Anti-Money Laundering

Graph Convolutional Networks for Financial Forensics

Priyabrata Senapati

CS886: Deep Learning for NLP

Outline:

- ➢Motivation
- ➢Background
- ➢Convolutional Neural Network
- ➢Graph Convolutional Neural Network
- ➢Graph Convolution Neural Networks for Financial Forensics

Motivation

- Financial transaction data is generally unstructured in nature. This makes most deep learning methods difficult to learn due to the nonlinearity of the data.

- Graph neural networks are effective framework for representation learning for graphs from both structured and unstructured data.

Background

$$z = w \cdot x + b$$

Where:

z = weighted sum of inputs

x = input vector

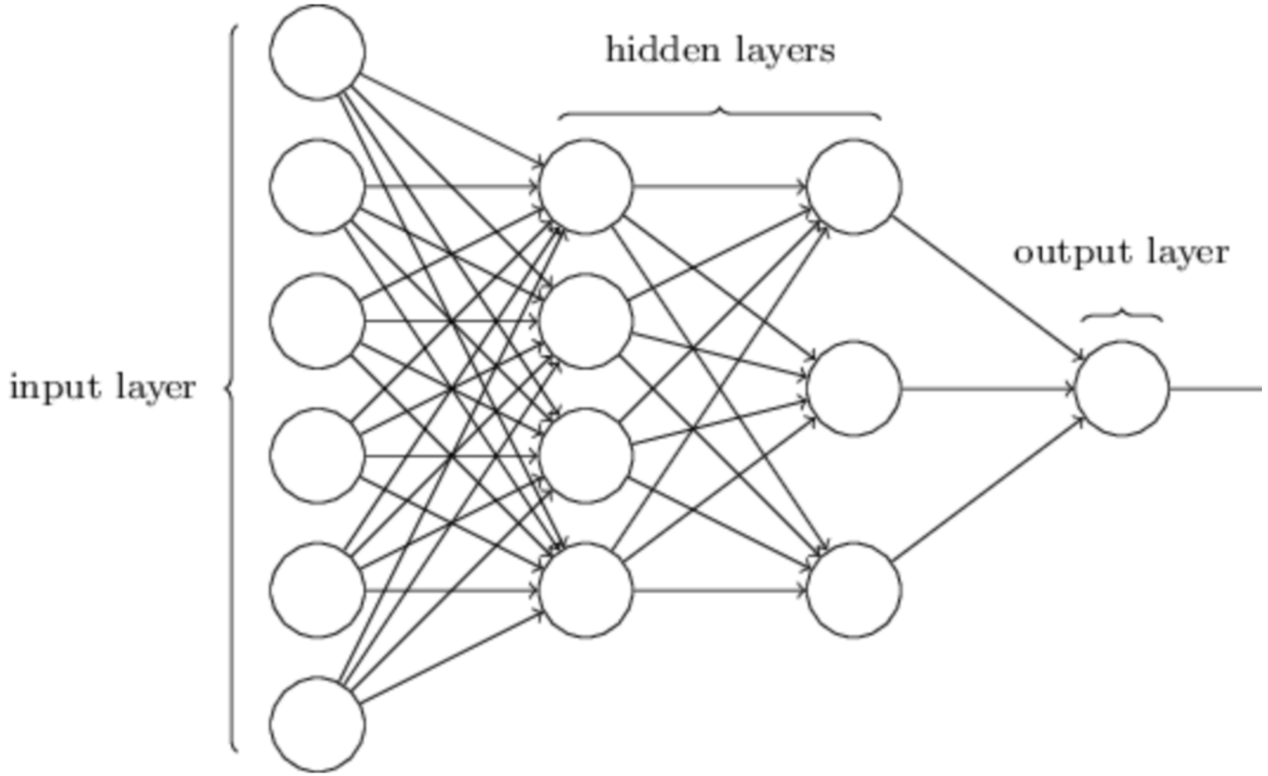
b = bias

$$y = f(z)$$

Where:

y = output

f = activation function (ex. Sigmoid, Tanh, ReLU)



Convolutional Neural Network

What We See

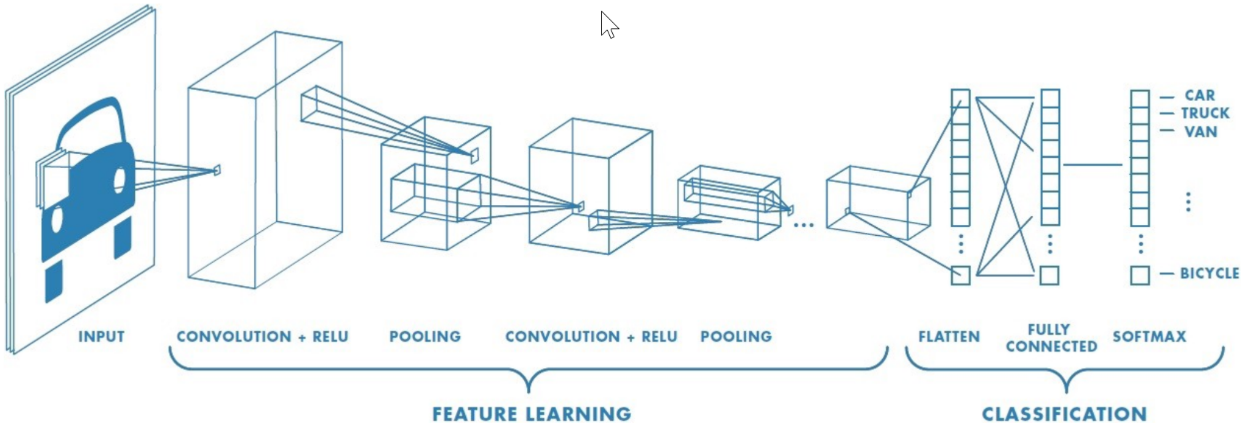
08	02	22	97	38	15	00	40	00	75	04	05	07	78	52	12	50	77	91	08
49	49	99	40	17	81	18	57	60	87	17	40	98	43	69	48	04	56	62	00
81	49	31	73	55	79	14	29	93	71	40	67	53	88	30	03	49	13	36	65
52	70	95	23	04	60	11	42	69	24	68	56	01	32	56	71	37	02	36	91
22	31	16	71	51	67	63	89	41	92	36	54	22	40	40	28	66	33	13	80
24	47	32	60	99	03	45	02	44	75	33	53	78	36	84	20	35	17	12	50
32	98	81	28	64	23	67	10	26	38	40	67	59	54	70	66	18	38	64	70
67	26	20	68	02	62	12	20	95	63	94	39	63	08	40	91	66	49	94	21
24	55	58	05	66	73	99	26	97	17	78	78	96	83	14	88	34	89	63	72
21	36	23	09	75	00	76	44	20	45	35	14	00	61	33	97	34	31	33	95
78	17	53	28	22	75	31	67	15	94	03	80	04	62	16	14	09	53	56	92
16	39	05	42	96	35	31	47	55	58	88	24	00	17	54	24	36	29	85	57
86	56	00	48	35	71	89	07	05	44	44	37	44	60	21	58	51	54	17	58
19	80	81	68	05	94	47	69	28	73	92	13	86	52	17	77	04	89	55	40
04	52	08	83	97	35	99	16	07	97	57	32	16	26	26	79	33	27	98	66
88	36	68	87	57	62	20	72	03	46	33	67	46	55	12	32	63	93	53	69
04	42	16	73	38	25	39	11	24	94	72	18	08	46	29	32	40	62	76	36
20	69	36	41	72	30	23	88	34	62	99	69	82	67	59	85	74	04	36	16
20	73	35	29	78	31	90	01	74	31	49	71	48	86	81	16	23	57	05	54
01	70	54	71	83	51	54	69	16	92	33	48	61	43	52	01	89	19	67	48

What Computers See

Convolutional Neural Network

- Neural Networks are good at image recognition
- When we have a large pixel image, neural networks are slow and consume more computational power
- Eg., a 256*256*3 size image requires 196,608 input parameters whereas an image of size of 2000*2000*3 requires 12 million input parameters.

Convolutional Neural Network



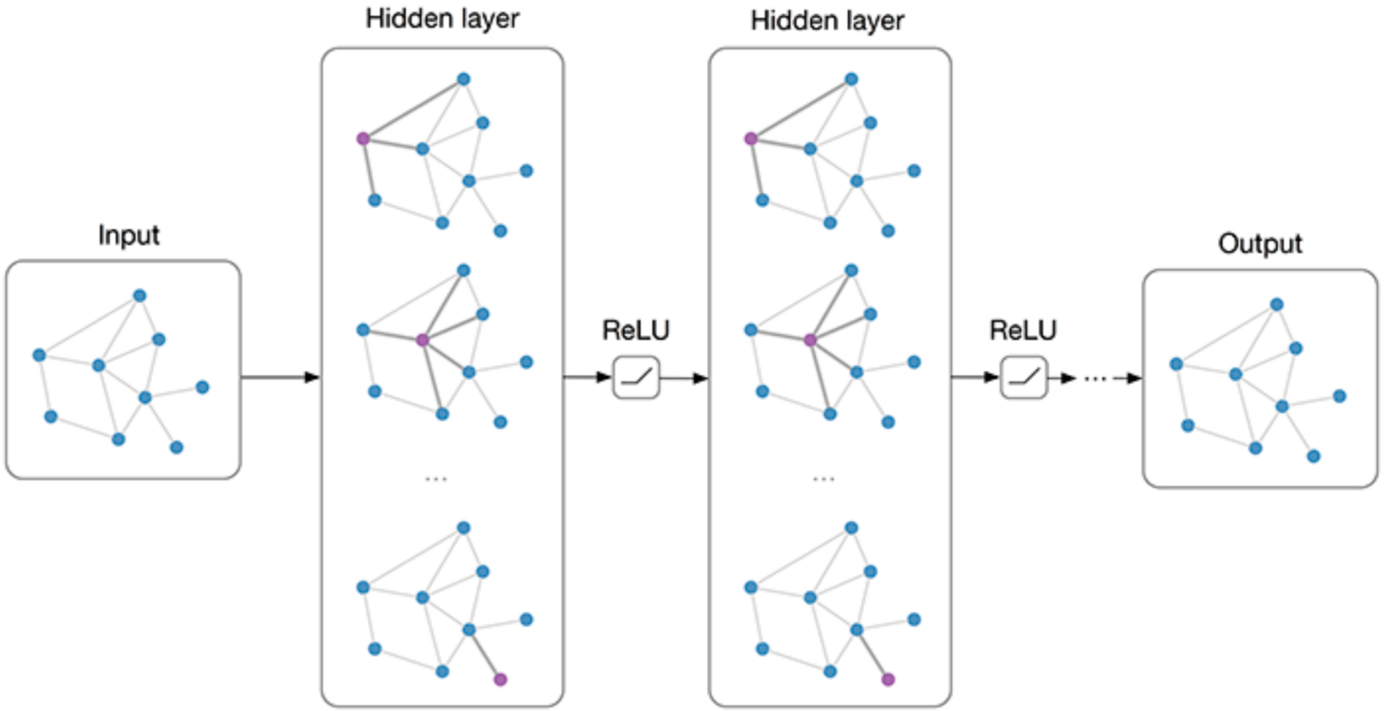
Feature Learning:

- Convolution
- Filter
- Convolved Layer
- ReLU
- Pooling Layer

Classification

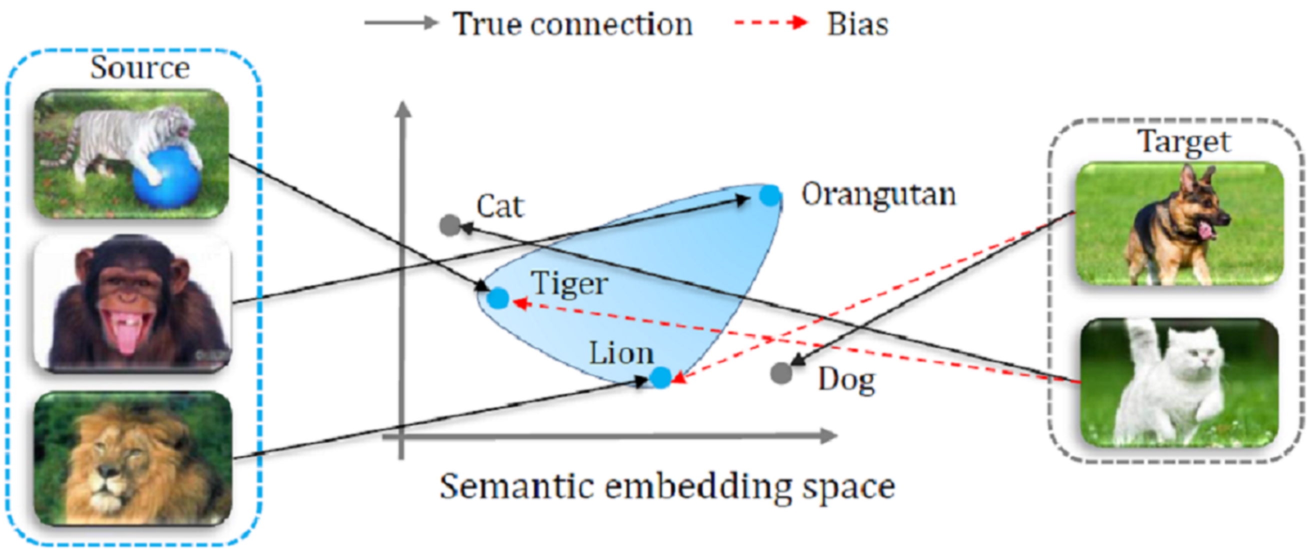
- Fully Connected Layer

Why Graph Neural Networks (GNN)?



Multi-layer Graph Convolutional Network (GCN) with first-order filters.

Why Graph Neural Networks?



- GNNs are connectionist models that capture the dependence of graphs through signals passing between nodes of graphs.
- They retain state information to capture the properties of neighbor nodes.
- Application in NLP: Graphs also incorporate semantic information from word embeddings of the category labels of the images. Non structured data including images and texts can also be modelled in the form of graphs.

Graph Convolutional Neural (GCN) Network

Definition

- In GCN, our goal is to learn a function of signals/features on a graph $G = (V, E)$, where V is the set of vertices (or nodes) of G and E is the number of edges of G.
- x_i is the feature description for all node i in a $N \times D$ matrix X, where N is number of nodes and D is input features.
- We can represent a graph in a matrix form, like an adjacency matrix A.

Graph Convolutional Neural Network

- The GCN model produces a node level output matrix Z of $N \times F$ dimension, where F is the number of output features per node. The outputs can be modeled by introducing pooling operations.
- Neural network layers can be written as a non-linear function $H^{(l+1)} = f(H^{(l)}, A)$, $H^{(0)} = X$ and $H^{(L)} = Z$ (or z for graph level outputs), where L is number of layers.

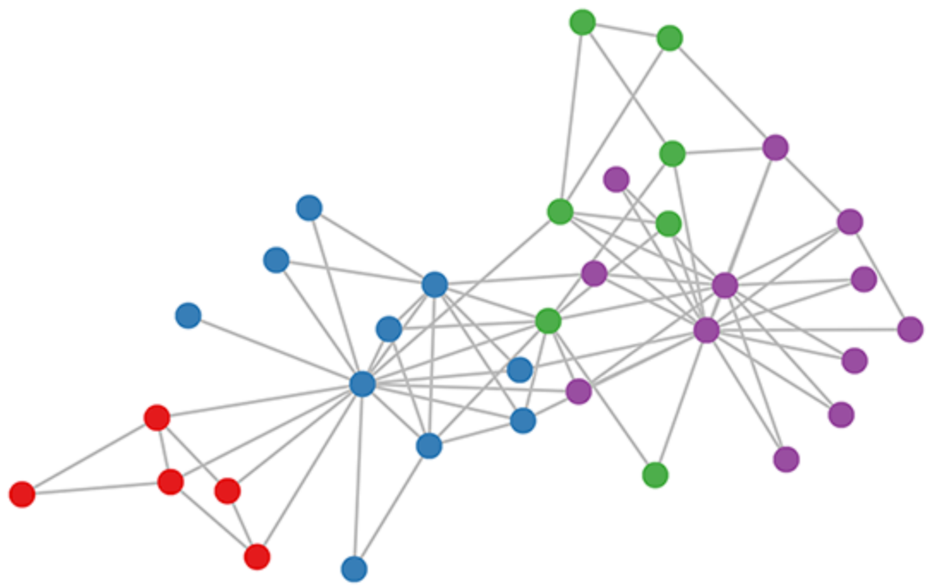
Example:

- $f\left(H^{(l)}, A\right) = \sigma(AH^{(l)}W^{(l)})$, $W^{(l)}$ is a weight matrix for lth neural network layer. σ is a nonlinear activation function like ReLU.

Graph Convolutional Neural Network

Embedding Karate Club Network (Brandes et al., 2008)

- 3 layer GCN with randomly initialized weights. $X = I$ since we don't have any node features.
- 3-layer GCN performs propagation steps through forward pass. The model produces embedding of these nodes that resembles the actual graph structure.
- Here embedding means projecting an input into another more convenient representation space.



Graph Convolutional Neural Network for Financial Forensics

- Hundreds of billions in money laundering funds, drug cartels, human trafficking, terrorism and political corruption.

- AML regulations have negative effect on low-income people.

Graph Convolutional Neural Network for Financial Forensics

- Market value collapsed in 2018 after the money laundering scandal.
- Estonia hub has $200 billion illicit money flowing from various countries.
- Besides Danske bank, Swede bank and Deutsche bank also lost huge sums of money from the scandal.
- 2018 cash flows were $529 billion where $37 billion were transaction fees which was roughly 24% of global aid of $153 billion.

Graph Convolutional Neural Network for Financial Forensics

Can bitcoin help?

- Crypto enthusiasts have championed financial inclusion. When price skyrockets the transactional fee differential diminishes. This makes cryptocurrencies like bitcoin interesting.

- It is a both good and bad. Bad because, criminals can obfuscate identities and hide in plain sight. Good because open data enables new forensic analysis and crowdsourcing not possible with traditional financial system

Graph Convolutional Neural Network for Financial Forensics

Elliptic Data Set

- London based start up Elliptic.
- They monitor suspicious activities in cryptocurrencies.
- Label suspicious address.
- Perform advance analytic to track bad elements.

The Elliptic Data Set

203,769 nodes (Bitcoin transactions)

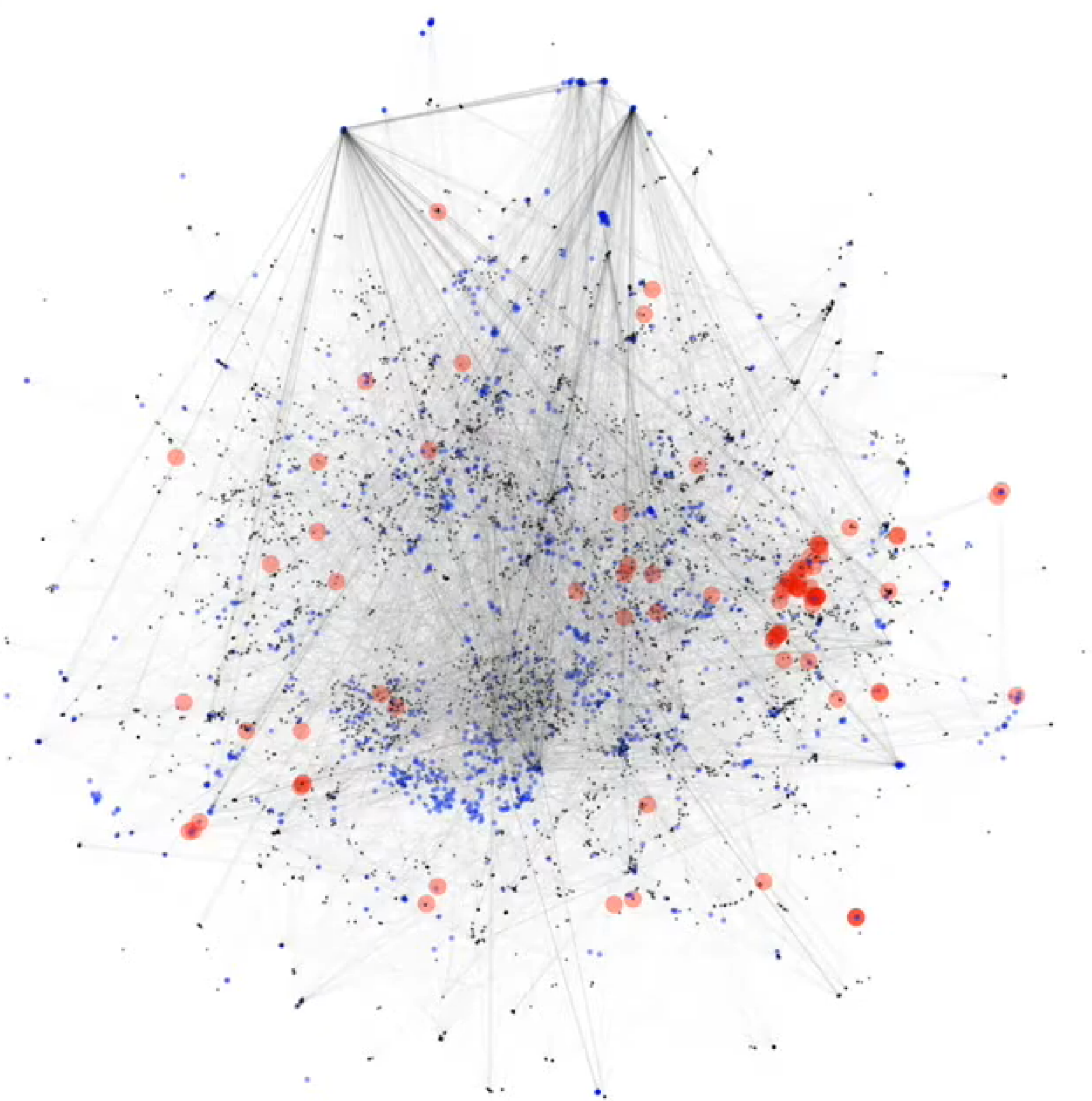
234,355 edges (directed flows)

21% licit labels (known exchanges, wallet providers, miners, licit services, etc.)

2% illicit labels (known scams, malware, terrorist organizations, ransomware, Ponzi schemes, etc.)

94 local features (LF) e.g. time step, in/out count activity, transaction fee

72 one-hop aggregate features (AF) (e.g. max, min, standard deviation, and correlation coefficients of the neighbor transactions)



Graph Convolutional Neural Network for Financial Forensics

- Consider the bitcoin transaction graph from Elliptic data set $G = (N, E)$, N is the set of node transactions and E is the set of edges representing the flow of bitcoin.
- $H^{(l+1)} = \sigma(\hat{A}H^{(l)}W^{(l)}), W^{(l)}$ is a weight matrix for lth neural network layer. σ is a nonlinear activation function like ReLU. $\hat{A}$ is normalization of A.

Graph Convolutional Neural Network for Financial Forensics

Experiments on the Elliptic Data Set

Task and Set-up

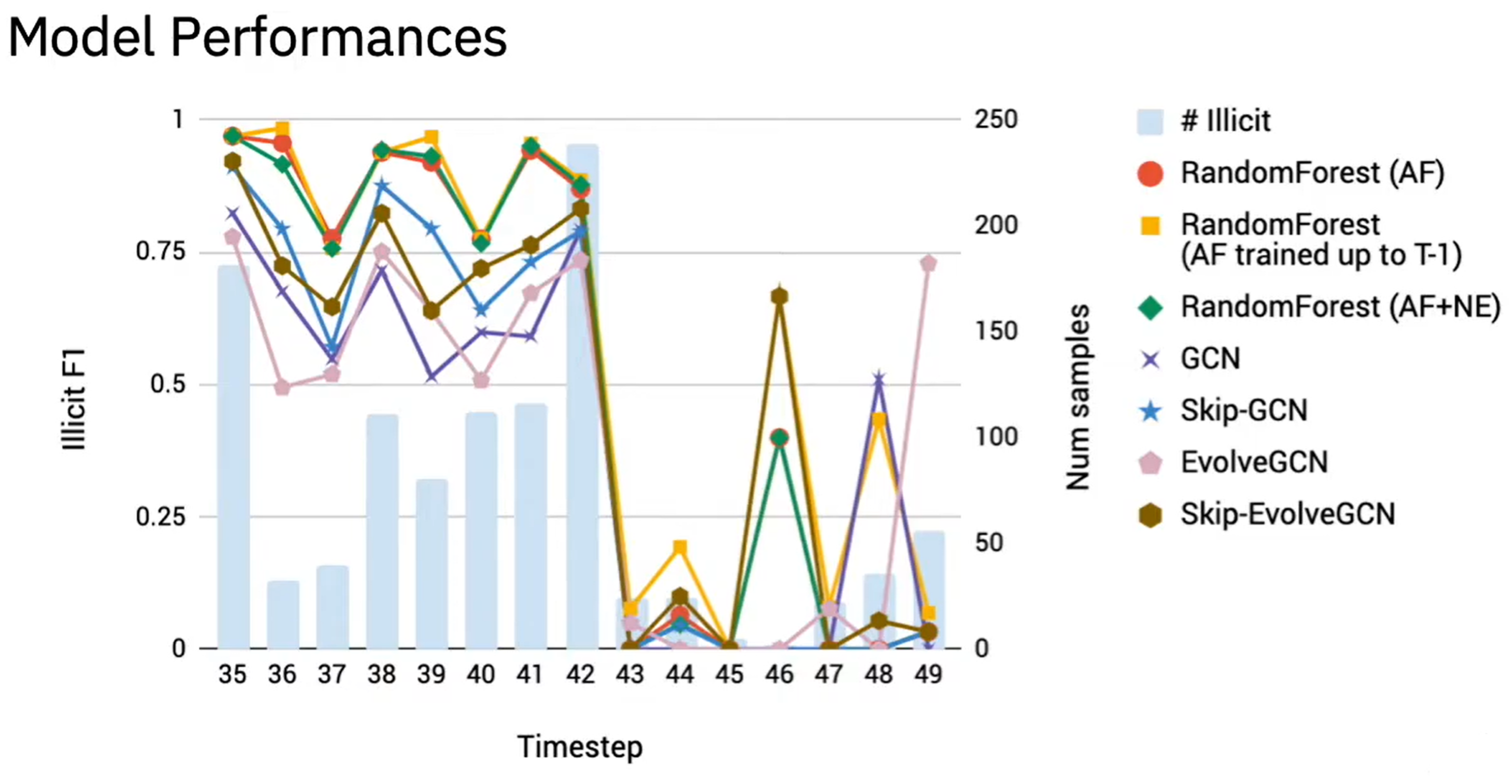
- **Binary classification task** to identify illicit node transactions
- Trained GCN using **weighted cross entropy loss** to prioritize illicit nodes
- Local Features (LF)
  Aggregate Features (AF)
  Node Embeddings (NE)

Noteworthy Results

- Red highlight: best model is Random Forest GCN hybrid
- Blue highlight: GCN provides a 7% increase in Precision to a graph agnostic Random Forest

Method	Illicit Precision	Illicit Recall	Illicit F_1	MicroAVG F_1
Logistic Regr^{AF}	0.404	0.593	0.481	0.931
Logistic Regr^{AF+NE}	0.537	0.528	0.533	0.945
Logistic Regr^{LF}	0.348	0.668	0.457	0.920
Logistic Regr^{LF+NE}	0.518	0.571	0.543	0.945
RandomForest^{AF}	0.956	0.670	0.788	0.977
RandomForest^{AF+NE}	0.971	0.675	0.796	0.978
RandomForest^{LF}	0.803	0.611	0.694	0.966
RandomForest^{LF+NE}	0.878	0.668	0.759	0.973
MLP^{AF}	0.694	0.617	0.653	0.962
MLP^{AF+NE}	0.780	0.617	0.689	0.967
MLP^{LF}	0.637	0.662	0.649	0.958
MLP^{LF+NE}	0.6819	0.5782	0.6258	0.986
GCN	0.812	0.512	0.628	0.961
Skip-GCN	0.812	0.623	0.705	0.96

Graph Convolutional Neural Network for Financial Forensics



References

- https://tkipf.github.io/graph-convolutional-networks/
- https://towardsdatascience.com/how-to-do-deep-learning-on-graphs-with-graph-convolutional-networks-7d2250723780
- https://medium.com/@sunitachoudhary103/how-to-deal-the-graphs-data-in-deep-learning-with-graph-convolutional-networks-gcn-39f69db072ee
- https://datawarrior.wordpress.com/2018/08/08/graph-convolutional-neural-network-part-i/
- https://medium.com/@purnasaigudikandula/a-beginner-intro-to-convolutional-neural-networks-684c5620c2ce
- https://arxiv.org/pdf/1901.00596.pdf
- http://persagen.com/files/misc/scarselli2009graph.pdf
- https://arxiv.org/abs/1908.02591
- https://arxiv.org/abs/1609.02907
- https://www.markrweber.com/
- https://towardsdatascience.com/https-medium-com-aishwaryajadhav-applications-of-graph-neural-networks-1420576be574

THANK YOU

Questions?