

Accuracy score
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
47.92%
57.10%
68.93%
73.06%
76.79%
77.70%
79.06%
48
1331
795
438
237
139
339
0
1
2
3
4
5
6+
Node degree

# Chapter 8: Scaling up Graph Neural Networks with GraphSAGE

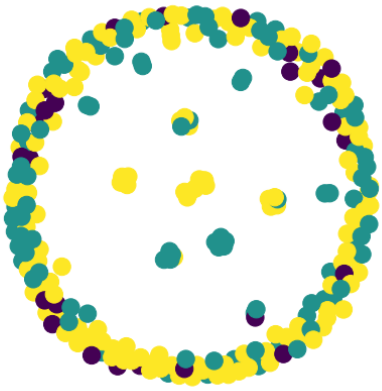




Subgraph 0

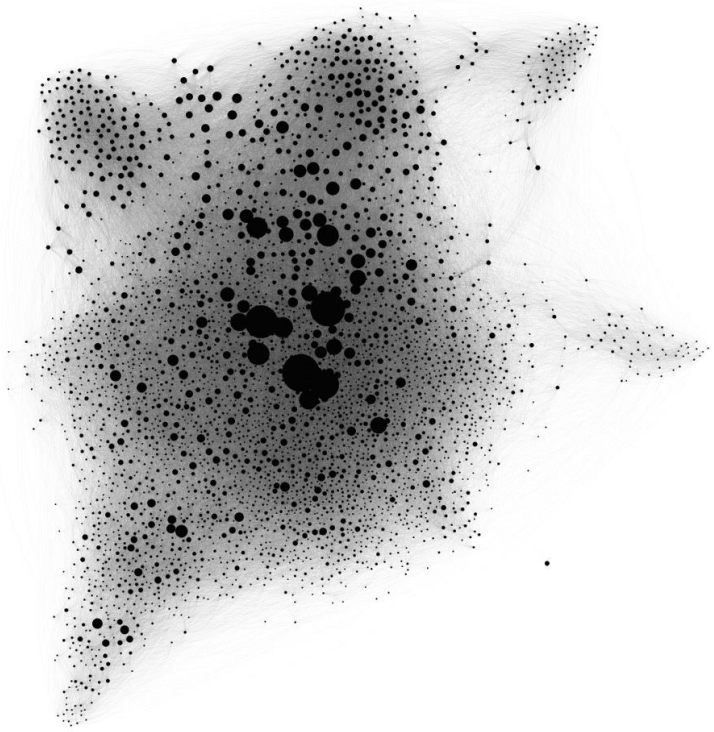


Subgraph 1

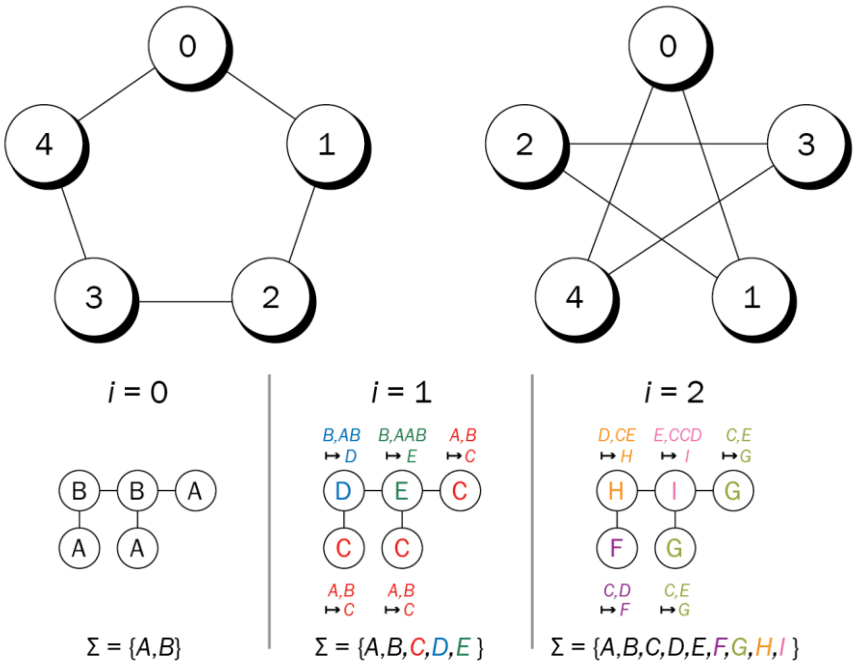


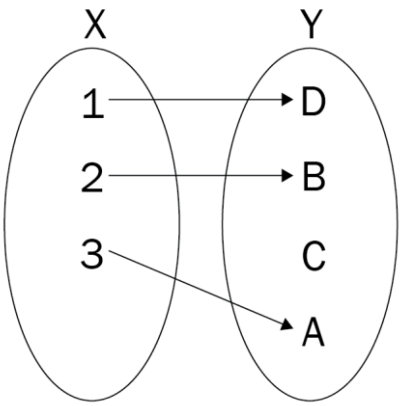
Subgraph 2

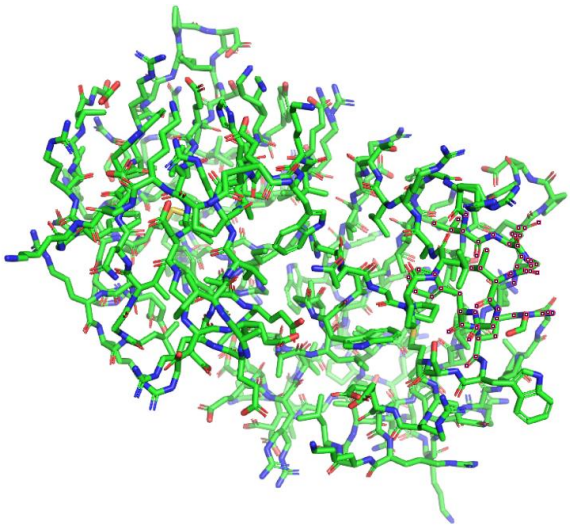
Subgraph 3

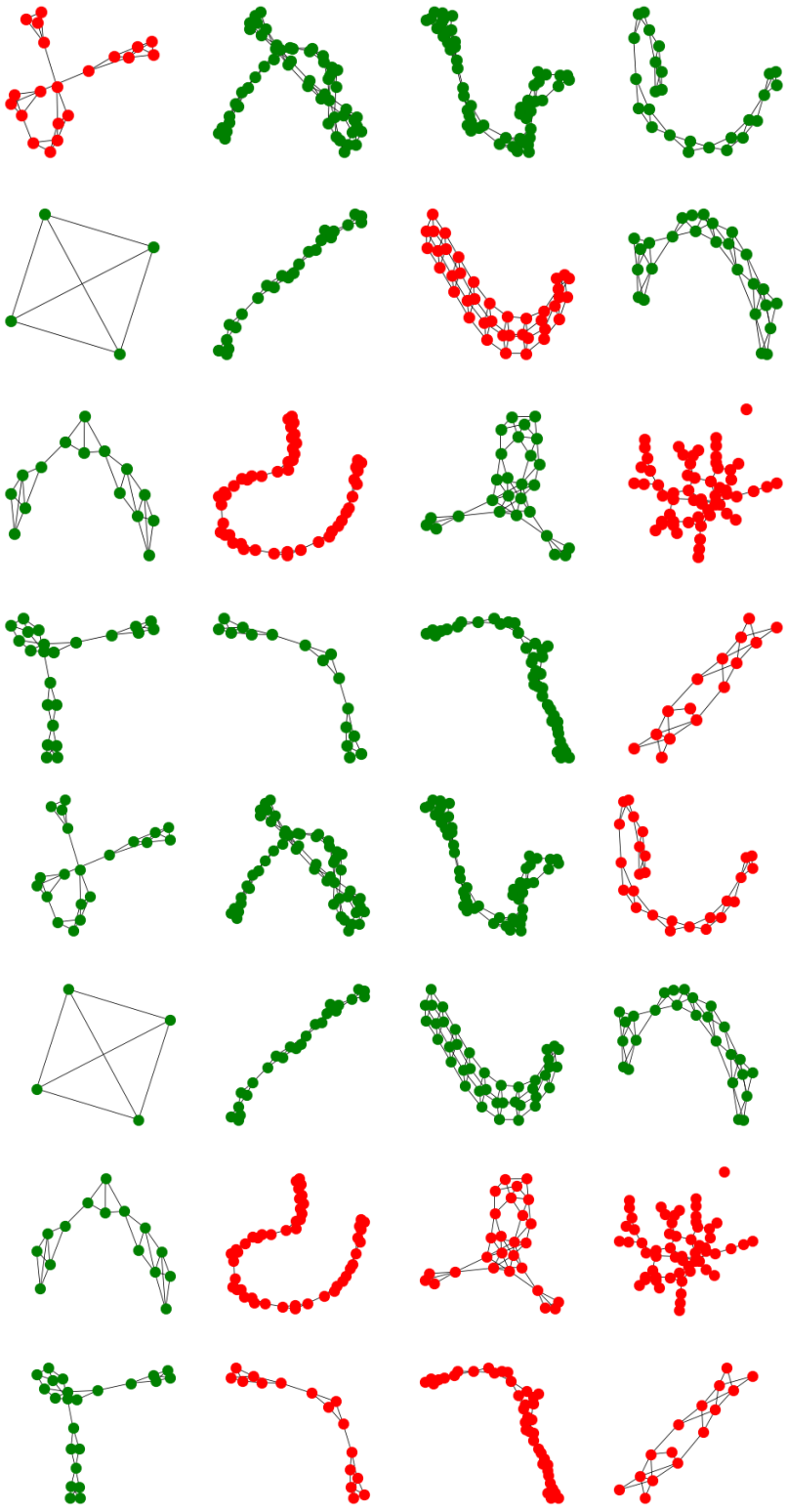


# Chapter 9: Defining Expressiveness for Graph Classification

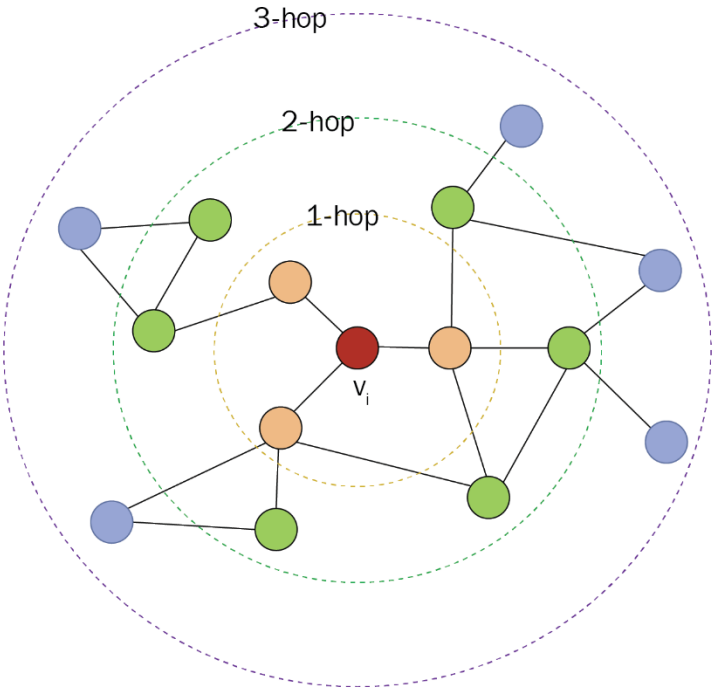


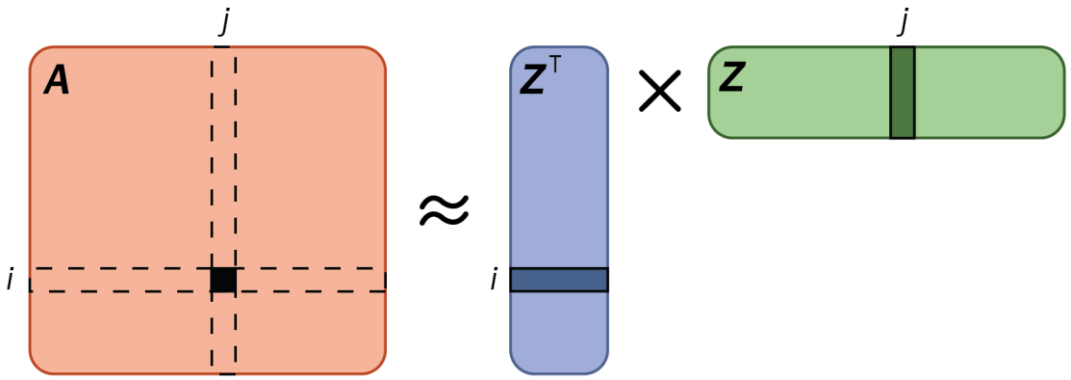



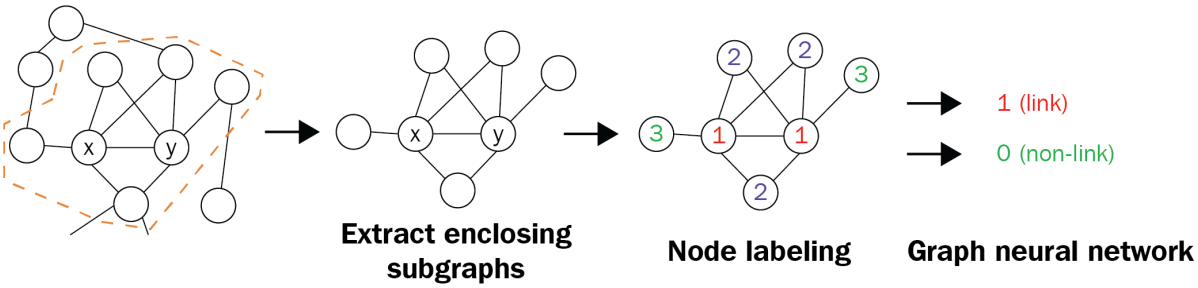





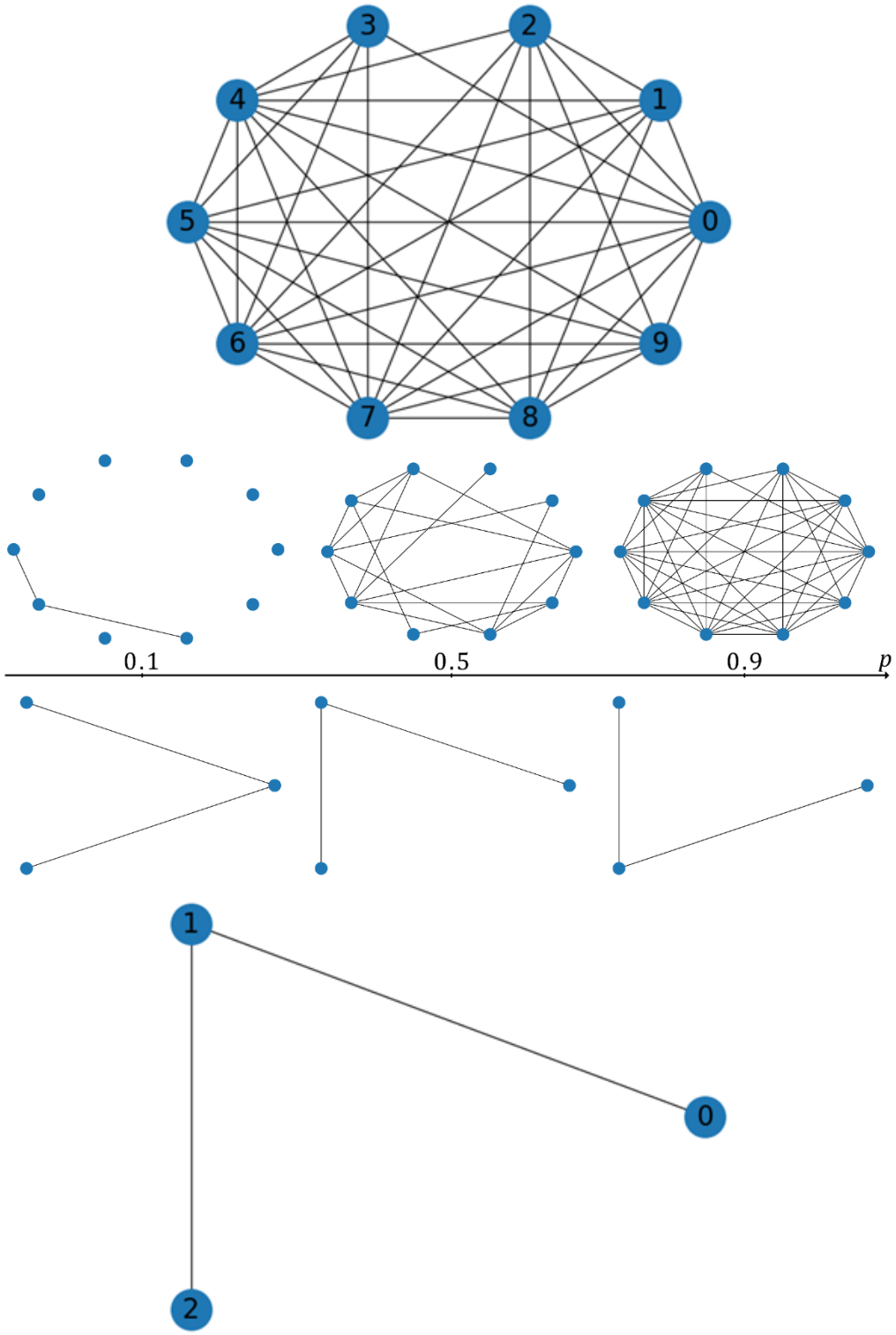
# Chapter 10: Predicting Links with Graph Neural Networks

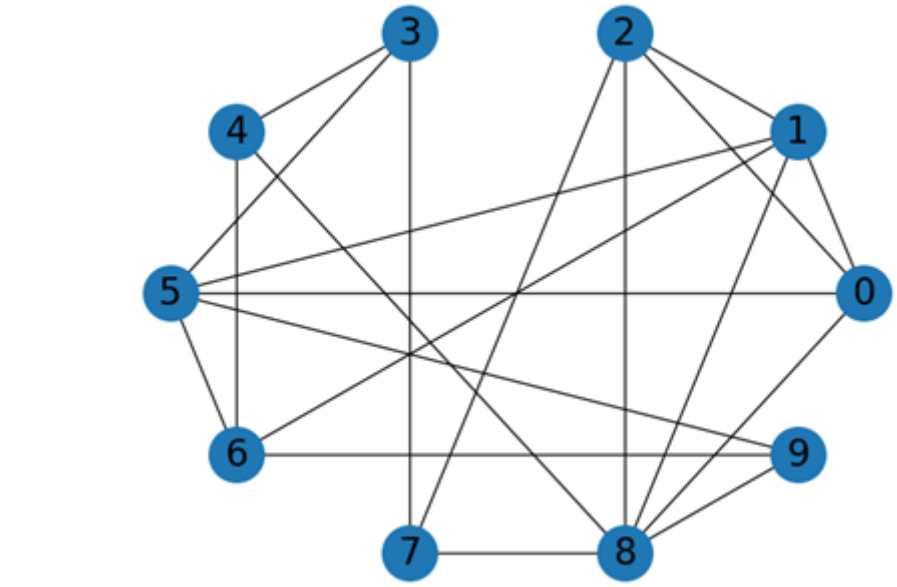


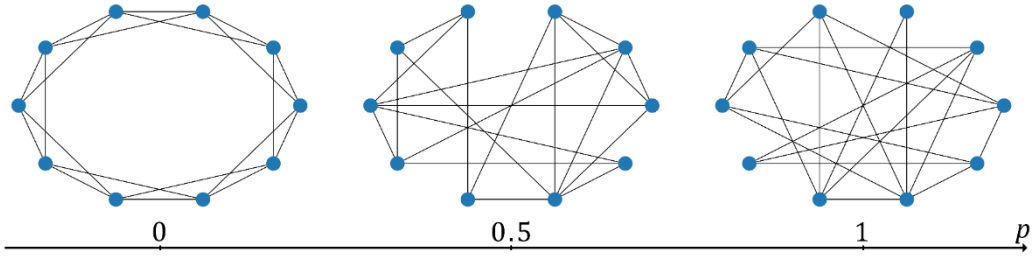



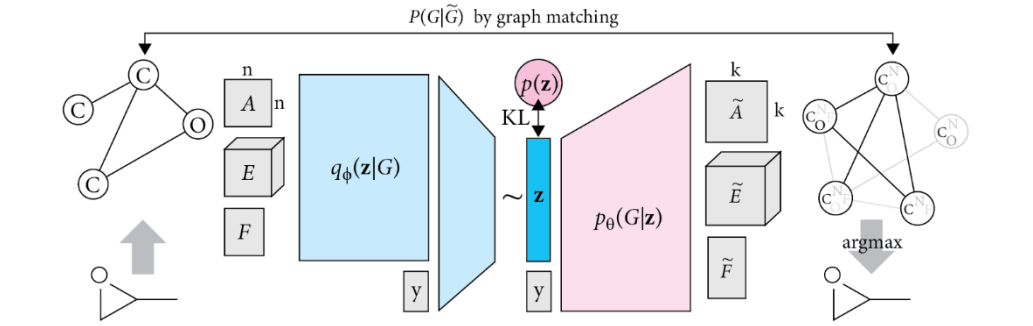



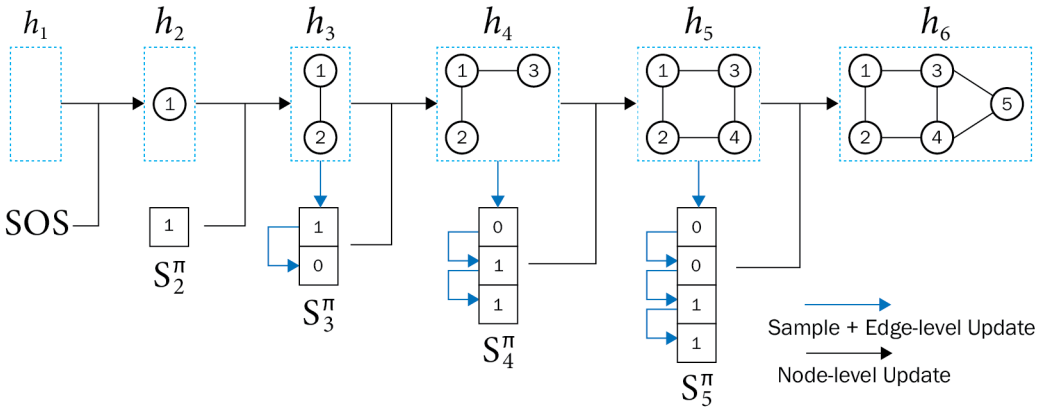
# Chapter 11: Generating Graphs Using Graph Neural Networks

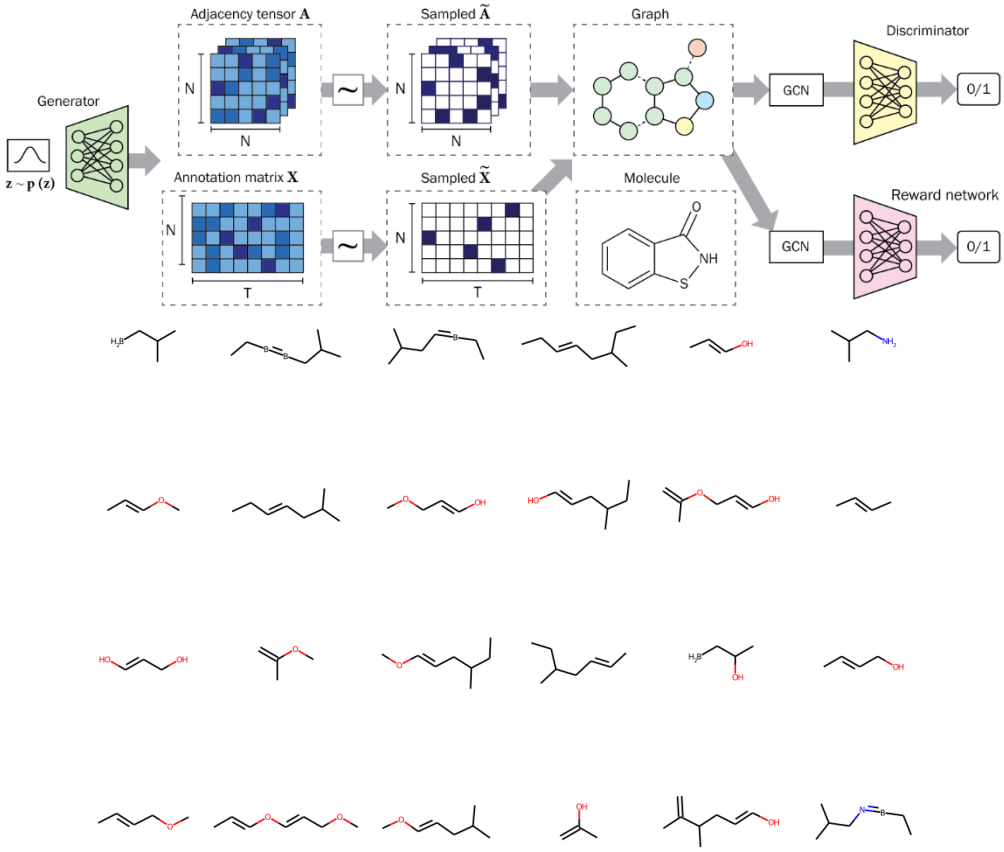



3
2
4
1
5
0
6
9
7
8

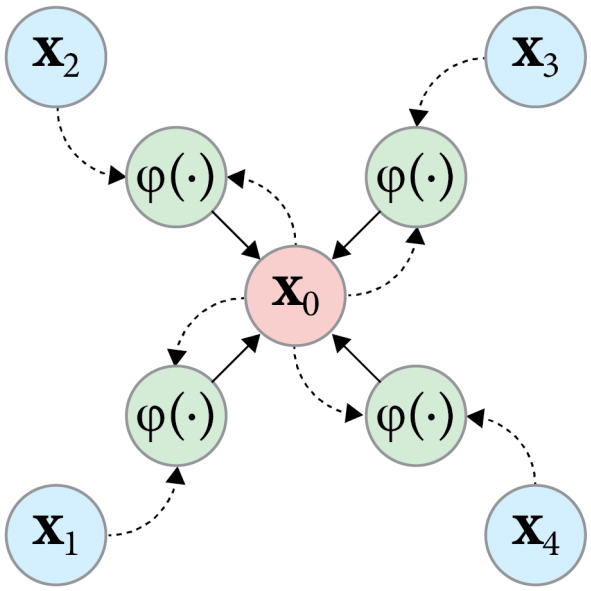

0
0.5
1
p

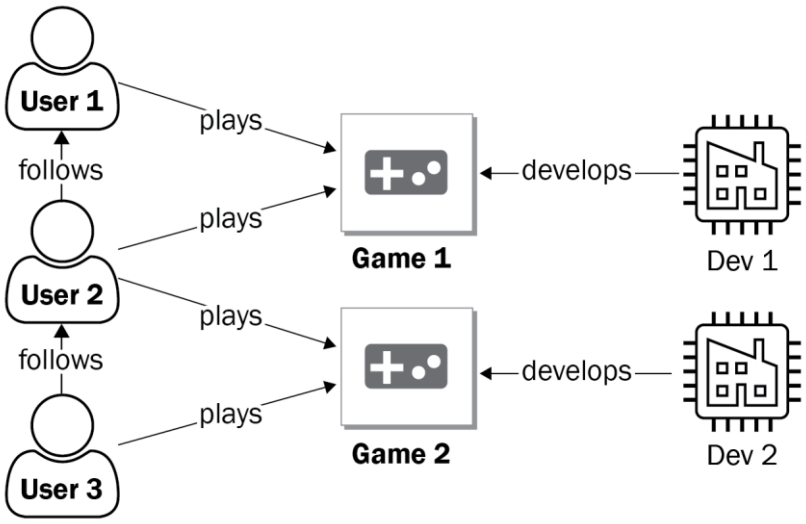

P(G|G̃) by graph matching
n
A
n
E
F
qϕ(z|G)
y
p(z)
KL
z
y
pθ(G|z)
k
Ã
k
Ẽ
F̃
argmax
1
1
2
1
2
1
3
2
1
3
2

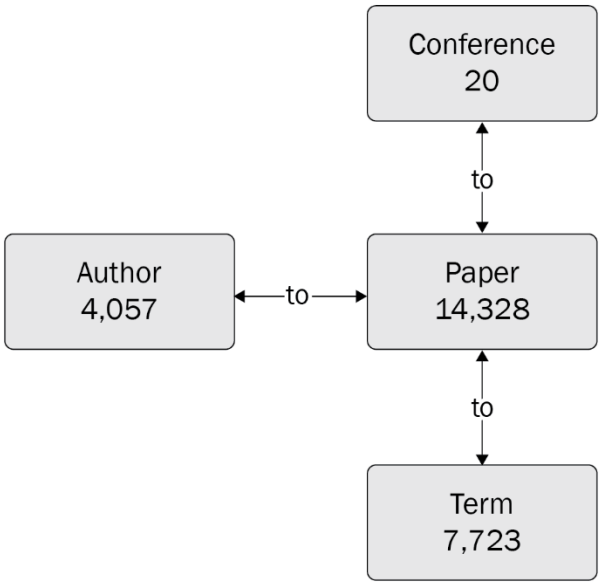

h1
h2
h3
h4
h5
h6
SOS
1
S2π
1
0
S3π
0
1
1
S4π
0
0
1
1
S5π
Sample + Edge-level Update
Node-level Update

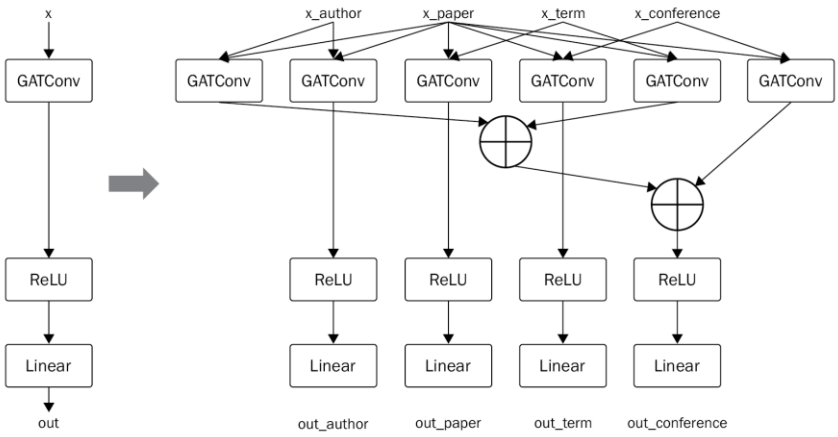

Generator
z ~ p (z)
Adjacency tensor A
N
N
Annotation matrix X
N
T
Sampled Ã
N
N
Sampled X̃
N
T
Graph
Molecule
GCN
GCN
Discriminator
Reward network
0/1
0/1

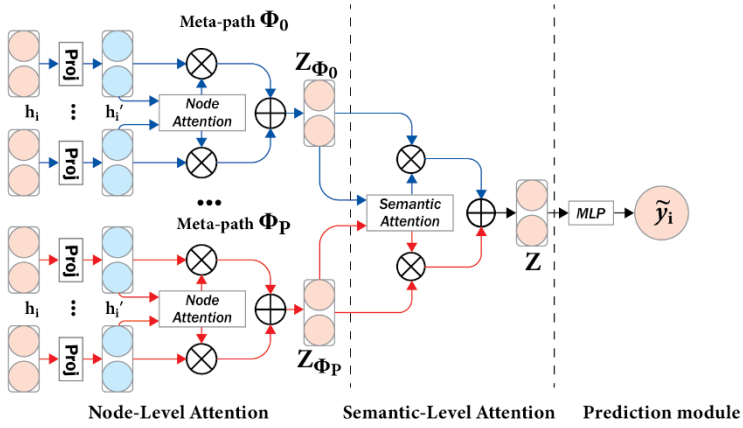
# Chapter 12: Learning from Heterogeneous Graphs



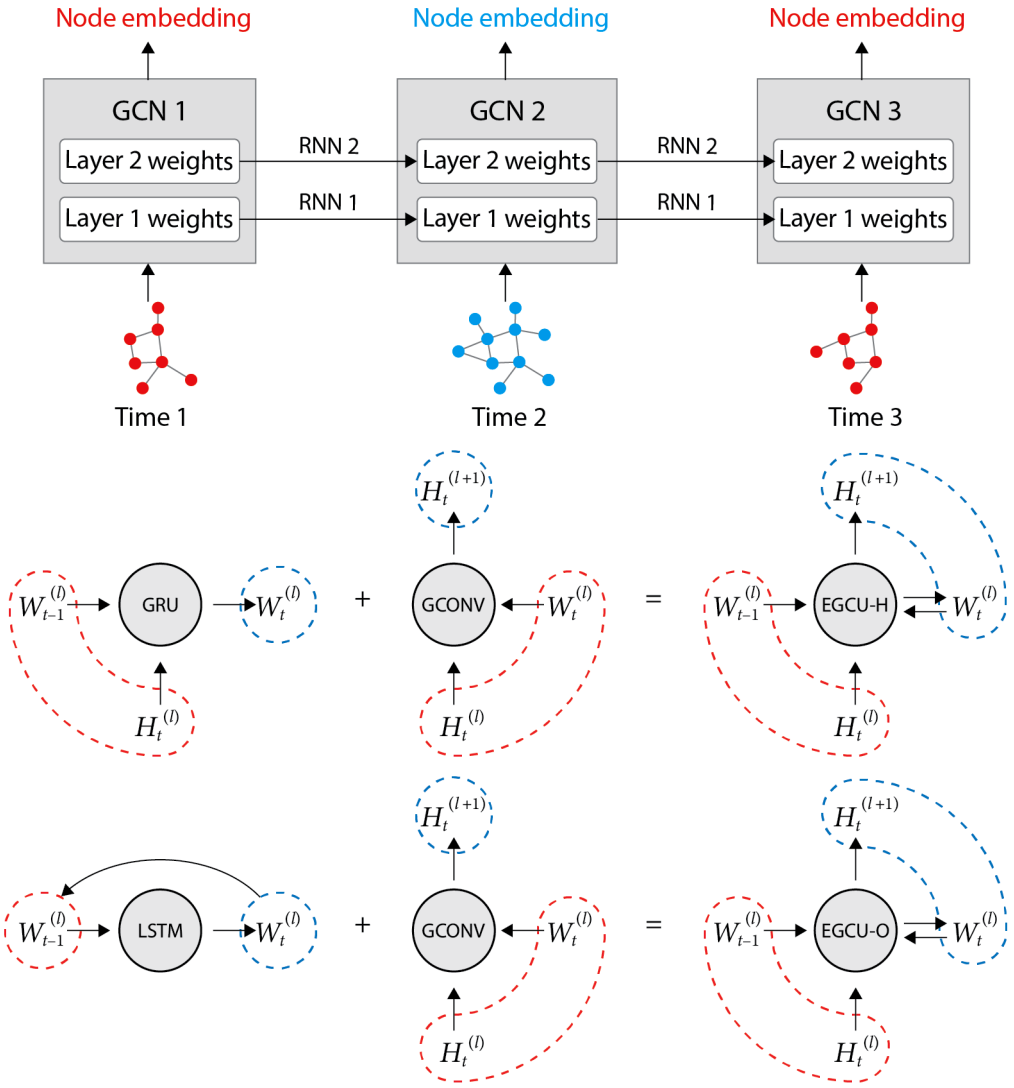



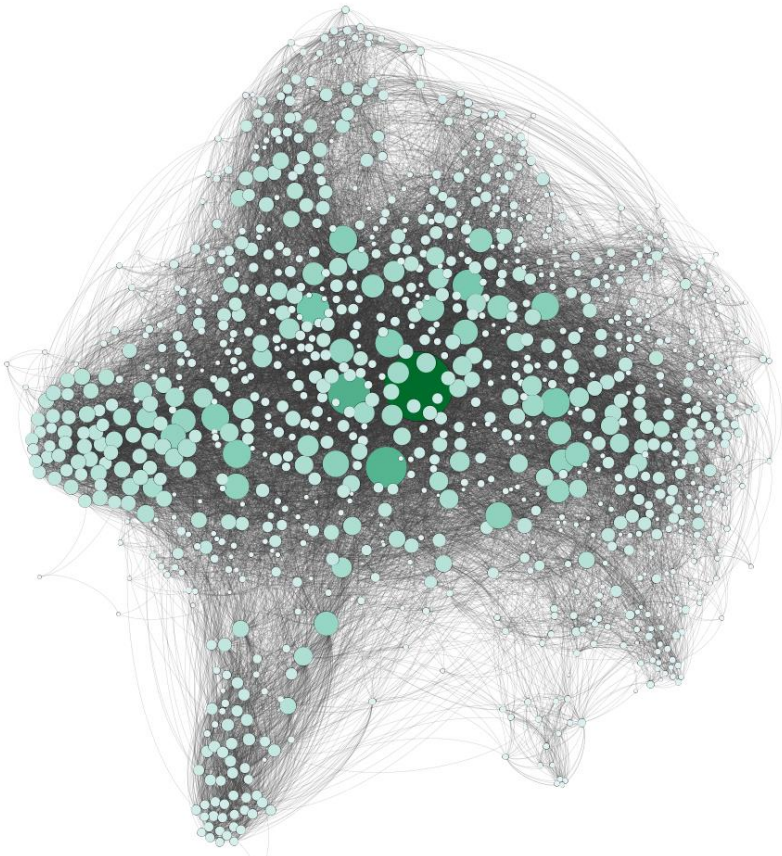



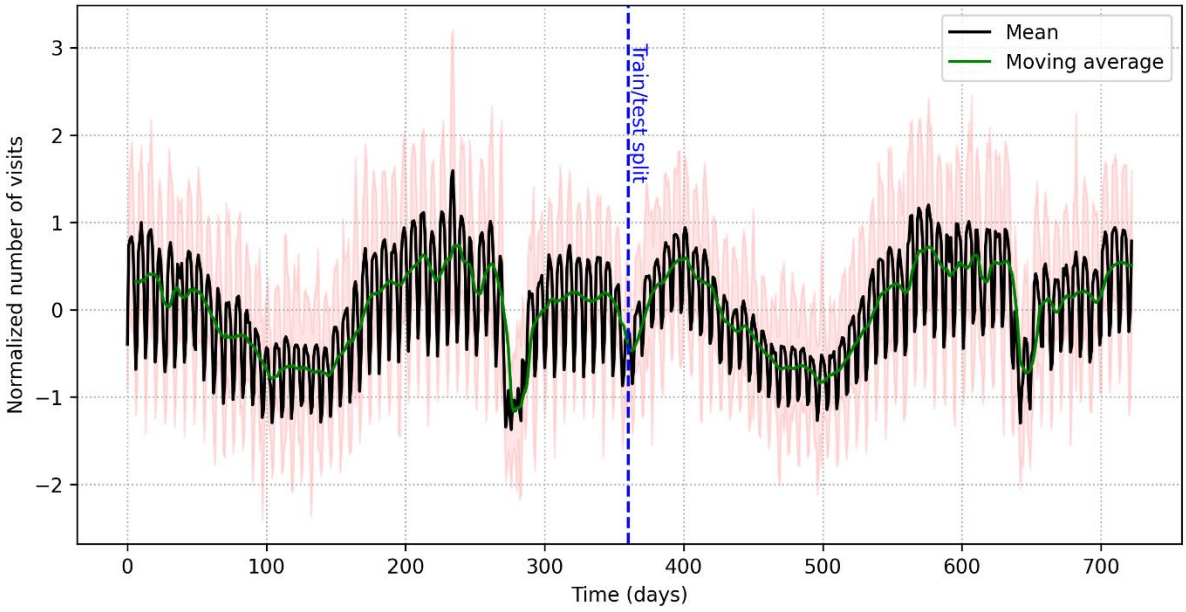

x
GATConv
ReLU
Linear
out
x_author
x_paper
x_term
x_conference
GATConv
GATConv
GATConv
GATConv
GATConv
GATConv
ReLU
ReLU
ReLU
ReLU
Linear
Linear
Linear
Linear
out_author
out_paper
out_term
out_conference

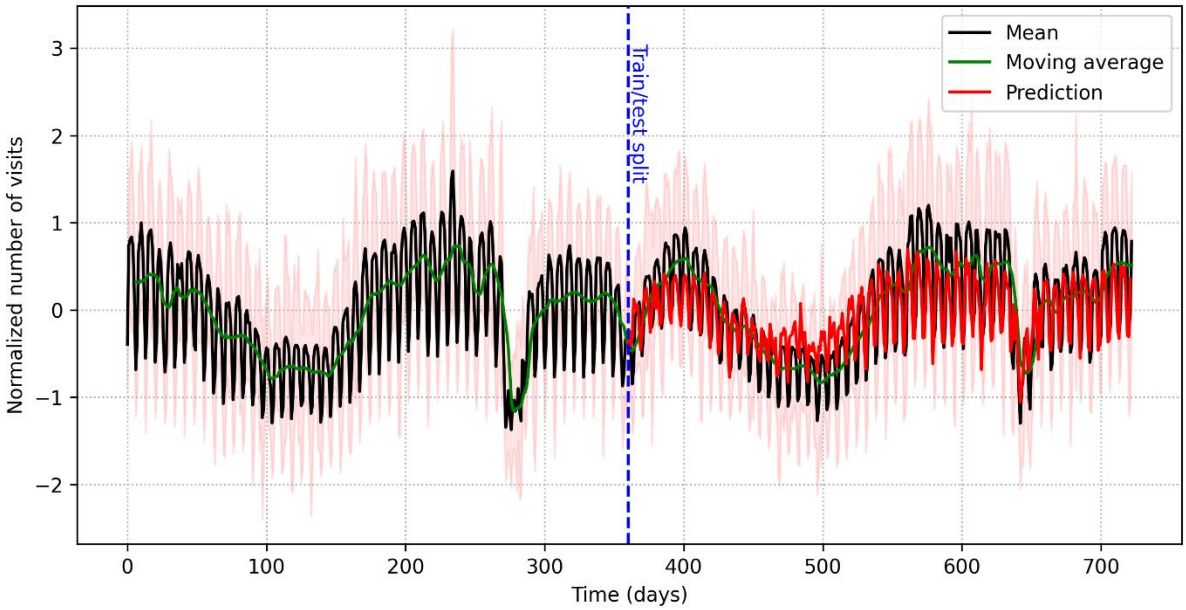

Meta-path Φ0
Proj
Proj
hi
hi'
Node Attention
ZΦ0
Meta-path ΦP
Proj
Proj
hi
hi'
Node Attention
ZΦP
Semantic Attention
Z
MLP
ỹi
Node-Level Attention
Semantic-Level Attention
Prediction module

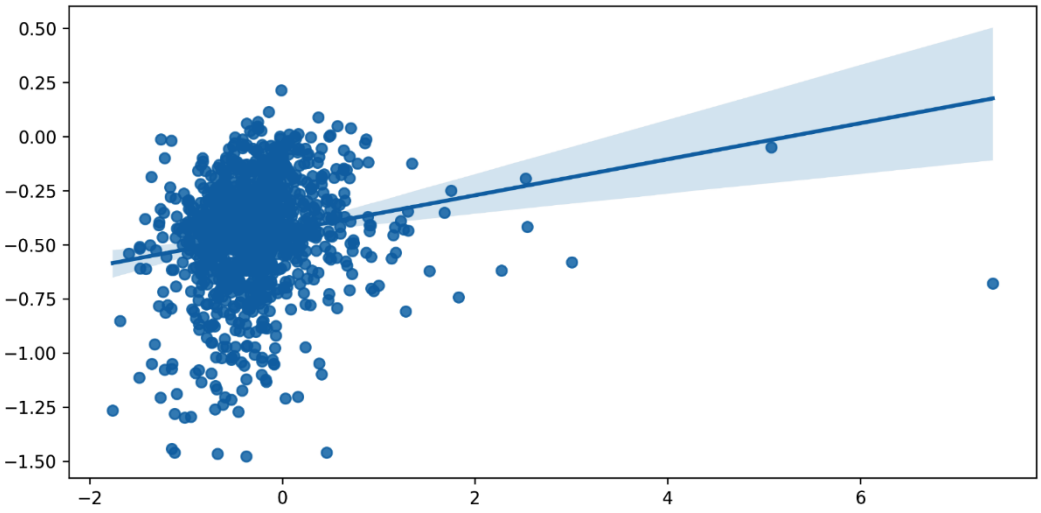
# Chapter 13: Temporal Graph Neural Networks

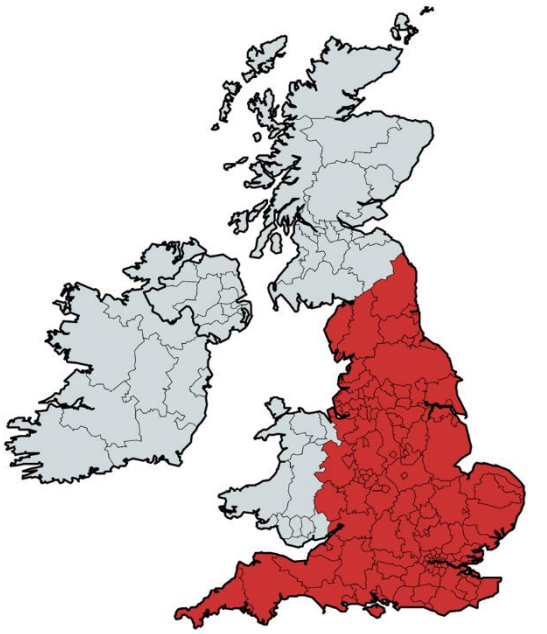


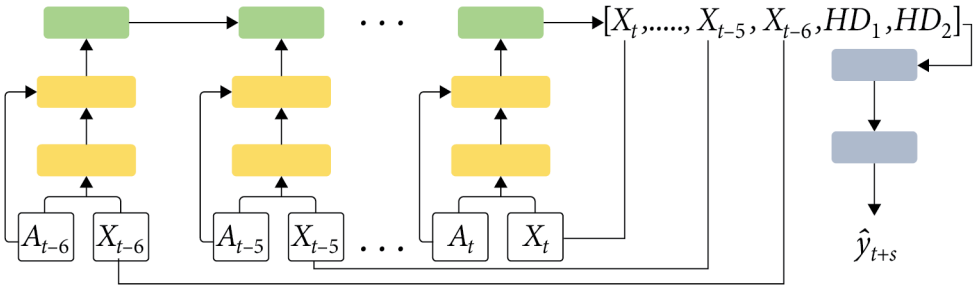



Mean
Moving average
Train/test split
Normalized number of visits
Time (days)
3
2
1
0
−1
−2
0
100
200
300
400
500
600
700

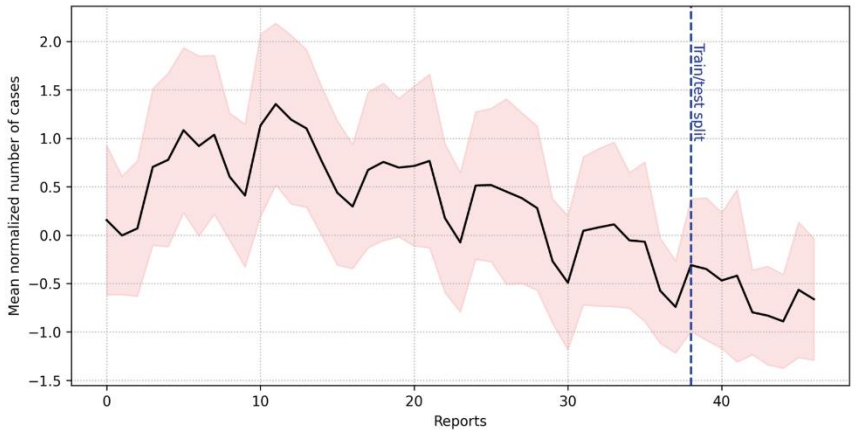

Mean
Moving average
Prediction
Train/test split
Normalized number of visits
Time (days)

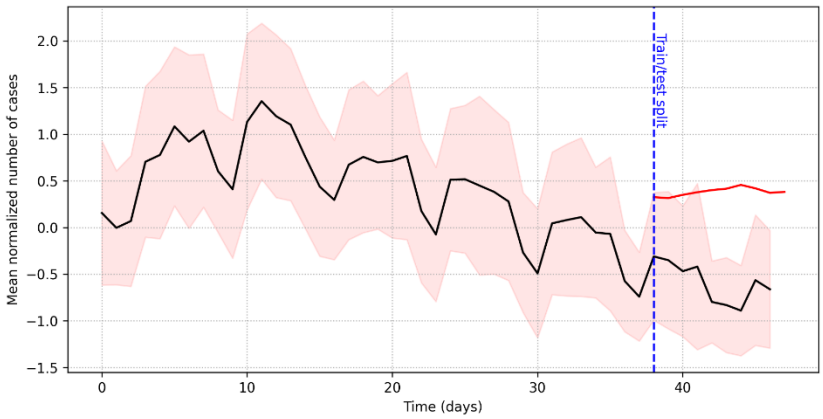


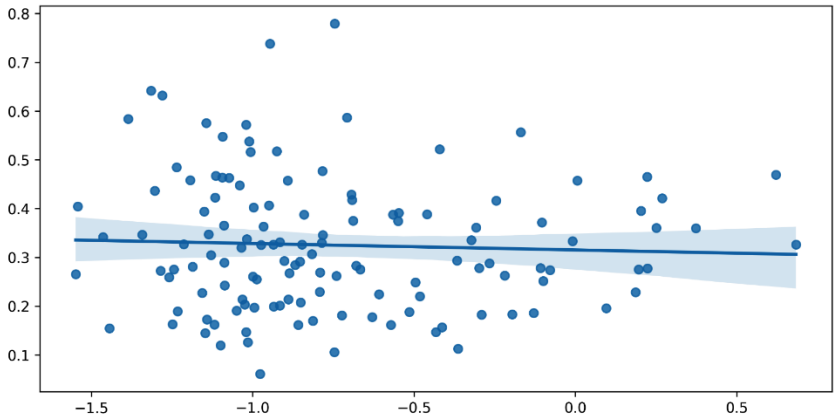




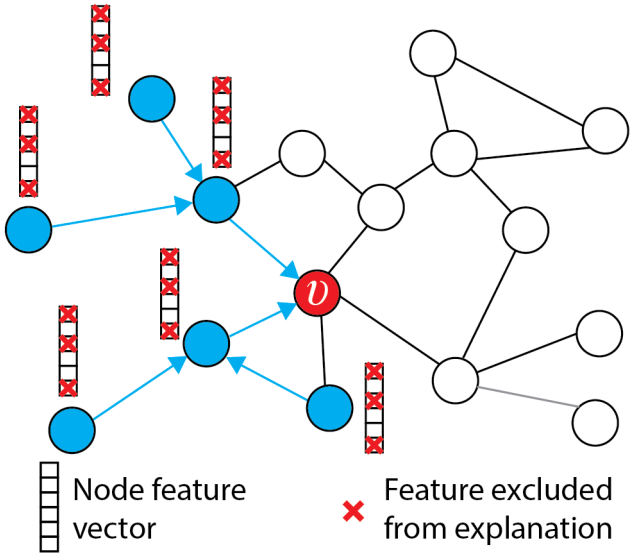

*MPNN + ReLU + BN + DR* *2 × LSTM* *Lin + ReLU*

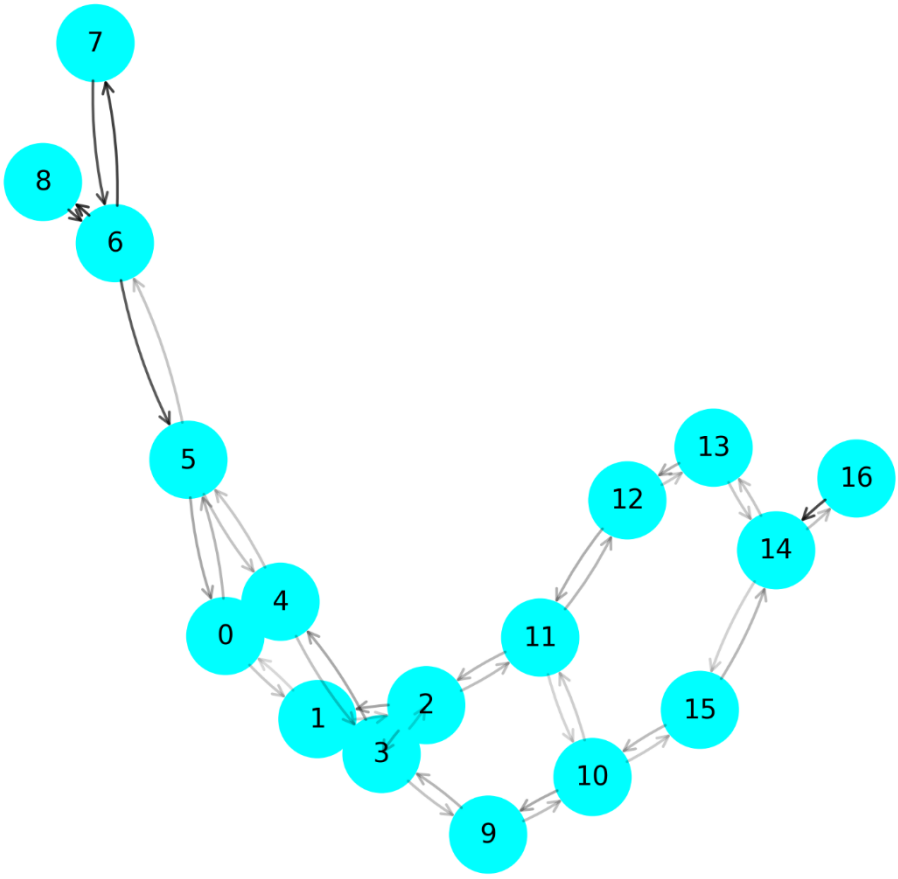


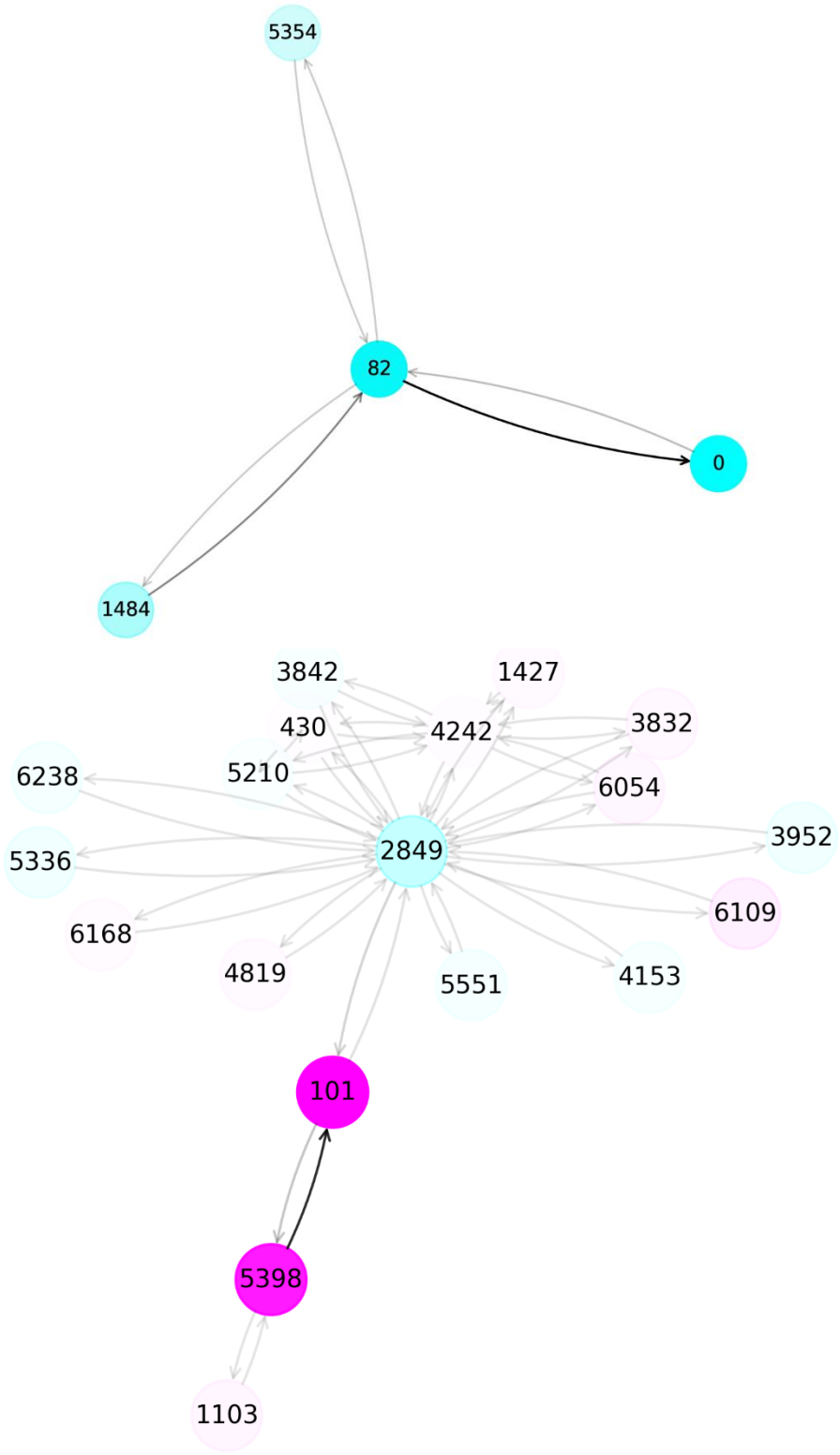



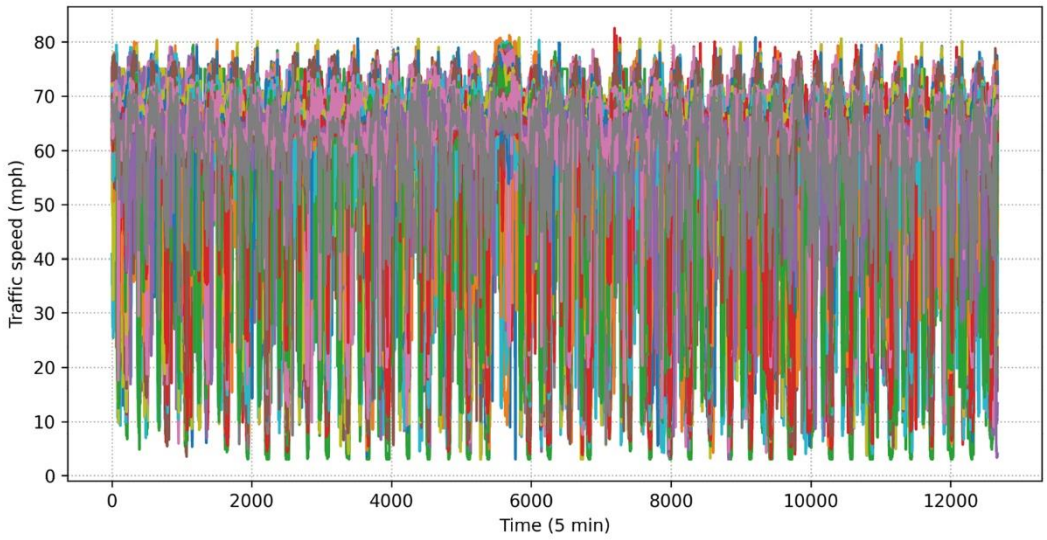
# Chapter 14: Explaining Graph Neural Networks

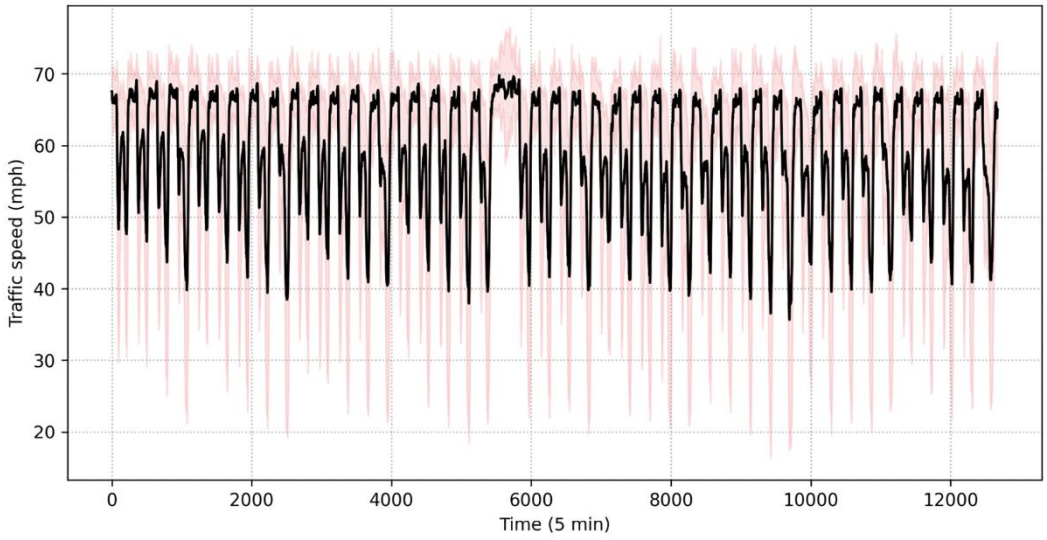


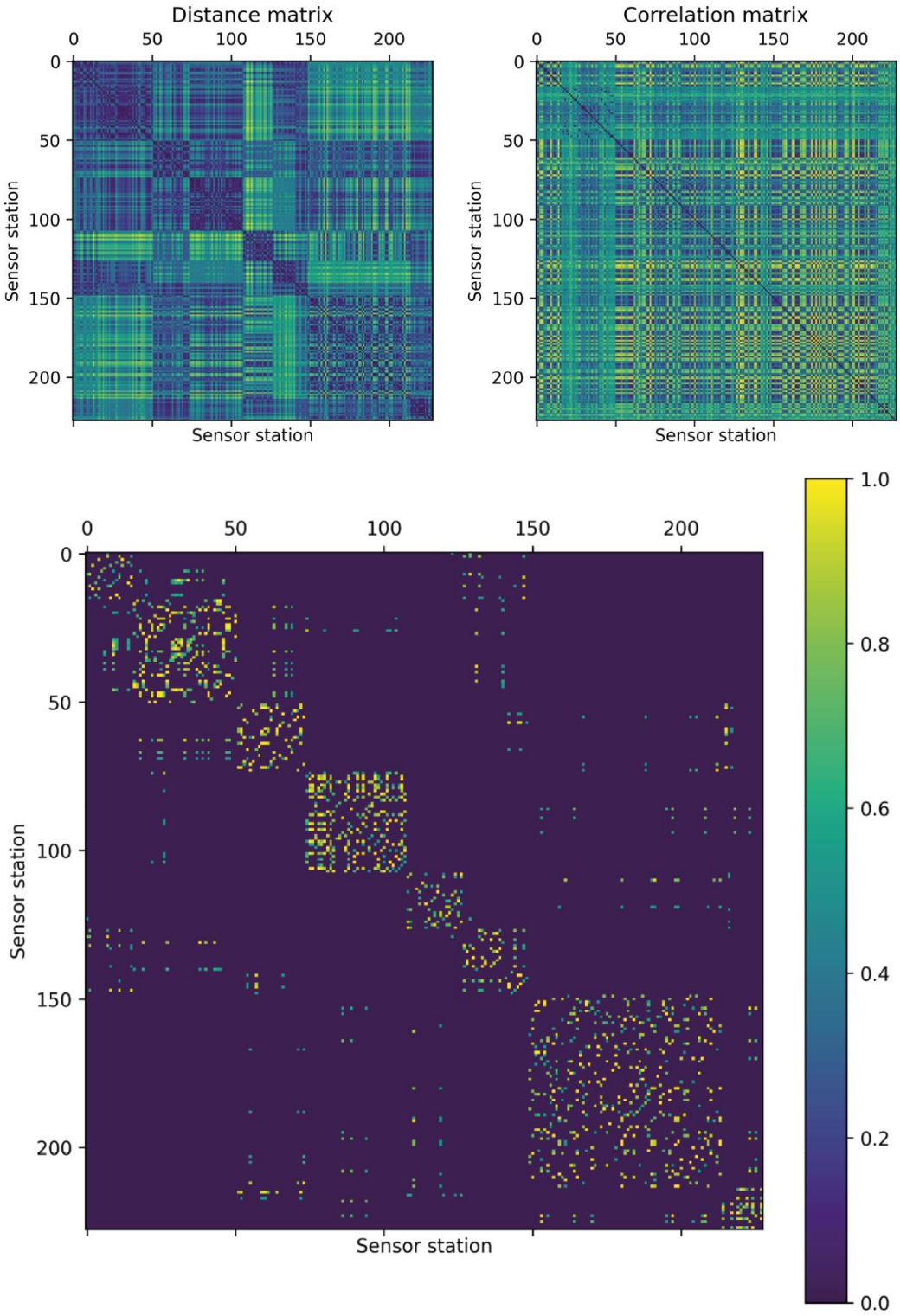



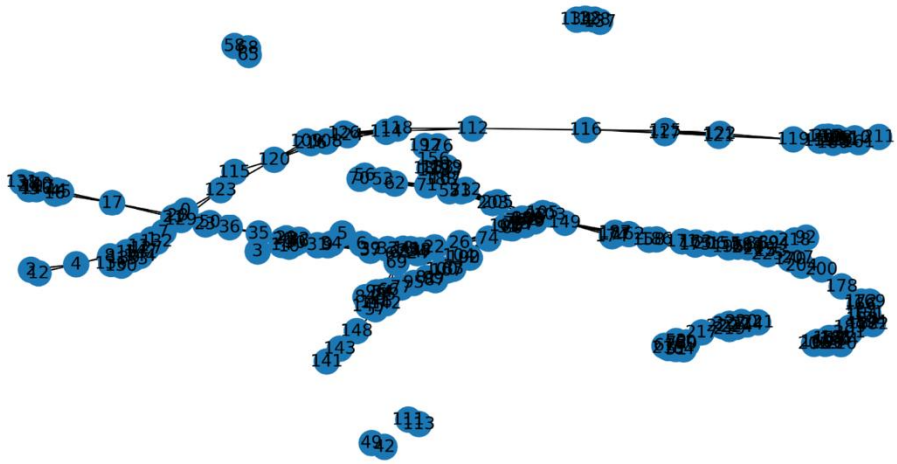

5354
82
0
1484
3842
1427
430
4242
3832
6238
5210
6054
3952
5336
2849
6109
6168
4819
5551
4153
101
5398
1103

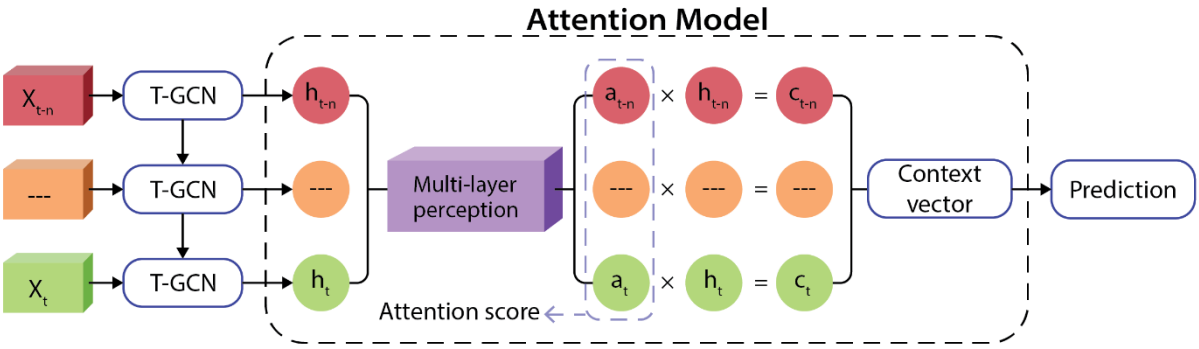
# Chapter 15: Forecasting Traffic Using A3T-GCN

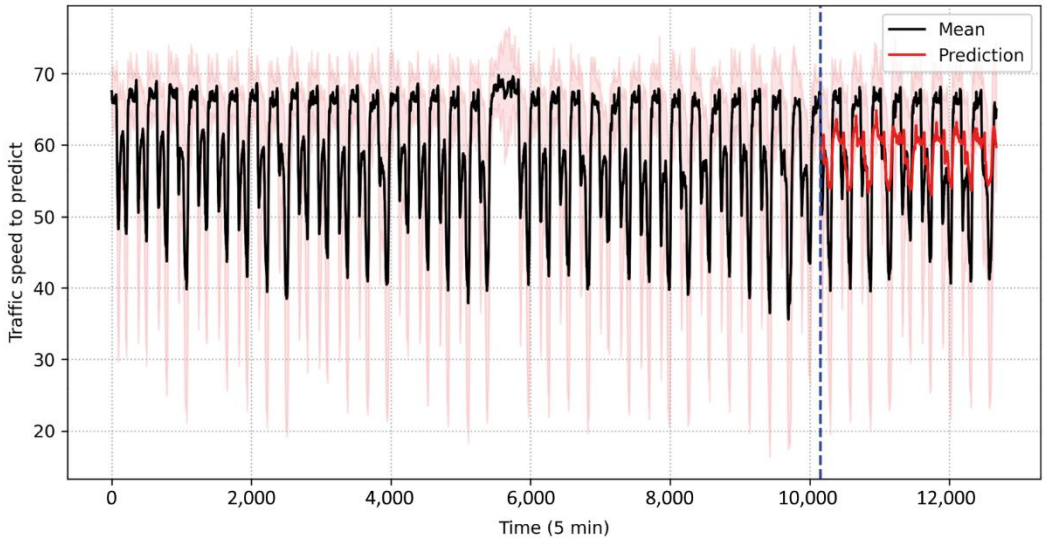


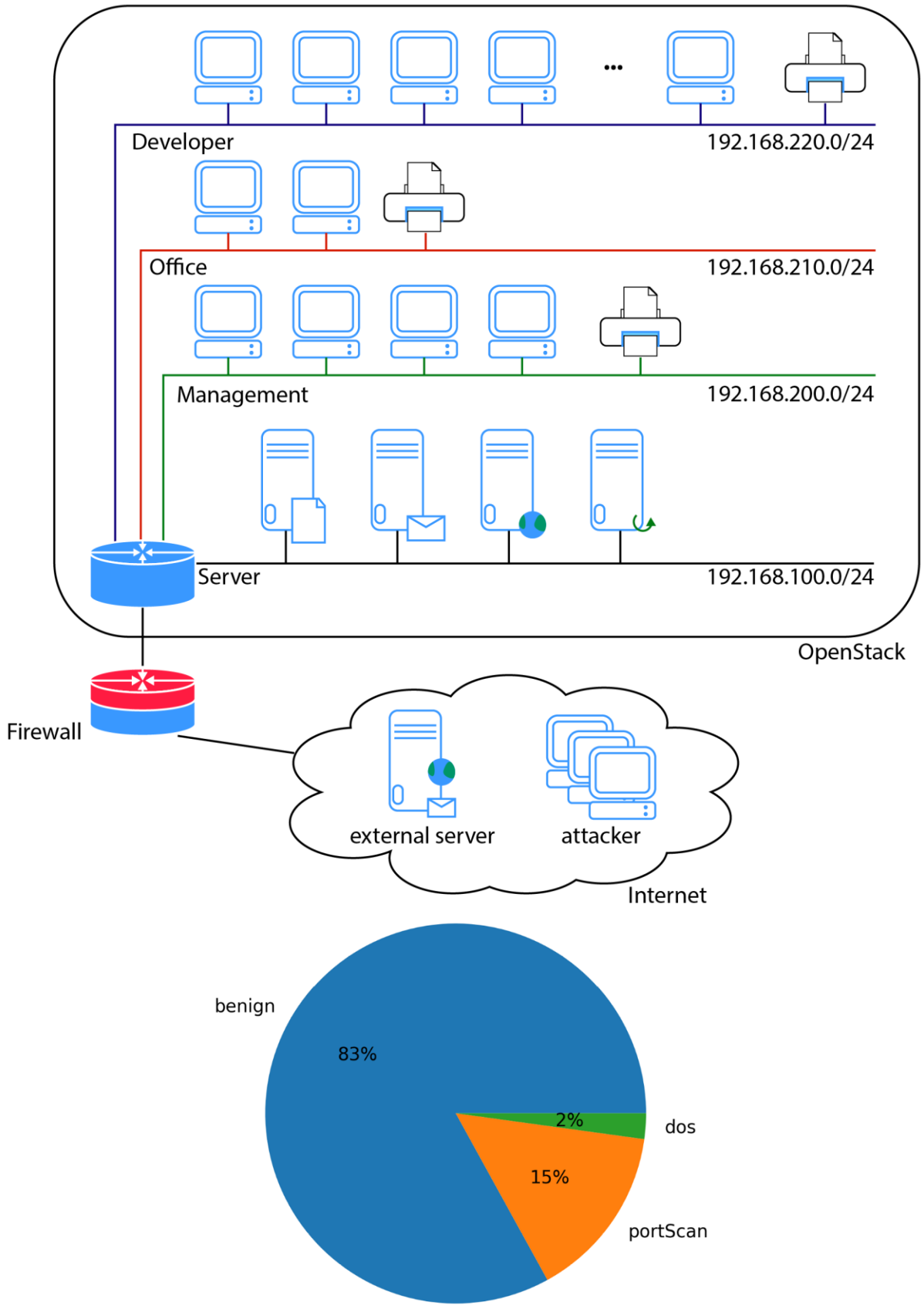

Distance matrix
Correlation matrix
Sensor station
Sensor station
0
50
100
150
200
1.0
0.8
0.6
0.4
0.2
0.0

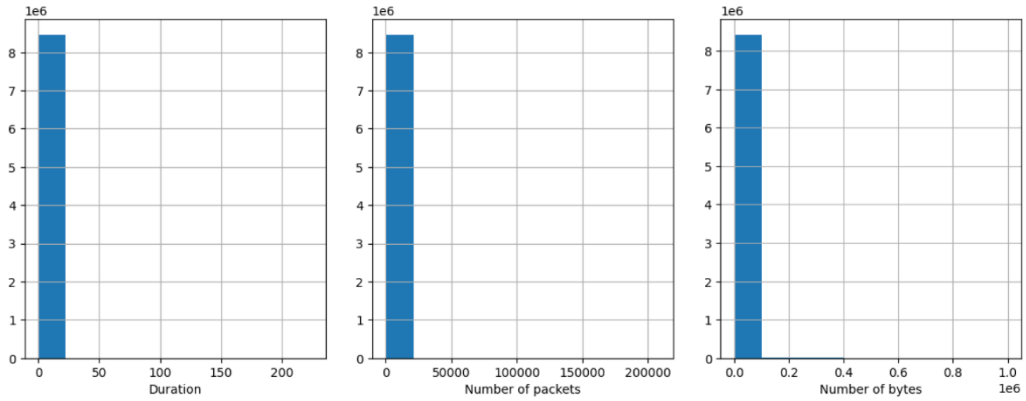


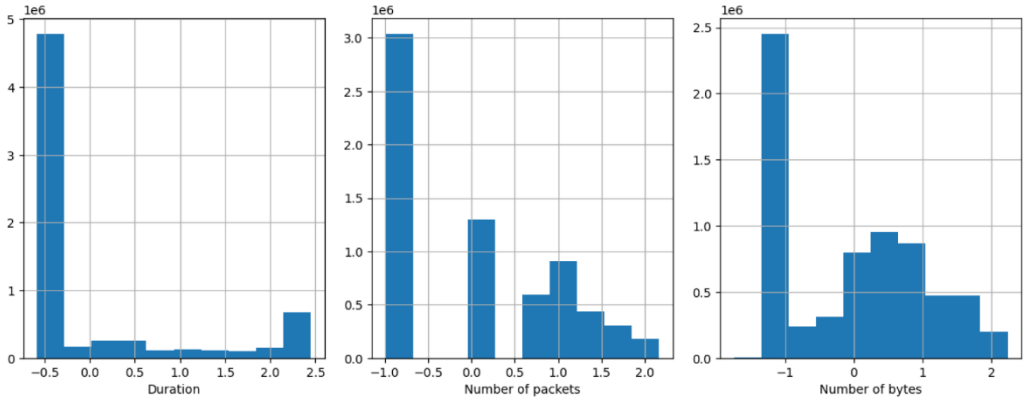

Attention Model
X t-n
T-GCN
h t-n
---
T-GCN
---
Multi-layer perception
a t-n
h t-n
c t-n
---
---
---
Context vector
Prediction
X t
T-GCN
h t
a t
h t
c t
Attention score

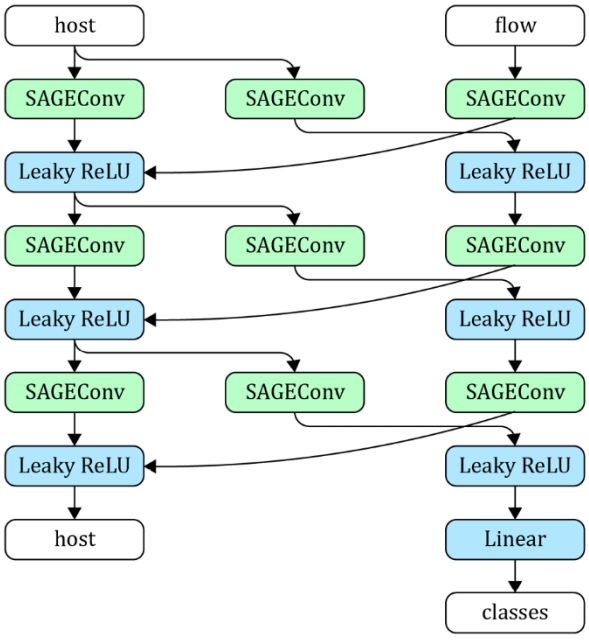

Mean
Prediction
Traffic speed to predict
70
60
50
40
30
20
0
2,000
4,000
6,000
8,000
10,000
12,000
Time (5 min)

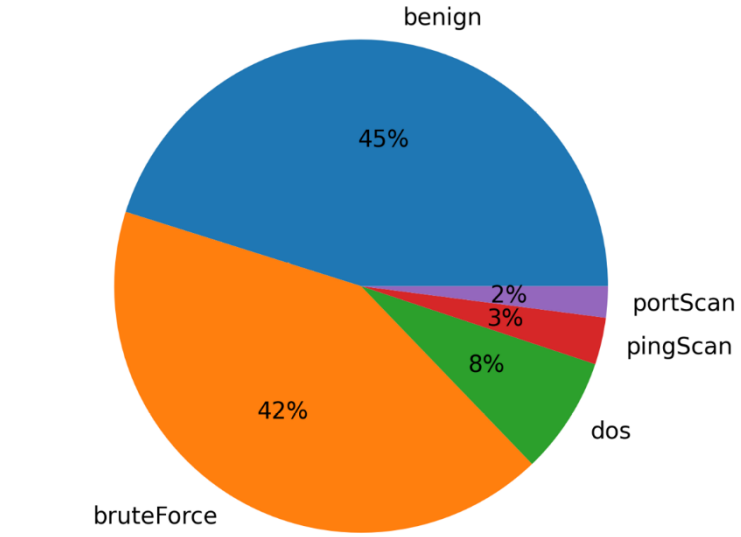
# Chapter 16: Detecting Anomalies Using Heterogeneous GNNs

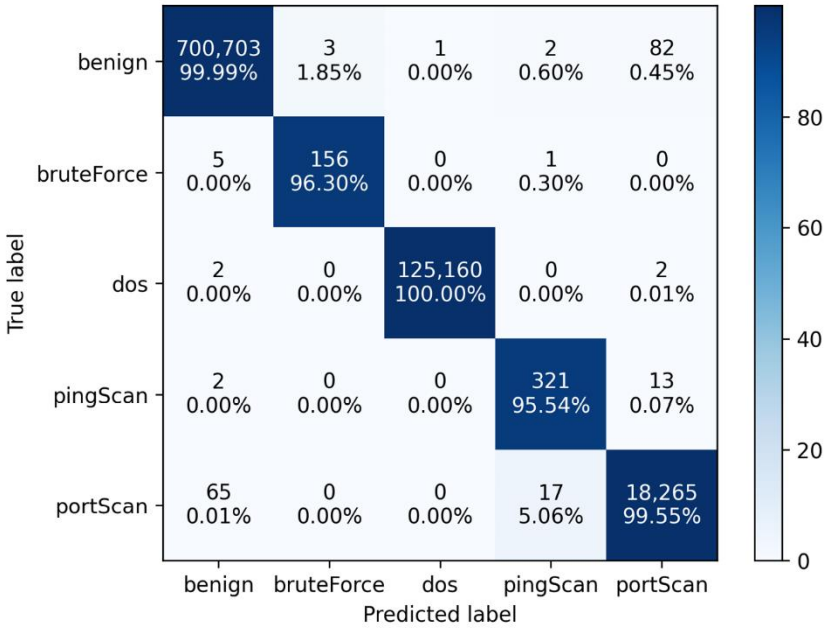



1e6
Duration
Number of packets
Number of bytes
1e6

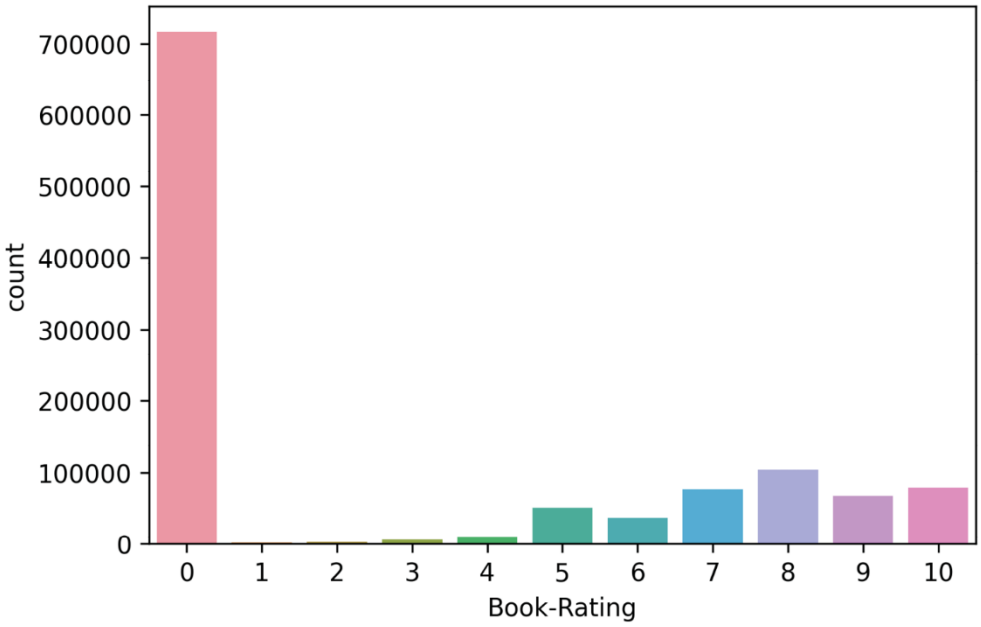

1e6
Duration
Number of packets
Number of bytes

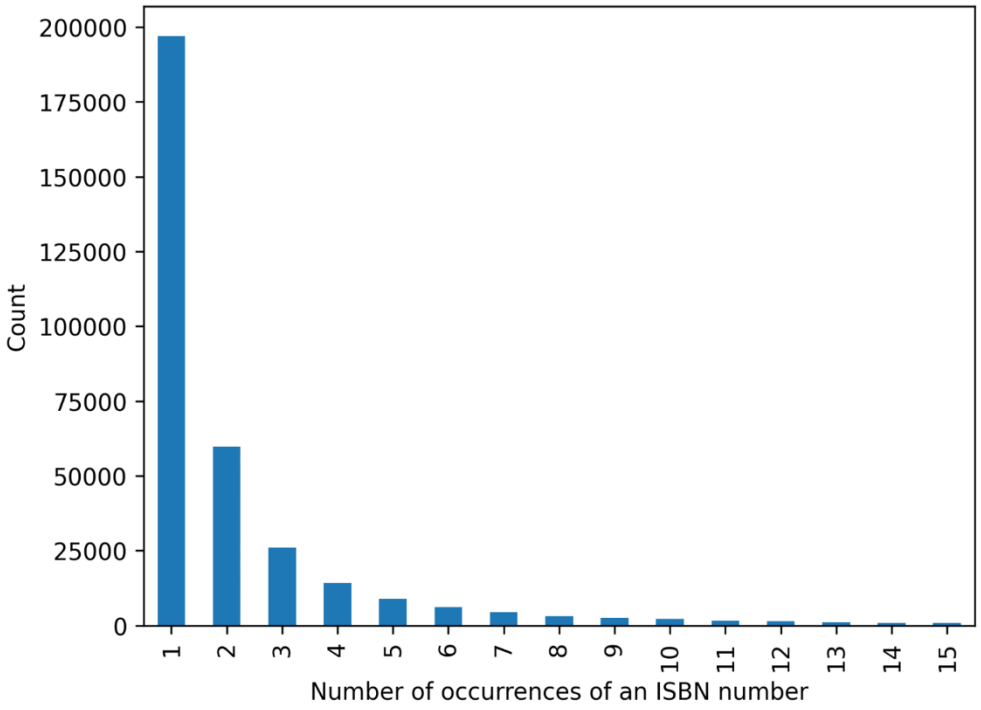

host
SAGEConv
SAGEConv
Leaky ReLU
SAGEConv
SAGEConv
Leaky ReLU
SAGEConv
SAGEConv
Leaky ReLU
SAGEConv
SAGEConv
Leaky ReLU
host
flow
SAGEConv
Leaky ReLU
SAGEConv
Leaky ReLU
SAGEConv
Leaky ReLU
Linear
classes

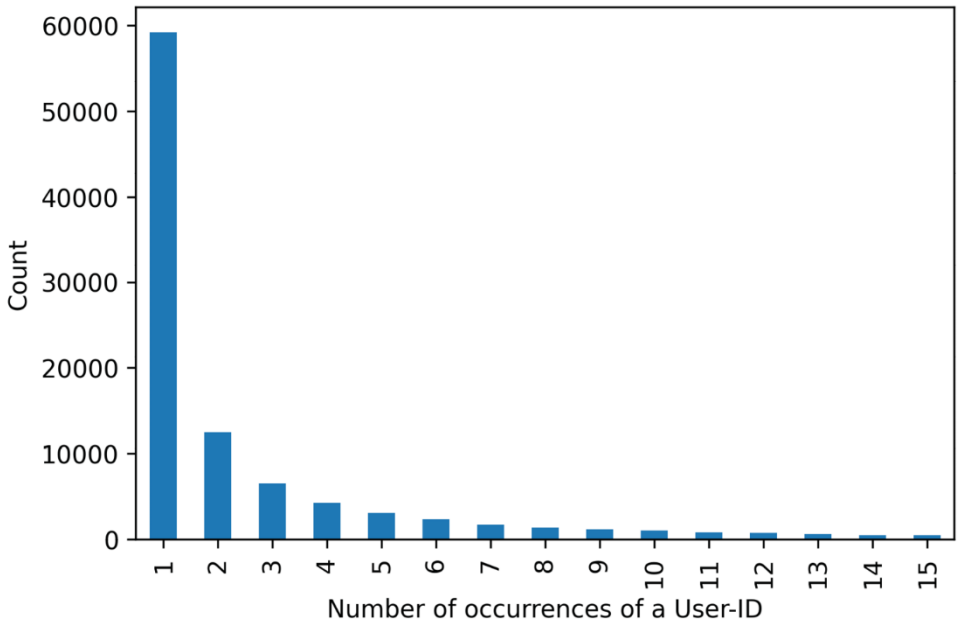

benign
45%
2%
portScan
3%
pingScan
8%
dos
42%
bruteForce

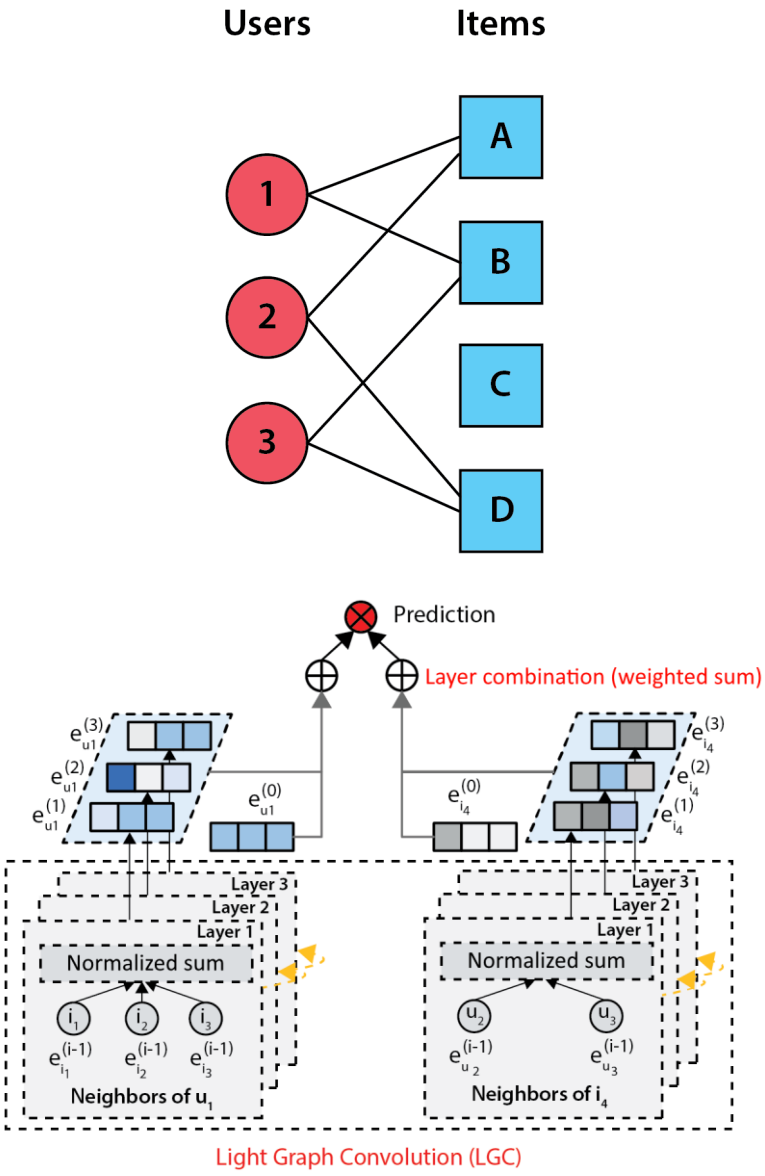

benign
bruteForce
dos
pingScan
portScan
True label
700,703
99.99%
3
1.85%
1
0.00%
2
0.60%
82
0.45%
5
0.00%
156
96.30%
0
0.00%
1
0.30%
0
0.00%
2
0.00%
0
0.00%
125,160
100.00%
0
0.00%
2
0.01%
2
0.00%
0
0.00%
0
0.00%
321
95.54%
13
0.07%
65
0.01%
0
0.00%
0
0.00%
17
5.06%
18,265
99.55%
benign
bruteForce
dos
pingScan
portScan
Predicted label
80
60
40
20
0

# Chapter 17: Recommending Books Using LightGCN




200000
175000
150000
125000
100000
75000
50000
25000
0
Count
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Number of occurrences of an ISBN number


60000
50000
40000
30000
20000
10000
0
Count
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Number of occurrences of a User-ID


Users
Items
1
2
3
A
B
C
D
Prediction
Layer combination (weighted sum)
Layer 3
Layer 2
Layer 1
Normalized sum
Neighbors of u1
Neighbors of i4
Light Graph Convolution (LGC)

# Chapter 18: Unlocking the Potential of Graph Neural Networks for Real-Word Applications

No Images…