

Rețele neuronale și deep learning

Cursul 9: Rețele neuronale pe grafuri



Prof. dr. Daniel Traian PELE

Academia de Studii Economice din București (ASE), Departamentul de Statistică și Econometrie
Academia Română, Institutul de Prognoză Economică (IPE)
IDA Institute Digital Assets ▪ AI4EFin ▪ MSCA Digital Finance

Obiectivele cursului

- Reprezentarea datelor financiare ca graf
  - Noduri, muchii și variabilele nodurilor
  - Matricea de adiacență, matricea gradelor și adiacența normalizată
- Schema de transmitere a mesajelor a unei rețele neuronale pe graf (GNN)
  - Cele trei straturi clasice: rețeaua convoluțională pe graf (GCN), eșantionarea și agregarea pe graf (GraphSAGE), rețeaua cu atenție pe graf (GAT)
- Antrenarea lor pe graful Bitcoin Elliptic pentru combaterea spălării banilor (AML), în ordine cronologică
  - Comparația cu regresia logistică și cu pădurea aleatoare
- Evaluarea rezultatelor ca un investigator
  - Precizia, recall-ul și aria de sub curba precizie-recall (PR-AUC)
  - Alertele pe analist, schimbarea în timp și stabilitatea explicațiilor

Abrevieri

AML	combaterea spălării banilor (anti-money laundering)
ASE	Academia de Studii Economice din București
BTC	bitcoin (unitate monetară)
CNN	rețea neuronală convoluțională
CPU	unitate centrală de procesare
GAT	rețea cu atenție pe graf (graph attention network)
GCN	rețea convoluțională pe graf (graph convolutional network)
GIN	rețea de izomorfism pe graf (graph isomorphism network)
GNN	rețea neuronală pe graf (graph neural network)
GraphSAGE	eșantionare și agregare pe graf (graph sample and aggregate)
IPE	Institutul de Prognoză Economică, Academia Română
MLP	perceptron multistrat
MPNN	rețea neuronală cu transmitere de mesaje (message passing)
PCA	analiza componentelor principale
PR-AUC	aria de sub curba precizie-recall
PyG	PyTorch Geometric (bibliotecă de programe)
ReLU	unitate liniară rectificată (rectified linear unit)
SHAP	explicații aditive bazate pe valorile Shapley (SHapley Additive exPlanations)
WL	testul de izomorfism Weisfeiler–Leman
XAI	inteligență artificială explicabilă

Cuprins

- Grafurile în detectarea infracțiunilor financiare
- Grafurile ca matrice
- Transmiterea mesajelor
- GCN, GraphSAGE și GAT
- Studiu de caz: tranzacții Bitcoin ilicite
- Rețelele GNN în producție
- Concluzii

Structura de rețea în detectarea spălării banilor

⊡ **Pe scurt:** O plată izolată poate părea obișnuită; lanțul de plăți din jurul ei, de cele mai multe ori, nu.

$$G = (\mathcal{V}, \mathcal{E}, \mathbf{X}), \qquad \mathbf{X} \in \mathbb{R}^{n \times d}, \qquad y_v \in \{0, 1\}$$

⊡ **unde**

- $\mathcal{V}$ = nodurile (tranzacțiile), $n = |\mathcal{V}|$
- $\mathcal{E}$ = muchiile (un flux de plată de la o tranzacție la următoarea)
- $\mathbf{X}$ = variabilele nodurilor, cîte d pentru fiecare nod
- $y_v = 1$ dacă tranzacția v este ilicită.

⊡ **Graful Elliptic** (Weber et al., 2019)

- 203 769 de tranzacții Bitcoin, 234 355 de fluxuri de plată
- 165 de variabile pe tranzacție, 49 de pași de timp

⊡ **Sarcina**

- Clasificarea nodurilor: ilicit (fraude, ransomware, piețe ilegale) sau licit

⊡ **În practică:** echipa de conformitate a unei bănci gîndește deja în lanțuri de plăți

- O rețea GNN învață din aceleași lanțuri.

(Weber et al., 2019; Bishop & Bishop, 2024, ch. 13)

Exercițiu practic: notebook, secțiunea 1

Descrierea setului de date Elliptic

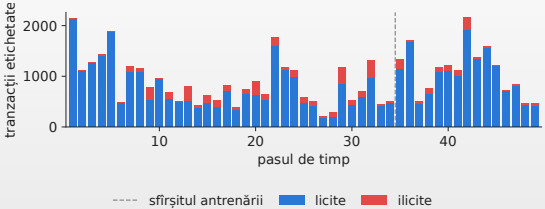
Element	Valoarea în copia noastră a datelor
Noduri, muchii	203 769 de tranzacții; 234 355 de fluxuri de plată orientate
Etichete	4 545 ilicite, 42 019 licite, 157 205 necunoscute (22,9 % etichetate)
Variabile	93 locale (sume, comisioane, intrări, ieșiri) și 72 agregate pe vecini
Timpul	49 de pași de circa două săptămîni; muchiile leagă tranzacții din același pas (100,0 %)
Antrenare / test	pașii 1–34 (29 894 etichetate) / pașii 35–49 (16 670 etichetate, 1 083 ilicite)

- **Dezechilibrul claselor**: doar 9,76 % dintre tranzacțiile etichetate sînt ilicite
  - Acuratețea singură nu spune nimic
- **Date anonimizate**: Elliptic nu publică denumirile variabilelor
  - Le notăm $f_1, \ldots, f_{165}$

(Weber et al., 2019)

Exercițiu practic: notebook, secțiunea 1

Evoluția în timp a tranzacțiilor ilicite



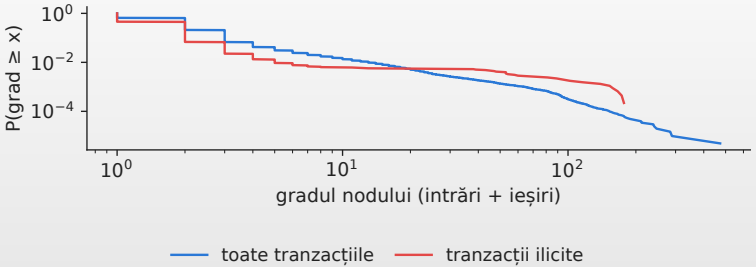
Sursa: Graful tranzacțiilor Bitcoin Elliptic (Weber et al., 2019). Date: graful tranzacțiilor Bitcoin Elliptic, 49 de pași de timp (circa două săptămîni fiecare).

- **Vîrful**: 342 de tranzacții ilicite în pasul 32
  - În medie, circa 4 159 de tranzacții pe pas
- **Împărțire în timp**: antrenare pe pașii 1–34, test o singură dată pe pașii 35–49, ca în Weber et al. (2019)
  - Pașii nu se amestecă

- **În practică:** un model este judecat pe lunile de după datele de antrenare, exact cum va fi folosit.

NN_L9_elliptic_timesteps

Distribuția gradelor în graful tranzacțiilor



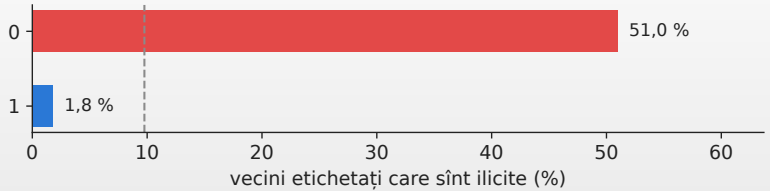
Sursa: Graful tranzacțiilor Bitcoin Elliptic (Weber et al., 2019). Date: graful tranzacțiilor Bitcoin Elliptic, 49 de pași de timp (circa două săptămîni fiecare).

- **Graf rar**: gradul mediu 2,30, mediana 2
  - 34,5 % dintre noduri au un singur vecin
- **Coadă grea**: cel mai mare nod central are 473 de vecini
  - O rețea GNN nu trebuie să lase nodurile centrale să le acopere pe celelalte (de aici normalizarea)

$\deg(v) = |\mathcal{N}(v)|$

NN_L9_degree_dist

Homofilia tranzacțiilor ilicite



Sursa: Graful tranzacțiilor Bitcoin Elliptic (Weber et al., 2019). Date: graful tranzacțiilor Bitcoin Elliptic, 49 de pași de timp (circa două săptămîni fiecare).

- **Homofilie**: 51,0 % dintre vecinii etichetați ai unui nod ilicit sînt ilicite, față de 1,8 % pentru un nod licit
- **Homofilia muchiilor**: 95,4 % dintre muchiile etichetate leagă noduri din aceeași clasă
  - Cu etichete aleatoare: 82,4 %

- **În practică:** homofilia este motivul pentru care ne uităm la vecini: cu cine tranzacționezi spune ceva despre tine.

NN_L9_homophily

Cuprins

- Grafurile în detectarea infracțiunilor financiare
- Grafurile ca matrice
- Transmiterea mesajelor
- GCN, GraphSAGE și GAT
- Studiu de caz: tranzacții Bitcoin ilicite
- Rețelele GNN în producție
- Concluzii

Matricea de adiacență ca reprezentare algebrică a grafului

- **Pe scurt:** Orice operație a unei rețele GNN este un produs între matricea variabilelor, o matrice de adiacență (normalizată) și o matrice de ponderi.

$$A_{uv} = \begin{cases} 1, & (u,v) \in \mathcal{E} \\ 0, & \text{altfel} \end{cases}, \qquad D = \operatorname{diag}\left(\sum_v A_{uv}\right)$$

$$\tilde{A} = A + I, \qquad \hat{A} = \tilde{D}^{-1/2}\tilde{A}\,\tilde{D}^{-1/2}$$

- **unde**
  - A = matricea de adiacență, simetrizată
  - D = matricea gradelor
  - $\tilde{A}$ adaugă bucle (nodul își păstrează variabilele)
  - $\hat{A}$ are elementele $1/\sqrt{\tilde{d}_u \tilde{d}_v}$.

- **Stocare rară**
  - A are $n^2 \approx 4 \cdot 10^{10}$ celule, dar numai 234 355 de muchii
  - Bibliotecile păstrează o listă de muchii $2 \times |\mathcal{E}|$, ca PyTorch Geometric (PyG)

- **De ce normalizăm**
  - Fără normalizare, un nod central de grad 473 ar aduna 473 de mesaje, iar scala ar exploda

- **În practică:** rețeaua de plăți a unei bănci se stochează ca listă de muchii, niciodată ca matrice densă.

(Kipf & Welling, 2017)

Exercițiu practic: notebook, secțiunea 2

Invarianța și echivarianța la permutări ale modelelor pe graf

- **Pe scurt:** Tranzacția 17 și tranzacția 4.203 sînt nume, nu poziții; renumerotarea lor nu trebuie să schimbe nicio predicție.

$$f\left(P\mathbf{X},\ PAP^\top\right) = P\, f(\mathbf{X}, A) \quad \forall P$$

- **unde**
  - P = orice matrice de permutare $n \times n$
  - f dă cîte un rînd de rezultate pentru fiecare nod: *echivarianță*
  - Un rezultat la nivelul întregului graf trebuie să fie *invariant*: $g(P\mathbf{X}, PAP^\top) = g(\mathbf{X}, A)$.

- **De ce nu un perceptron multistrat (MLP)?**
  - Pe rîndurile lui A ar avea nevoie de n intrări și s-ar schimba odată cu numerotarea
- **De ce nu o rețea neuronală convoluțională (CNN)?**
  - Presupune o grilă cu un număr fix de vecini ordonați
  - Un nod are între 1 și 473 de vecini, fără nicio ordine

- **În practică:** întrebați despre orice model cumpărat: se schimbă scorul dacă fișierul de intrare este sortat altfel? Nu are voie.

(Bishop & Bishop, 2024, sec. 13.1)

Semnale netede pe graf: laplacianul

- **Pe scurt:** Laplacianul grafului măsoară cît diferă un semnal de la un nod la vecinii lui; straturile GCN sînt filtre trece-jos ale acestui semnal.

$$L = D - A, \qquad \mathbf{x}^\top L\,\mathbf{x} = \sum_{(u,v)\in\mathcal{E}} \big(x_u - x_v\big)^2 \;\geq 0$$

$$L = U\Lambda U^\top, \qquad \hat{\mathbf{x}} = U^\top \mathbf{x} \;\text{ (transformata Fourier pe graf)}$$

- **unde**
  - $\mathbf{x} \in \mathbb{R}^n$ = o variabilă pe toate nodurile
  - $\mathbf{x}^\top L\mathbf{x}$ mic = vecinii seamănă (homofilie)
  - U, Λ = vectorii și valorile proprii ale lui L (frecvențele).

- **Filtru spectral**
  - $g_\theta \star \mathbf{x} = U\, g_\theta(\Lambda)\, U^\top \mathbf{x}$
  - Costisitor: U este densă, $O(n^2)$
- **GCN**
  - Un polinom de gradul întîi în L duce la $\hat{A}$ de pe slide-ul anterior
  - Derivarea, în Anexa A

- **În practică:** grafurile de fraudă cu homofilie sînt semnale netede: media pe vecini elimină zgomot, nu informație.

(Kipf & Welling, 2017; Hamilton, 2020, ch. 7)

Trei sarcini de predicție: la nivel de nod, de muchie și de graf

- **Pe scurt:** Aceleași straturi de transmitere a mesajelor răspund la trei tipuri de întrebări; se schimbă doar ieșirea.

$$\hat{y}_v = \sigma(\mathbf{w}^\top \mathbf{h}_v), \quad \hat{y}_{uv} = \sigma(\mathbf{h}_u^\top \mathbf{h}_v), \quad \hat{y}_G = \sigma\left(\mathbf{w}^\top \sum_v \mathbf{h}_v\right)$$

- **unde**
  - $\mathbf{h}_v$ = reprezentarea nodului v după ultimul strat
  - σ = funcția logistică
  - Suma face ca ieșirea pe graf să nu depindă de numerotare.

- **Nod**
  - Este această tranzacție ilicită? (acest curs)
- **Muchie**
  - Vor face aceste două firme comerț; este această plată un transfer prin intermediari?
- **Graf**
  - Este acest grup de conturi o rețea de spălare; este această rețea interbancară fragilă?

- **În practică:** stabiliți întîi unitatea de decizie (o plată, un client, o rețea), apoi sarcina pe graf.

(Hamilton, 2020, ch. 1)

Cuprins

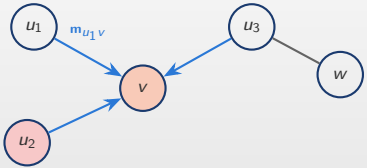
- Grafurile în detectarea infracțiunilor financiare
- Grafurile ca matrice
- Transmiterea mesajelor
- GCN, GraphSAGE și GAT
- Studiu de caz: tranzacții Bitcoin ilicite
- Rețelele GNN în producție
- Concluzii

Un strat GNN: funcțiile de mesaj, de agregare și de actualizare

- **Pe scurt:** Fiecare nod trimite un mesaj vecinilor; fiecare nod adună ce primește și își actualizează starea.

$$\mathbf{m}_v^{(l)} = \bigoplus_{u \in \mathcal{N}(v)} \phi^{(l)}\left(\mathbf{h}_u^{(l)}, \mathbf{h}_v^{(l)}, \mathbf{e}_{uv}\right)$$

$$\mathbf{h}_v^{(l+1)} = \psi^{(l)}\left(\mathbf{h}_v^{(l)}, \mathbf{m}_v^{(l)}\right), \qquad \mathbf{h}_v^{(0)} = \mathbf{x}_v$$



- **unde**
  - $\mathcal{N}(v)$ = vecinii lui v
  - ϕ = funcția mesajului
  - $\bigoplus$ = agregator care nu depinde de ordine (sumă, medie, maxim)
  - ψ = actualizarea (o rețea mică)
  - $\mathbf{e}_{uv}$ = variabilele muchiei, dacă există.

- **Rețeaua neuronală cu transmitere de mesaje (MPNN)**: fiecare strat din acest curs este o alegere a lui ϕ, $\bigoplus$, ψ

- **În practică:** ponderile ϕ, ψ sînt comune tuturor nodurilor: dimensiunea modelului nu crește odată cu graful.

(Gilmer et al., 2017; Bishop & Bishop, 2024, sec. 13.2)

Exercițiu practic: notebook, secțiunea 3

Agregatorul și puterea de discriminare a unei rețele GNN

- **Pe scurt:** Media și maximul uită cîți vecini au trimis mesaje; suma păstrează această informație și devine la fel de puternică precum testul Weisfeiler–Leman.

$$\mathbf{h}_v^{(l+1)} = \mathrm{MLP}^{(l)}\left((1+\varepsilon)\,\mathbf{h}_v^{(l)} + \sum_{u \in \mathcal{N}(v)} \mathbf{h}_u^{(l)}\right)$$

- **unde**
  - Stratul rețelei de izomorfism pe graf (GIN)
  - ε = scalar estimat
  - Suma pe un multiset este injectivă dacă MLP-ul este suficient de bogat, deci GIN deosebește orice vecinătăți pe care le deosebește testul Weisfeiler–Leman (WL).

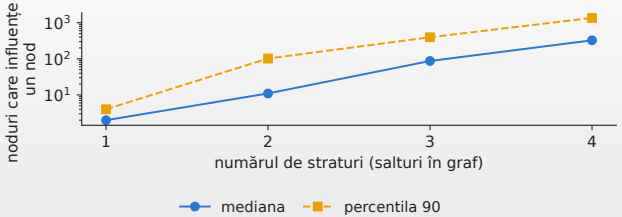
- **Exemplu**
  - Vecinii $\{a, a\}$ și $\{a\}$: aceeași medie, același maxim, sumă diferită

- **În fraudă**
  - “o plată de la un serviciu de amestecare” și “zece astfel de plăți” nu trebuie să arate la fel
  - Dar sumele nodurilor centrale cer normalizare; de aceea GCN folosește $\hat{A}$

- **În practică:** alegeți agregatorul după întrebare: contează numărul (suma), contează proporția (media), contează un singur semnal de alarmă (maximul).

(Xu et al., 2019)

Fiecare strat lărgește vecinătatea care ajunge la un nod



Sursa: Graful tranzacțiilor Bitcoin Elliptic (Weber et al., 2019). Date: graful tranzacțiilor Bitcoin Elliptic, 49 de pași de timp (circa două săptămîni fiecare).

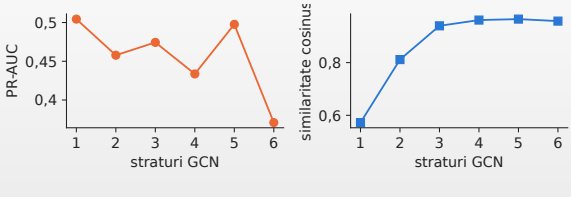
- **Cîmpul receptiv** al unei tranzacții de test: mediana este 2 noduri după 1 strat, 11 după 2, 325 după 4
- **Coada**: percentila 90 ajunge la 1 356 de noduri după 4 straturi
  - Nodurile centrale leagă părți îndepărtate ale grafului

- **În practică:** L straturi $= L$ salturi ale fluxului de bani
  - Două salturi acoperă deja mare parte din ce ar verifica manual un investigator.

(calcule proprii, 2.000 de noduri de test)

NN_L9_receptive_field

Netezirea excesivă (over-smoothing) în rețelele pe graf cu multe straturi



Sursa: Graful tranzacțiilor Bitcoin Elliptic (Weber et al., 2019). Date: estimări proprii; antrenare pe pașii 1-34, pașii de timp de test 35-49.

- **Precizia**: PR-AUC 0,504 cu 1 strat GCN, 0,458 cu 2, 0,371 cu 6
  - Adîncimea cea mai bună: 1
- **Similaritatea**: similaritatea cosinus medie a reprezentărilor crește de la 0,57 (1 strat) la 0,96 (6 straturi)
  - Mederea repetată le aduce spre un vector comun

- **În practică:** rețelele GNN sînt intenționat puțin adînci: două-trei straturi, apoi conexiuni reziduale dacă este nevoie de mai mult.

(calcule proprii, 2 semințe; Li, Han & Wu, 2018)

NN_L9_oversmoothing

Cuprins

- Grafurile în detectarea infracțiunilor financiare
- Grafurile ca matrice
- Transmiterea mesajelor
- GCN, GraphSAGE și GAT
- Studiu de caz: tranzacții Bitcoin ilicite
- Rețelele GNN în producție
- Concluzii

GCN: media vecinilor ponderată cu gradele, urmată de o transformare liniară

- **Pe scurt:** Un strat GCN este o regresie liniară pe media normalizată a nodului și a vecinilor lui, urmată de o unitate liniară rectificată (ReLU).

$$H^{(l+1)} = \mathrm{ReLU}\left(\hat{A}\, H^{(l)} W^{(l)}\right)$$

$$\mathbf{h}_v^{(l+1)} = \mathrm{ReLU}\left(\sum_{u \in \mathcal{N}(v) \cup \{v\}} \frac{W^{(l)\top} \mathbf{h}_u^{(l)}}{\sqrt{\tilde{d}_u \tilde{d}_v}}\right)$$

- **unde**
  - $H^{(l)} \in \mathbb{R}^{n \times d_l}$ = reprezentările nodurilor, $H^{(0)} = \mathbf{X}$
  - $W^{(l)}$ = ponderi comune tuturor nodurilor
  - $\mathrm{ReLU}(x) = \max(0, x)$
  - ieșirea: $\mathrm{softmax}(\hat{A} H^{(1)} W^{(2)})$.

- **Rețeaua GCN folosită**
  - 2 straturi, 100 de unități ascunse, dropout 0,2: 26 902 parametri
  - Antrenare pe tot graful odată, 66 s pe sămînță, pe unitatea centrală de procesare (CPU)

- **Costul**
  - $O(|\mathcal{E}|\, d_l + n\, d_l d_{l+1})$ pe strat: liniar în numărul de muchii

- **În practică:** o rețea GCN este analogul pe graf al unei medii mobile: robustă, ieftină și oarbă la care vecin contează cel mai mult.

(Kipf & Welling, 2017)

Exercițiu practic: notebook, secțiunea 3

GraphSAGE: reprezentarea separată a nodului și eșantionarea vecinilor

- **Pe scurt:** GraphSAGE alătură starea proprie a nodului și media unui eșantion de vecini, deci funcționează și pe noduri nevăzute la antrenare.

$$\mathbf{h}_v^{(l+1)} = \mathrm{ReLU}\left(W_1^{(l)} \mathbf{h}_v^{(l)} + W_2^{(l)} \frac{1}{|\mathcal{S}(v)|} \sum_{u \in \mathcal{S}(v)} \mathbf{h}_u^{(l)} \right)$$

- **unde**
  - $\mathcal{S}(v) \subseteq \mathcal{N}(v)$ = un eșantion aleator de cel mult k vecini (toți, în rularea noastră pe tot graful)
  - W_1, W_2 = ponderi separate pentru nod și pentru vecinătatea lui.

- **Inductiv**
  - Ponderile nu depind de graf: tranzacțiile de mîine primesc un scor fără reantrenare
- **Scalabil**
  - Mini-loturi de eșantioane pe k salturi: costul este limitat la k^L pe nod, nu de gradul nodurilor centrale
- **Pe Elliptic**
  - F1 pe clasa ilicită 0,564 (abatere standard 0,022), PR-AUC 0,582

- **În practică:** sistemele de plăți care dau scoruri în timp real au nevoie de straturi inductive precum GraphSAGE.

(Hamilton, Ying & Leskovec, 2017)

Exercițiu practic: notebook, secțiunea 4

GAT: ponderi de atenție învățate pentru vecini

⊡ **Pe scurt:** Un strat GAT ponderează fiecare vecin cu un scor învățat: atenția din cursul 7, restrînsă la muchii.

$$\alpha_{vu} = \frac{\exp\left(\mathrm{LeakyReLU}(\mathbf{a}^\top [W\mathbf{h}_v \,\|\, W\mathbf{h}_u])\right)}{\sum_{k \in \mathcal{N}(v)\cup\{v\}} \exp\left(\mathrm{LeakyReLU}(\mathbf{a}^\top [W\mathbf{h}_v \,\|\, W\mathbf{h}_k])\right)}$$

$$\mathbf{h}_v^{(l+1)} = \Big\|_{k=1}^{K} \ \mathrm{ReLU}\left(\sum_u \alpha_{vu}^k W^k \mathbf{h}_u\right)$$

⊡ **unde**

- $\mathbf{a}$ = vectorul de atenție
- $\|$ = concatenare
- K = numărul de capete (aici 4)
- Ponderile α_{vu} însumează unu pe vecinii lui v.

⊡ **Legătura cu transformerele**

- Un transformer este o rețea GAT pe graful complet al tokenilor

⊡ **Pe Elliptic**

- 27 302 parametri; F1 pe clasa ilicită 0,489, PR-AUC 0,496

⊡ **În practică:** ponderile de atenție sînt o primă explicație, nu una finală: arată unde s-a uitat modelul, nu de ce.

(Veličković et al., 2018)

Exercițiu practic: notebook, secțiunea 4

Antrenarea pe un singur graf mare: măști, ponderi de clasă, oprire timpurie

- **Pe scurt:** Toate nodurile transmit mesaje, dar în funcția de pierdere intră doar nodurile etichetate de antrenare; pașii de validare decid oprirea.

$$\mathcal{L}(\theta) = -\frac{1}{|\mathcal{V}_{tr}|} \sum_{v \in \mathcal{V}_{tr}} \left[w_1 \, y_v \log \hat{p}_v + w_0 (1 - y_v) \log(1 - \hat{p}_v) \right]$$

- **unde**
  - $\mathcal{V}_{tr}$ = nodurile etichetate din pașii 1–29
  - $\hat{p}_v$ = probabilitatea estimată de ilicit
  - $w_1 = 0{,}7$, $w_0 = 0{,}3$ = ponderile de clasă din Weber et al. (2019)
  - Adam, rata de învățare 0,003, penalizarea ponderilor $5 \cdot 10^{-4}$.

- **Validarea**
  - Nodurile etichetate din pașii 30–34: epoca cea mai bună după PR-AUC, răbdare 40 de epoci
  - Epoca mediană reținută pentru GCN: 47
- **Standardizare fără scurgeri de informație**
  - Mediile și abaterile standard ale variabilelor numai din pașii 1–34

- **În practică:** nodurile necunoscute nu se pierd: ele transmit mesaje între nodurile etichetate (învățare semi-supervizată).

(Kipf & Welling, 2017; Weber et al., 2019)

Exercițiu practic: notebook, secțiunea 3

Cuprins

- Grafurile în detectarea infracțiunilor financiare
- Grafurile ca matrice
- Transmiterea mesajelor
- GCN, GraphSAGE și GAT
- Studiu de caz: tranzacții Bitcoin ilicite
- Rețelele GNN în producție
- Concluzii

Opt modele evaluate pe un set de test cronologic

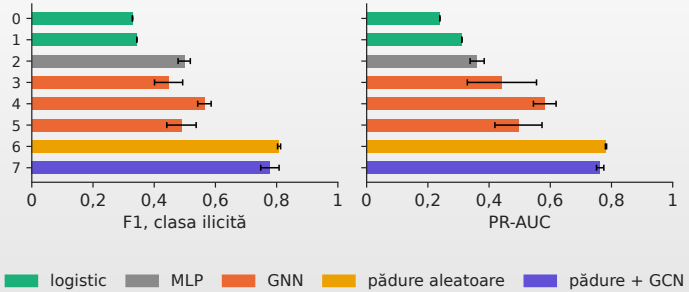
Model	Ce vede
Regresie logistică	doar variabilele locale (93), apoi toate cele 165
Pădure aleatoare	toate variabilele; 100 de arbori, 50 de variabile la fiecare împărțire (Weber et al., 2019)
MLP (fără graf)	toate variabilele, aceleași straturi ca GCN, dar fără mesaje: izolează valoarea muchiilor
GCN, GraphSAGE, GAT	toate variabilele și graful; 2 straturi, 100 de unități
Pădure + GCN	toate variabilele și cele 100 de ieșiri ascunse GCN ale fiecărui nod

- **Măsuri pe pașii 35–49**: precizia, recall-ul și F1 pe clasa ilicită, la pragul 0,5
  - PR-AUC (fără prag)
- **Incertitudinea**: 3 semințe pentru fiecare rețea și pădure
  - Bootstrap pe nodurile de test pentru diferențe

(Weber et al., 2019)

Exercițiu practic: notebook, secțiunea 2

Rezultatele pe Elliptic: pădurea aleatoare și contribuția straturilor pe graf



Sursa: Graful tranzacțiilor Bitcoin Elliptic (Weber et al., 2019). Date: estimări proprii; antrenare pe pașii 1-34, pașii de timp de test 35-49.

- **F1 pe clasa ilicită**: pădurea 0,808, pădurea + GCN 0,778, GCN 0,447, MLP 0,498, logistic 0,343
- **Valoarea muchiilor**: PR-AUC GCN minus MLP, cu aceleași straturi: +8,1 puncte

(calcule proprii; media ± abaterea standard pe 3 semințe)

NN_L9_model_comparison

Precizie, recall și costul antrenării

Model	Prec.	Recall	F1	PR-AUC	Sec.
Logistic, locale	0,208	0,789	0,329	0,239	0,8
Logistic, toate	0,220	0,771	0,343	0,311	1,8
MLP	0,517	0,483	0,498	0,361	59,2
GCN	0,484	0,494	0,447	0,442	65,5
GraphSAGE	0,569	0,561	0,564	0,582	143,2
GAT	0,488	0,500	0,489	0,496	231,2
Pădure aleatoare	0,914	0,725	0,808	0,782	4,9
Pădure + GCN	0,933	0,668	0,778	0,763	186,0

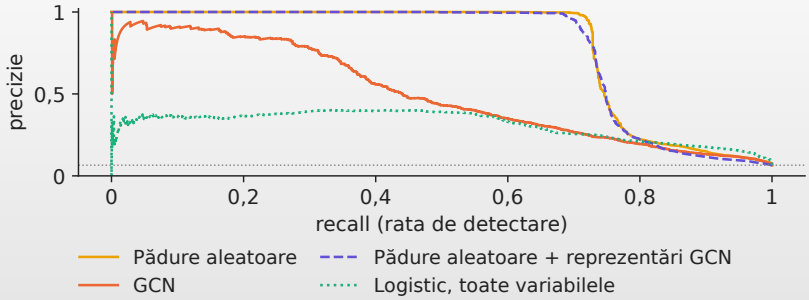
- **Cum citim tabelul**
  - Precizia = ponderea alertelor care sînt ilicite; recall = ponderea ilicitelor găsite
  - Verde = cel mai bun, roșu = cel mai slab din coloană
- **Pădurea**
  - Precizie 0,914, recall 0,725
- **Logistic, ponderi de clasă 0,3/0,7**
  - Precizie 0,220, recall 0,771: ponderile cumpără recall cu alarme false

- **În practică:** modelul potrivit depinde de costul unei erori: un caz de spălare ratat sau o oră pierdută de un analist.

(calcule proprii; sec. = timpul de antrenare pe sămînță, CPU)

Exercițiu practic: notebook, secțiunea 2

Curbele precizie-recall ale modelelor



Sursa: Graful tranzacțiilor Bitcoin Elliptic (Weber et al., 2019). Date: estimări proprii; antrenare pe pașii 1-34, pașii de timp de test 35-49.

- **PR-AUC**: pădurea 0,782, pădurea + GCN 0,763, GCN 0,442, logistic 0,311
  - Un scor aleator ar da 6,50 % (linia punctată)

PR-AUC $= \sum_k (R_k - R_{k-1}) P_k$

NN_L9_pr_curves

Testul bootstrap al diferenței pe tranzacțiile de test

- **Pe scurt:** Un singur set de test dă un singur număr; reeșantionarea nodurilor de test arată cît s-ar putea mișca acest număr.

$$\Delta^{*b} = \text{PR-AUC}_A\left(\mathcal{T}^{*b}\right) - \text{PR-AUC}_B\left(\mathcal{T}^{*b}\right), \quad b = 1, \ldots, B$$

$$\text{CI}_{95\%} = \left[q_{0.025}(\Delta^*),\ q_{0.975}(\Delta^*)\right]$$

- **unde**
  - $\mathcal{T}^{*b}$ = noduri de test extrase cu întoarcere
  - $B = 1.000$
  - Scorurile = media pe 3 semințe
  - Aceeași extragere pentru ambele modele (comparație pereche).

- **Pădurea minus GCN**
  - $\Delta = 0{,}285$, intervalul de 95 % [0,259; 0,313]
- **Pădurea + GCN minus pădurea**
  - $\Delta = -0{,}008$, intervalul de 95 % [−0,015; −0,002]
- un interval care nu conține 0 susține o diferență reală pe această perioadă de test

- **În practică:** raportați intervalul, nu doar cîștigătorul: un comitet antifraudă decide pe intervale, nu pe a treia zecimală.

(Efron & Tibshirani, 1993)

Exercițiu practic: notebook, secțiunea 2

Agregatele de vecinătate din variabile și performanța pădurii aleatoare

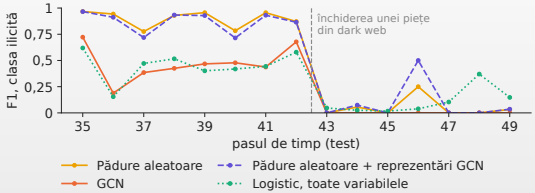
- **Transmitere de mesaje construită manual**
  - 72 dintre cele 165 de variabile sînt agregate pe vecinii de la un salt (maxim, minim, abatere standard, corelație)
  - Este un strat GNN fix, calculat de Elliptic înainte de publicarea datelor
- **Doar variabilele locale**
  - Regresia logistică are F1 0,329, față de 0,343 cu agregatele
- **Arborii**
  - Pragurile pe sume și numere se potrivesc tiparelor de fraudă care sînt reguli (“mai mult de k intrări”)
- **Unde își justifică rețelele GNN costul**
  - Grafuri brute, fără agregate construite manual
  - Tipare pe mai multe salturi (stratificarea prin multe conturi)
  - Scoruri inductive pentru noduri noi și reprezentări pentru alte modele
- **Combinați**
  - Pădure + reprezentări GCN: variația lui F1 este −3,0 puncte

În practică: înainte de a antrena o rețea GNN, verificați dacă furnizorul datelor nu a agregat deja vecinătatea.

(Weber et al., 2019)

Exercițiu practic: notebook, secțiunea 2

Schimbarea conceptului după închiderea unei piețe ilegale



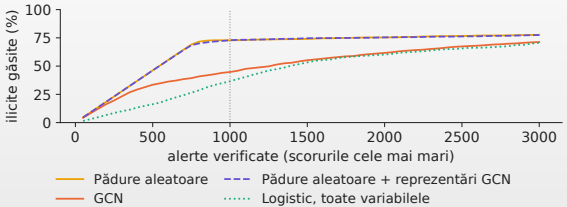
Sursa: Graful tranzacțiilor Bitcoin Elliptic (Weber et al., 2019). Date: estimări proprii; antrenare pe pașii 1-34, pașii de timp de test 35-49.

- **Înainte de pasul 43**: F1 mediu pădure 0,898, GCN 0,473
  - **de la pasul 43**: 0,048 și 0,000
- **Comportament nou**: după închiderea unei piețe ilegale importante, fluxurile ilicite și-au schimbat forma
  - Niciun model antrenat pe pașii 1–34 nu o văzuse (Weber et al., 2019)

- **În practică:** monitorizați F1 pe fiecare perioadă și reantrenați pe etichete recente
  - Un model AML static îmbătrînește în cîteva săptămîni.

NN_L9_f1_by_timestep

Tranzacțiile ilicite identificate între primele 1 000 de alerte



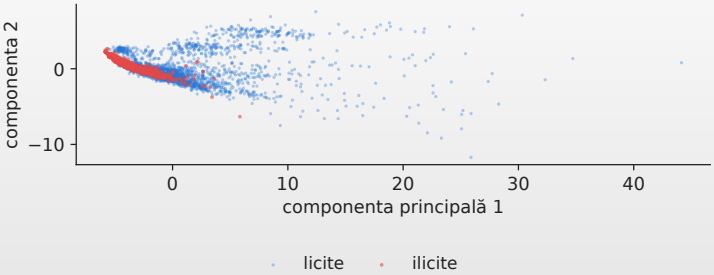
Sursa: Graful tranzacțiilor Bitcoin Elliptic (Weber et al., 2019). Date: estimări proprii; antrenare pe pașii 1-34, pașii de timp de test 35-49.

- **Primele 1 000 de scoruri** din perioada de test conțin 72,9 % dintre toate tranzacțiile ilicite pentru pădure, 44,8 % pentru GCN și 72,9 % pentru pădure + GCN
- **Rata de reușită** a acestor alerte: 78,9 % (pădurea), 48,5 % (GCN), din 1 083 de tranzacții ilicite de test

- **În practică:** dimensionați echipa după această curbă: bugetul de alerte, nu F1, este constrîngerea unui departament de conformitate.

NN_L9_alert_budget

Tranzacțiile ilicite se grupează în spațiul de reprezentare al GCN



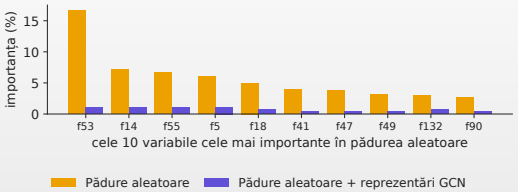
Sursa: Graful tranzacțiilor Bitcoin Elliptic (Weber et al., 2019). Date: estimări proprii; antrenare pe pașii 1-34, pașii de timp de test 35-49.

- **Stratul ascuns** al rețelei GCN (100 de dimensiuni), proiectat prin analiza componentelor principale (PCA) pe primele două componente: 79,2 % din variață
- **Refolosire**: acești vectori sînt intrările modelului pădure + GCN și pot alimenta orice alt model de risc

(calcule proprii; 1 083 de noduri ilicite și 3.000 de noduri licite de test)

NN_L9_gcn_embeddings

Efectul variabilelor de graf asupra explicațiilor



Sursa: Graful tranzacțiilor Bitcoin Elliptic (Weber et al., 2019). Date: estimări proprii; antrenare pe pașii 1-34, pașii de timp de test 35-49.

- **Masa importanței**: primele 10 variabile ale pădurii au 58,2 % din importanță
  - Cu reprezentările GCN păstrează 7,4 %, reprezentările iau 73,9 %
- **Stabilitatea pe 3 reantrenări**: corelația Spearman medie a importanțelor 0,973 (pădurea) și 0,896 (pădure + GCN)

- **În practică:** comparați explicațiile între reantrenări înainte de a interpreta o schimbare de model ca o schimbare de explicație.

NN_L9_importance_shift

Cuprins

- Grafurile în detectarea infracțiunilor financiare
- Grafurile ca matrice
- Transmiterea mesajelor
- GCN, GraphSAGE și GAT
- Studiu de caz: tranzacții Bitcoin ilicite
- Rețelele GNN în producție
- Concluzii

Erori frecvente în aplicarea modelelor pe graf în finanțe

- **Scurgeri de informație prin muchii**
  - Un nod de test nu trebuie să primească mesaje din viitor
  - În Elliptic 100,0 % dintre muchii rămîn în același pas de timp, deci împărțirea în timp este curată
- **Puține etichete**
  - Doar 22,9 % dintre noduri sînt etichetate; etichetele vin din investigații anterioare și favorizează cazurile deja găsite
- **Schimbarea în timp**
  - Infractorii se adaptează: închiderea din pasul 43 a schimbat tiparele
  - Rețelele GNN temporale (EvolveGCN) lasă ponderile să evolueze în timp
- **Explicabilitatea**
  - Un raport de activitate suspectă are nevoie de o justificare pe care un supraveghetor o poate citi
  - GNNExplainer găsește subgraful și variabilele din spatele unei predicții

- **În practică:** testați un model pe graf așa cum va funcționa: pe perioade ulterioare, cu etichetele disponibile atunci.

(Pareja et al., 2020; Ying et al., 2019)

Aplicații ale modelelor bazate pe grafuri în finanțe

- **Plăți și carduri**
  - Grafuri deținător–comerciant: un terminal compromis leagă multe victime
- **Credit**
  - Rețele de garanți și furnizori: neplata se propagă prin creditul comercial
- **Cripto**
  - Grafuri de adrese și tranzacții (acest curs), detectarea serviciilor de amestecare, gruparea portofelelor
- **Risc sistemic**
  - Expuneri interbancare și rețele de dependență în cozi: cine cade dacă o bancă cade
- **Piețe**
  - Grafuri de acțiuni legate prin corelație, lanț de aprovizionare sau acționariat, pentru prognoza randamentelor
- **Energie**
  - Rețelele electrice și zonele de preț interconectate sînt grafuri prin construcție

În practică: ori de cîte ori riscul unei unități depinde de contrapărțile ei, un model pe graf merită încercat.

(Hamilton, 2020; Bishop & Bishop, 2024, ch. 13)

Cuprins

- Grafurile în detectarea infracțiunilor financiare
- Grafurile ca matrice
- Transmiterea mesajelor
- GCN, GraphSAGE și GAT
- Studiu de caz: tranzacții Bitcoin ilicite
- Rețelele GNN în producție
- Concluzii

Idei principale

- **Un strat GNN** = mesaj, agregare care nu depinde de ordine, actualizare
  - GCN face media cu ponderi după grad, GraphSAGE separă nodul de vecini, GAT învață ponderile
- **Adîncimea** = numărul de salturi: două straturi ajung la o mediană de 11 tranzacții
  - Mai multe straturi netezesc excesiv (PR-AUC 0,371 cu 6 straturi)
- **Pe Elliptic** muchiile ajută o rețea (GCN față de MLP: +8,1 puncte PR-AUC), dar o pădure aleatoare pe variabile deja agregate conduce (F1 0,808)
- **Judecați modelele AML** pe perioade ulterioare, cu PR-AUC, bugete de alerte și intervale bootstrap
  - Urmăriți schimbarea în timp și stabilitatea explicațiilor

Cercetările grupului nostru: grafuri, fraudă și explicații

- **Stabilitatea explicațiilor în detectarea fraudei cu variabile de graf** (Diaconu & Pele, 2026)
  - Lucrarea tratează reantrenarea completă a modelului drept unitate de inferență și compară explicațiile SHapley Additive exPlanations (SHAP) ale modelelor de fraudă cu și fără reprezentări de graf, față de controale fără graf, cu aceeași dimensiune și covarianță
- **Rezultate**: adăugarea de variabile poate micșora mecanic atribuirile variabilelor originale
  - După ce se ține cont de variabilitatea reantrenării, controalele fără graf reproduc cea mai mare parte a variației stabilității SHAP marginale, iar ordonările locale, la nivel de rînd, se schimbă
- **Energy Financial Risk Meter** (Toma, Dima, Grecu, Pele & Lessmann, 2026): măsoară riscul sistemic din sectorul energetic al Uniunii Europene pe baza rețelei de dependențe în cozi dintre firme

Rezultatele de pe acest slide sînt cele din lucrări; cifrele de pe celelalte slide-uri provin din calculele noastre.

(Academia de Studii Economice din București; Academia Română, Institutul de Prognoză Economică)

Lecturi și resurse suplimentare

- **Bishop & Bishop (2024)**, capitolul 13: rețele neuronale pe grafuri, de la transmiterea mesajelor la atenția pe graf
- **Hamilton (2020)**, *Graph Representation Learning*, versiune gratuită: cs.mcgill.ca/~wlh/grl_book.
- **PyTorch Geometric**, tutoriale: pytorch-geometric.readthedocs.io.
- **Setul Elliptic** și lucrarea originală: arXiv:1908.02591.
- **Acest curs**: cod, date, notebook-uri — github.com/danpele/NEURAL_BIZ.
- **Exerciții practic**: notebook-ul reproduce toate graficele și cifrele acestui curs

Test 1: permutări

- Renumerotăm tranzacțiile grafului.
  - Ce trebuie să se întîmple cu scorurile GCN?

Răspuns

- Se permută la fel și altfel nu se schimbă: un strat GNN este echivariant la permutări, pentru că agregarea pe vecini nu depinde de ordine

Test 2: adîncimea

- De ce nu folosim zece straturi GCN, ca să vedem zece salturi ale fluxului de bani?

Răspuns

- Medierea repetată face toate reprezentările asemănătoare (over-smoothing): pe Elliptic, PR-AUC scade de la 0,458 (2 straturi) la 0,371 (6 straturi)

Test 3: modelul de referință puternic

- De ce pădurea aleatoare fără graf bate rețeaua GCN pe Elliptic?

Răspuns

- 72 dintre variabile sînt deja agregate pe vecini, un strat de transmitere a mesajelor construit manual
  - În plus, arborii prind bine pragurile de tip regulă ale fraudei

Test 4: evaluarea

- Doar 6,50 % dintre tranzacțiile de test sînt ilicite.
  - Ce măsură l-ar induce în eroare pe un ofițer de conformitate?

Răspuns

- Acuratețea: dacă etichetăm totul ca licit avem dreptate în majoritatea cazurilor și nu găsim niciun infractor
  - Folosim precizia, recall-ul, PR-AUC și bugetul de alerte

Test 5: atenția pe grafuri

- Care este legătura dintre un strat GAT și transformerul din cursul 7?

Răspuns

- Ambele ponderează vecinii cu un softmax al unor scoruri învățate
  - Transformerul aplică atenția pe graful complet al tokenilor, GAT o restrînge la muchiile grafului

Bibliografie (1/2)

- Bishop, C. M. & Bishop, H. (2024). *Deep Learning: Foundations and Concepts*, ch. 13. Springer. doi:10.1007/978-3-031-45468-4
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. doi:10.1023/A:1010933404324
- Diaconu, D. & Pele, D. T. (2026). SHAP stability across random and forward-time validation in graph-augmented fraud detection. Submitted to *Expert Systems with Applications*. Code: github.com/QuantLet/GNN_XAI_Stability
- Efron, B. & Tibshirani, R. J. (1993). *An Introduction to the Bootstrap*. Chapman & Hall. doi:10.1201/9780429246593
- Fey, M. & Lenssen, J. E. (2019). Fast graph representation learning with PyTorch Geometric. ICLR Workshop on Representation Learning on Graphs and Manifolds. arXiv:1903.02428
- Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O. & Dahl, G. E. (2017). Neural message passing for quantum chemistry. *Proceedings of ICML*, PMLR 70, 1263–1272. arXiv:1704.01212
- Hamilton, W. L., Ying, R. & Leskovec, J. (2017). Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems*, 30. arXiv:1706.02216
- Hamilton, W. L. (2020). *Graph Representation Learning*. Morgan & Claypool. doi:10.2200/S01045ED1V01Y202009AIM046
- Kipf, T. N. & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations*. arXiv:1609.02907

Bibliografie (2/2)

- Li, Q., Han, Z. & Wu, X.-M. (2018). Deeper insights into graph convolutional networks for semi-supervised learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1). doi:10.1609/aaai.v32i1.11604
- Lundberg, S. M. & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30. arXiv:1705.07874
- Pareja, A., Domeniconi, G., Chen, J. et al. (2020). EvolveGCN: Evolving graph convolutional networks for dynamic graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(4), 5363–5370. doi:10.1609/aaai.v34i04.5984
- Toma, F.-M., Dima, D.-M., Grecu, R., Pele, D. T. & Lessmann, S. (2026). Energy Financial Risk Meter: Systemic risk in the EU energy sector. SSRN Working Paper 6985301. doi:10.2139/ssrn.6985301
- Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P. & Bengio, Y. (2018). Graph attention networks. *International Conference on Learning Representations*. arXiv:1710.10903
- Weber, M., Domeniconi, G., Chen, J., Weidele, D. K. I., Bellei, C., Robinson, T. & Leiserson, C. E. (2019). Anti-money laundering in Bitcoin: Experimenting with graph convolutional networks for financial forensics. KDD Workshop on Anomaly Detection in Finance. arXiv:1908.02591
- Xu, K., Hu, W., Leskovec, J. & Jegelka, S. (2019). How powerful are graph neural networks? *International Conference on Learning Representations*. arXiv:1810.00826
- Ying, R., Bourgeois, D., You, J., Zitnik, M. & Leskovec, J. (2019). GNNExplainer: Generating explanations for graph neural networks. *Advances in Neural Information Processing Systems*, 32. arXiv:1903.03894

Anexa A: de la filtrele spectrale la stratul GCN

$$g_\theta \star \mathbf{x} \approx \sum_{k=0}^{K} \theta_k T_k(\tilde{L})\,\mathbf{x}, \quad \tilde{L} = \tfrac{2}{\lambda_{\max}} L_{sym} - I$$

$$K = 1,\ \lambda_{\max} \approx 2: \quad g_\theta \star \mathbf{x} \approx \theta_0 \mathbf{x} - \theta_1 D^{-1/2} A D^{-1/2} \mathbf{x}$$

$$\theta = \theta_0 = -\theta_1: \quad \theta\,(I + D^{-1/2} A D^{-1/2})\,\mathbf{x} \ \rightarrow\ \theta\,\hat{A}\,\mathbf{x}$$

- T_k = polinoamele Cebîșev
  - $L_{sym} = I - D^{-1/2} A D^{-1/2}$
  - Un polinom de grad K amestecă doar noduri aflate la cel mult K salturi
- Un singur parametru comun evită supraînvățarea
  - Trucul de renormalizare $I + D^{-1/2} A D^{-1/2} \rightarrow \tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2}$ ține valorile proprii în $(-1, 1]$
- Cu d variabile de intrare și d' ieșiri, θ devine matricea $W \in \mathbb{R}^{d \times d'}$

(Kipf & Welling, 2017)

Anexa B: testul Weisfeiler–Leman

$$c_v^{(l+1)} = \text{hash}\left(c_v^{(l)},\ \{\{c_u^{(l)} : u \in \mathcal{N}(v)\}\}\right)$$

- Se pornește cu aceeași culoare pentru toate nodurile
  - Se rafinează prin aplicarea unei funcții hash culorii nodului și multisetului de culori ale vecinilor
- Două grafuri cu histograme de culori diferite nu sînt izomorfe
  - Histogramele egale nu decid
- O rețea GNN cu transmitere de mesaje deosebește cel mult cît testul WL
  - GIN atinge această limită cu agregare prin sumă și o actualizare injectivă

(Xu et al., 2019)

Anexa C: setările experimentului

Setare	Valoare
Împărțirea	antrenare pașii 1–29, validare 30–34, test 35–49 (numai nodurile etichetate intră în funcția de pierdere)
Rețele	2 straturi, 100 de unități, ReLU, dropout 0,2; GAT: 4 capete a cîte 25; Adam, rata 0,003, penalizare $5 \cdot 10^{-4}$; cel mult 300 de epoci, răbdare 40
Ponderi de clasă	0,3 / 0,7
Pădure aleatoare	100 de arbori, 50 de variabile la fiecare împărțire, variabile nestandardizate
Semințe	1, 2, 3
Programe, echipament	PyTorch, PyTorch Geometric, scikit-learn; procesorul unui laptop, în total 0,1 minute

(Weber et al., 2019; Fey & Lenssen, 2019)