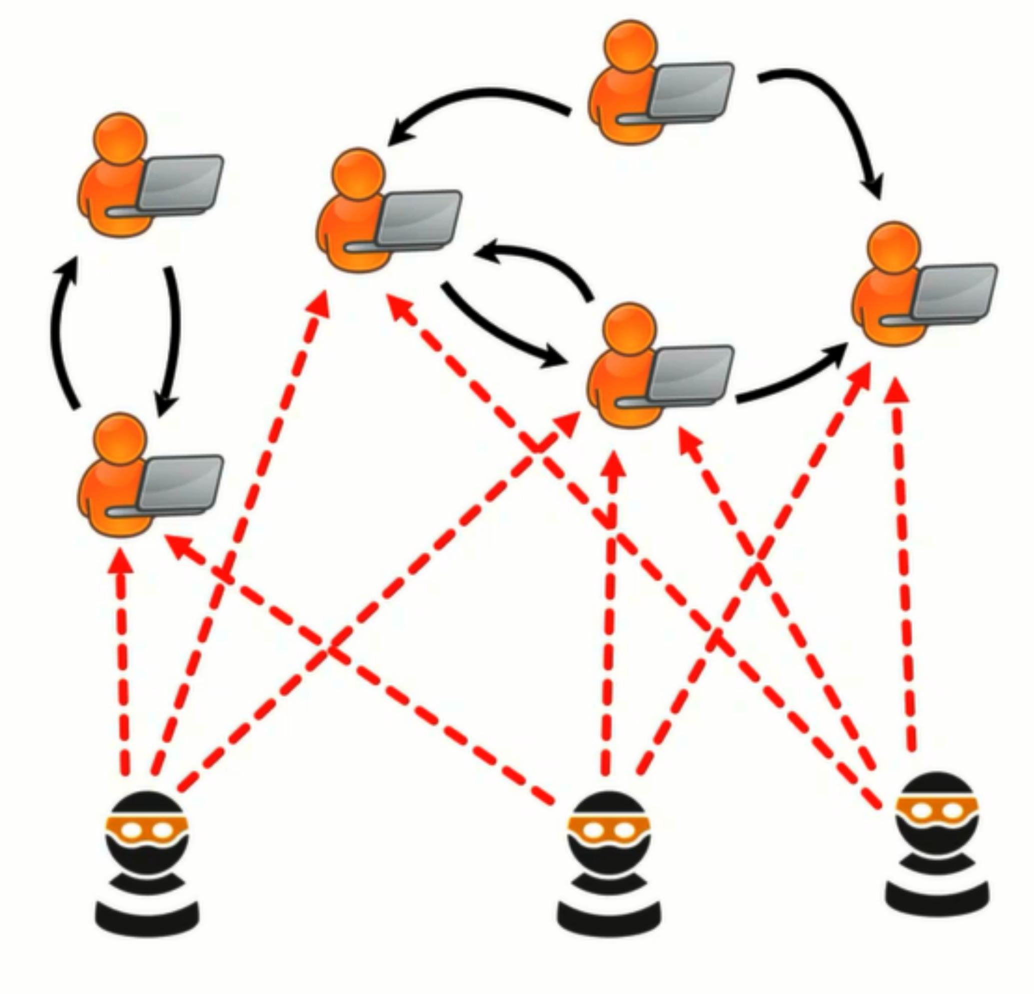
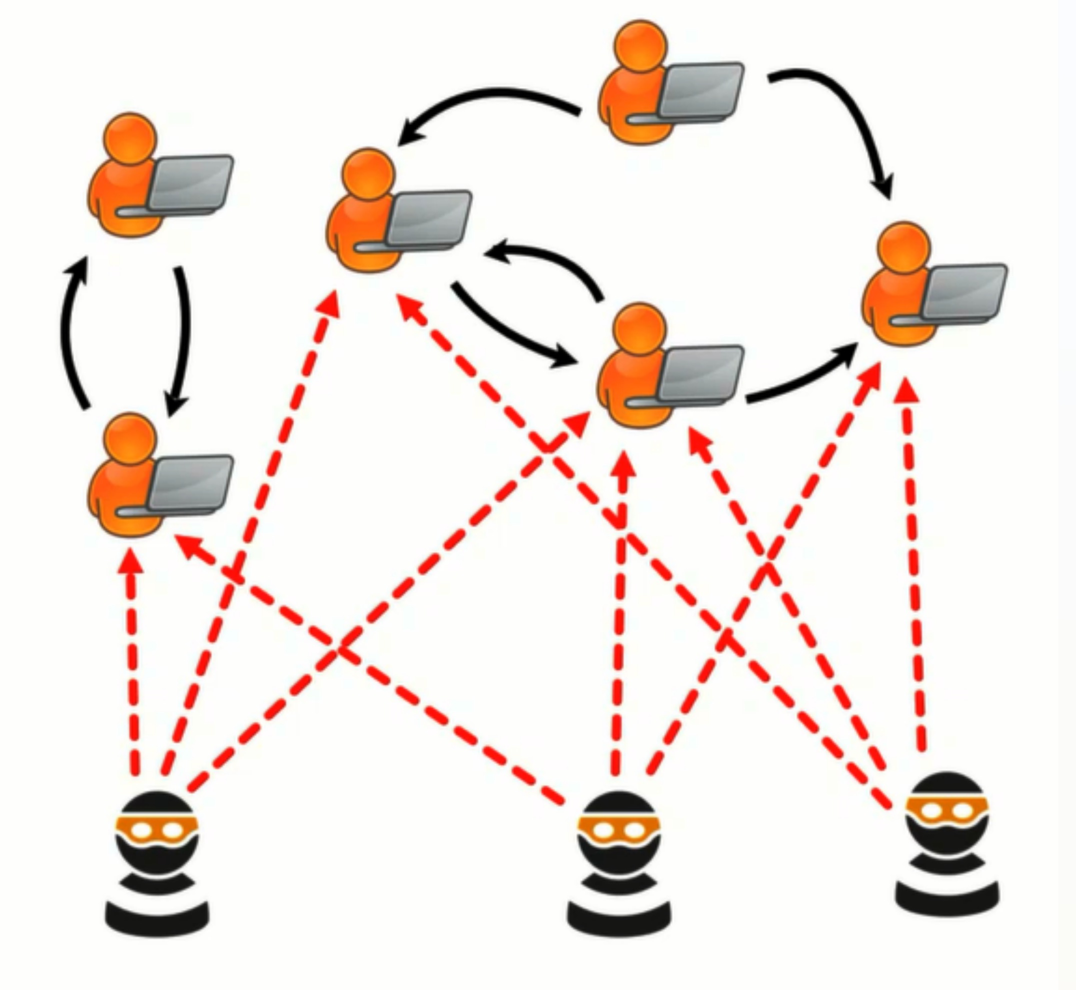


Graph Anomaly Detection with Deep Learning

Seminar Unsupervised Machine Learning

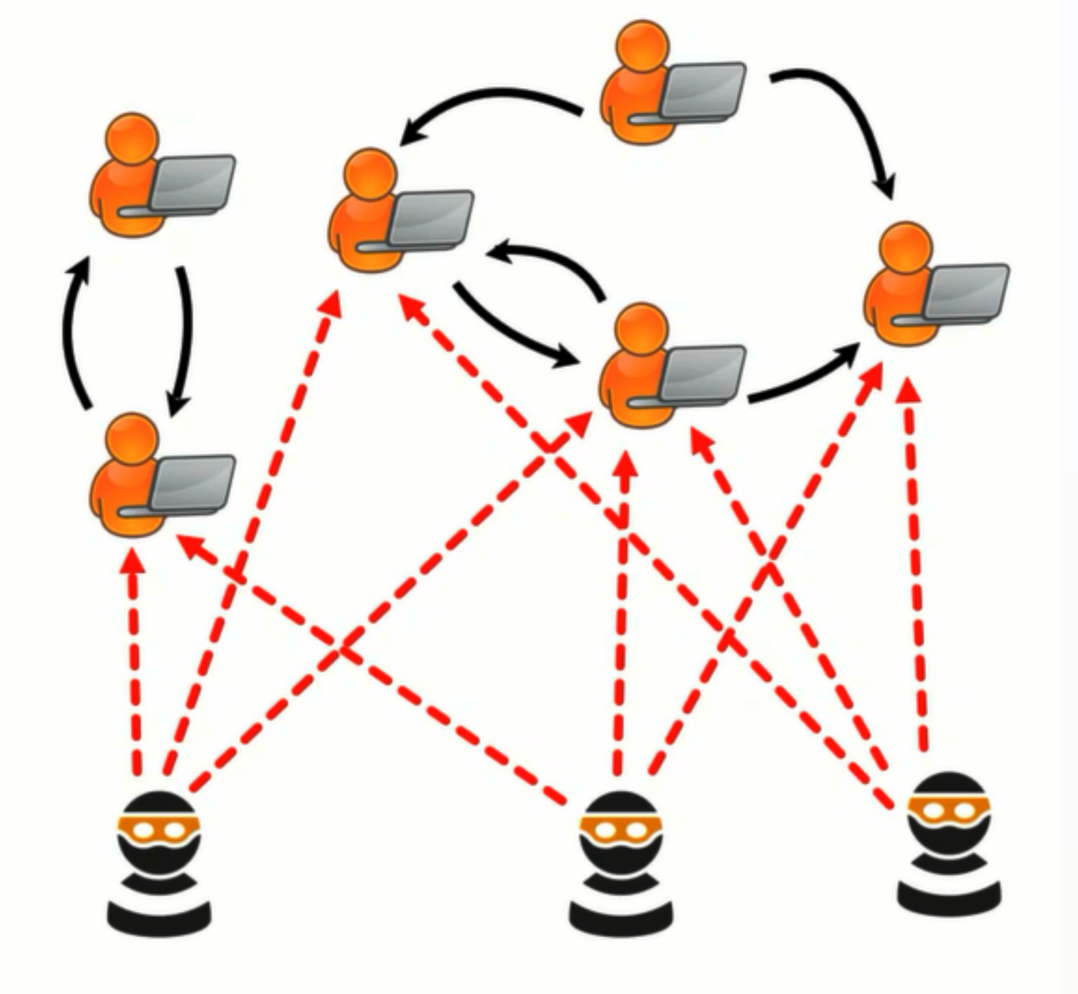
Students: Hemani Chand and Jatin Rattan
Supervisor: Simon Klüttermann

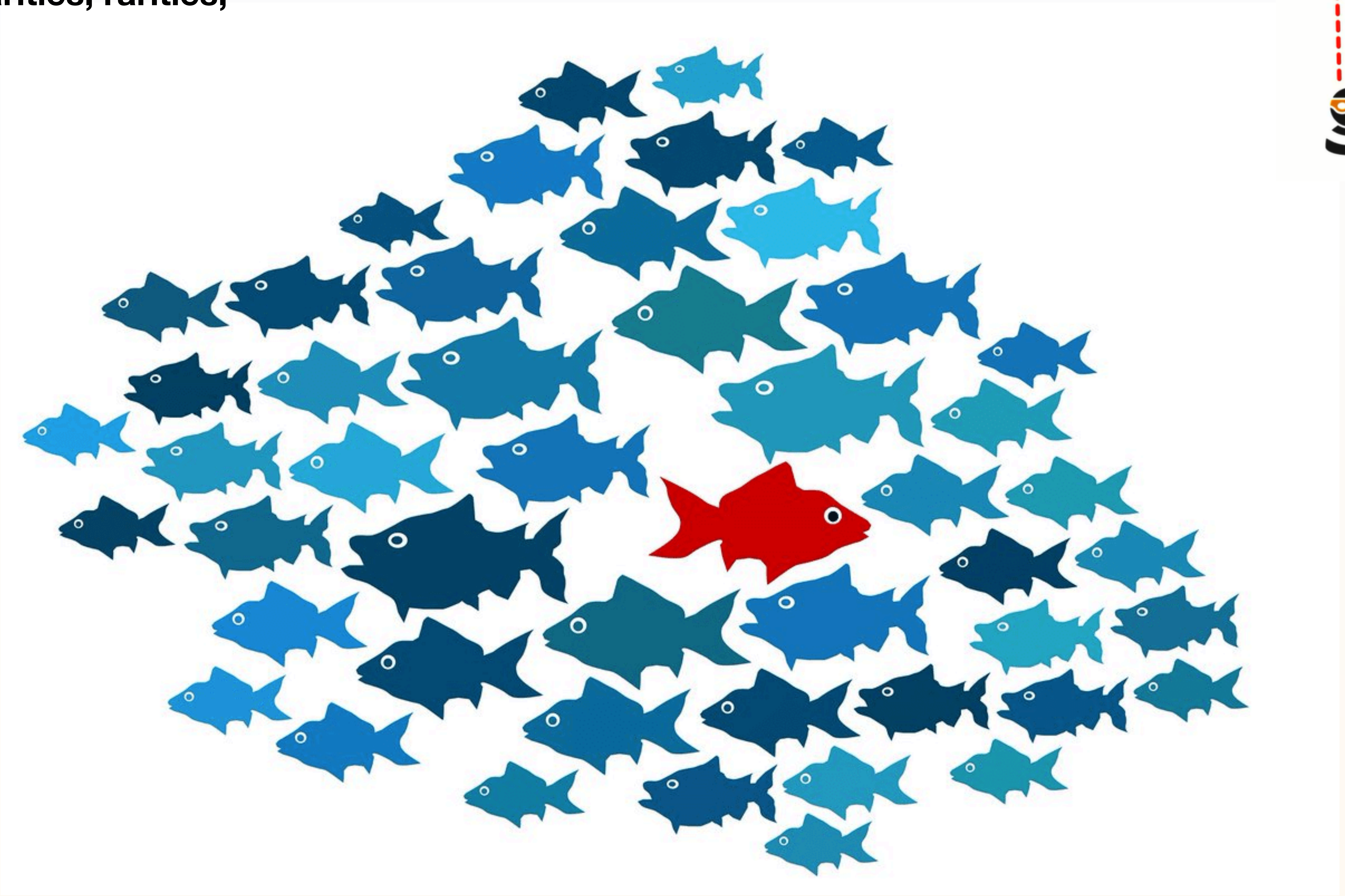
Table of Contents



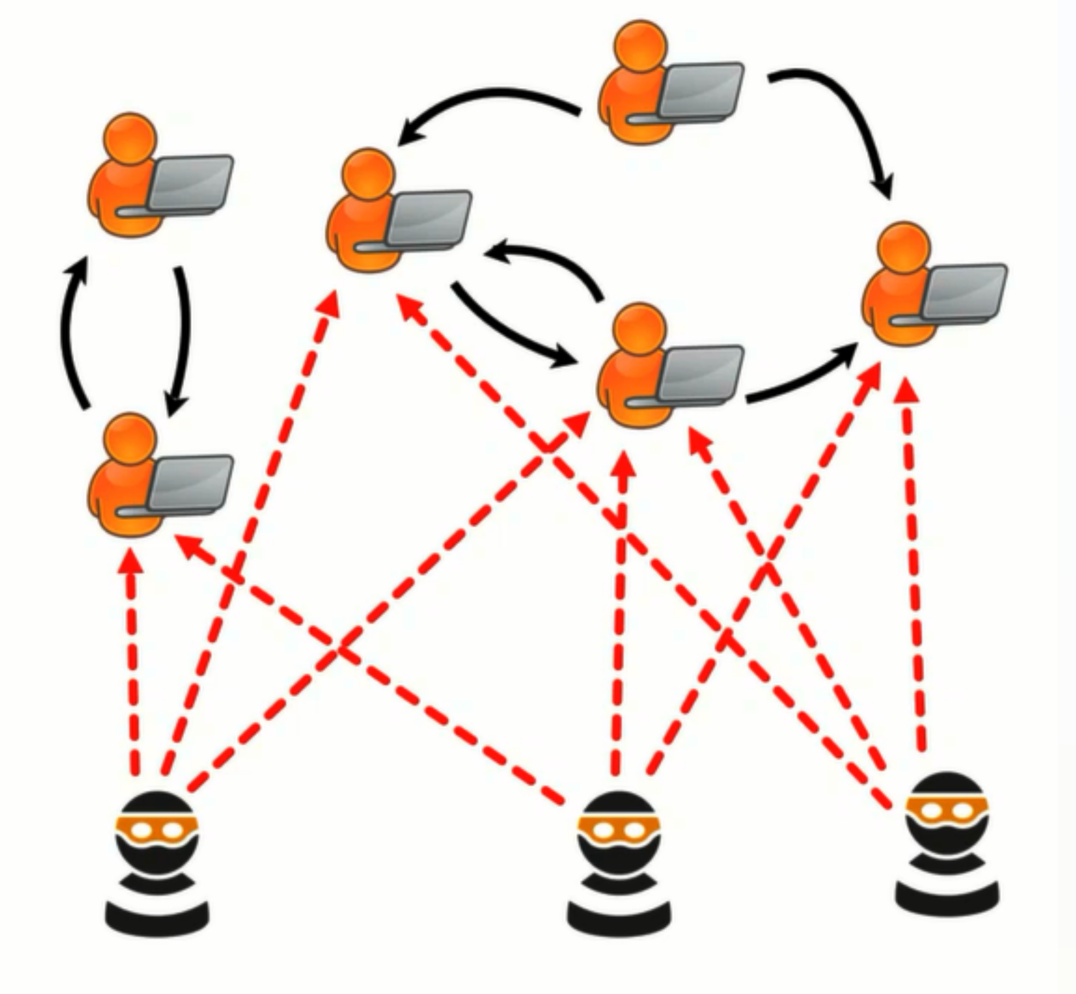
Anomaly Detection

Outliers, exceptions, peculiarities, rarities, novelties etc



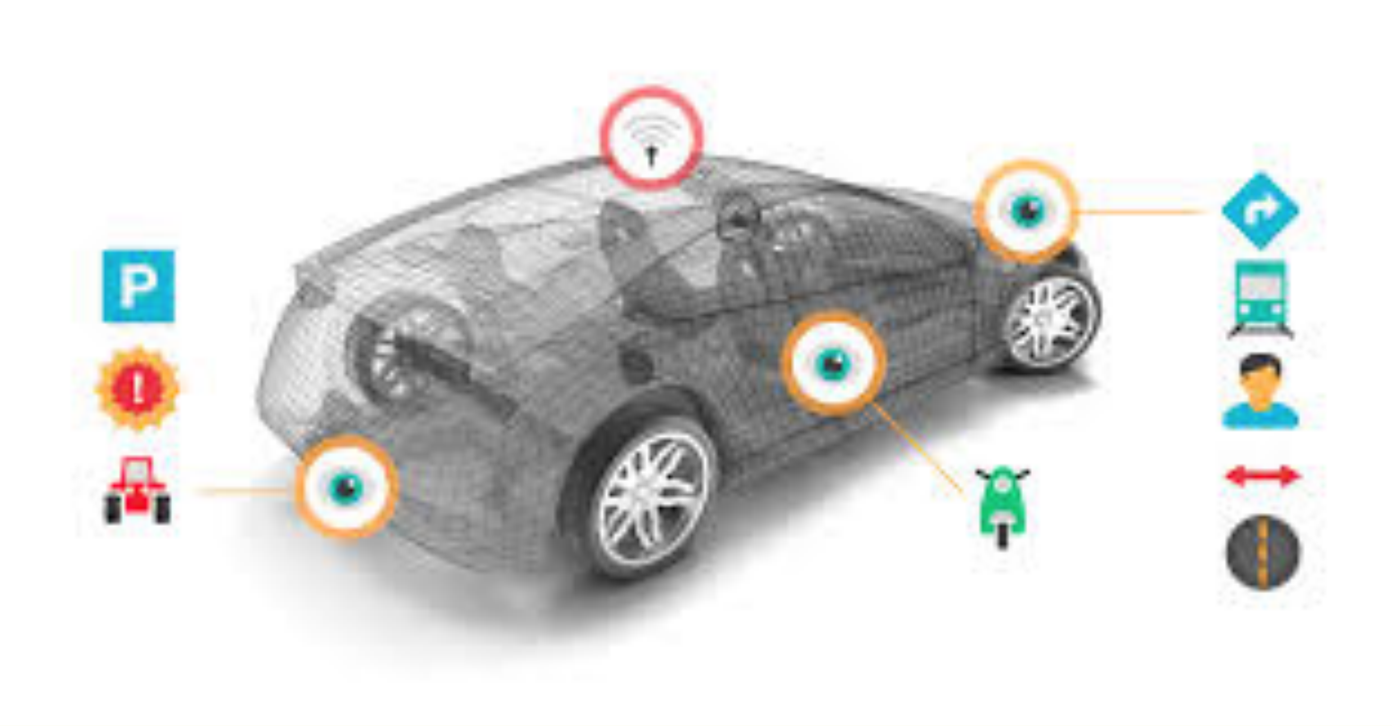


Source: https://static.tildacdn.com/tild3131-3237-4364-b662-663731666262/anomaly_detection.png



Applications

- Fraud detection

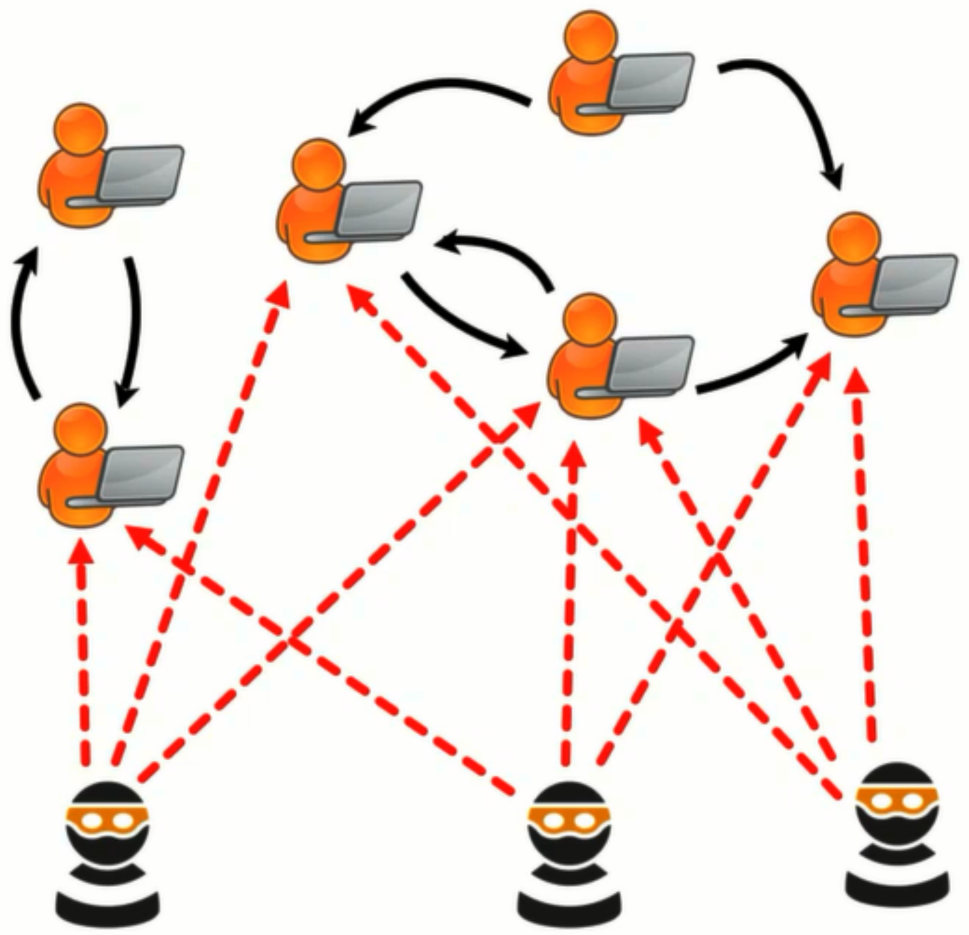


- Autonomous vehicle development

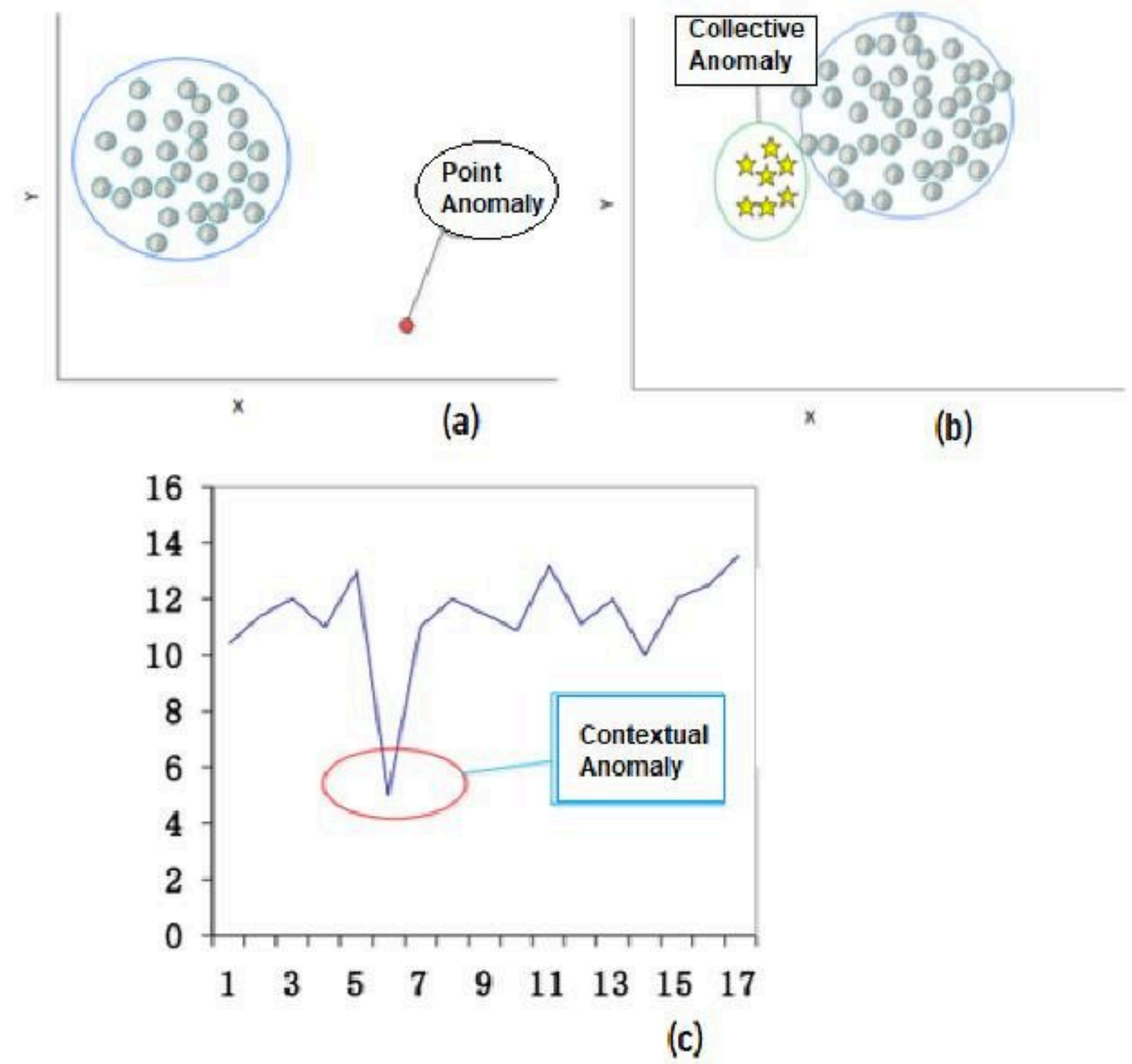
- Network intrusion detection

Source: https://miro.medium.com/max/969/1*gVEGuKzXELsr7_0zTueHfQ.png,
https://www.deciso.com/wp-content/uploads/2015/11/Intrusion-detected-hr-900px-405x360.jpg
https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTSZk9ykDoGRCb82Nz6hIozr2mw0Eqqfwcv9Q&usqp=CAU

Types of Anomalies



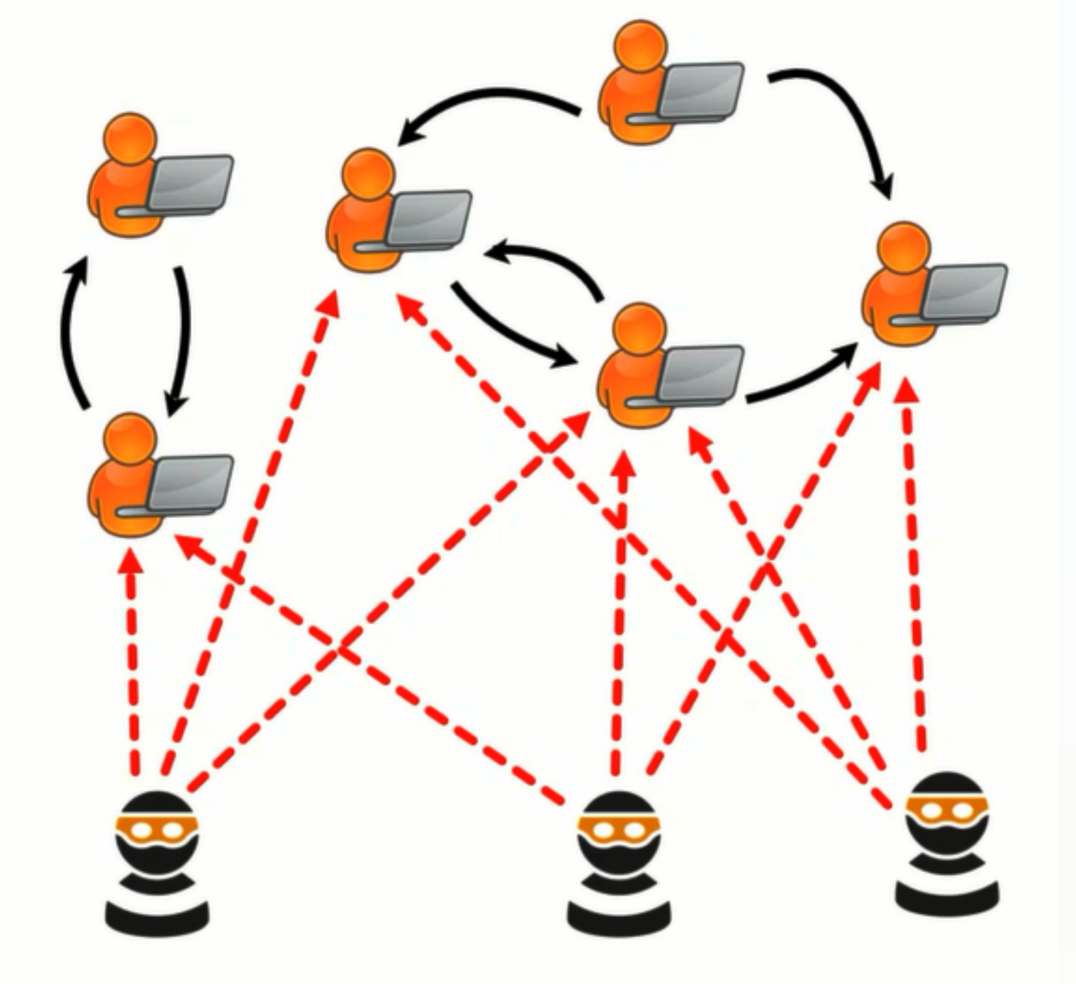
- Point anomaly
- Community anomaly
- Contextual anomaly

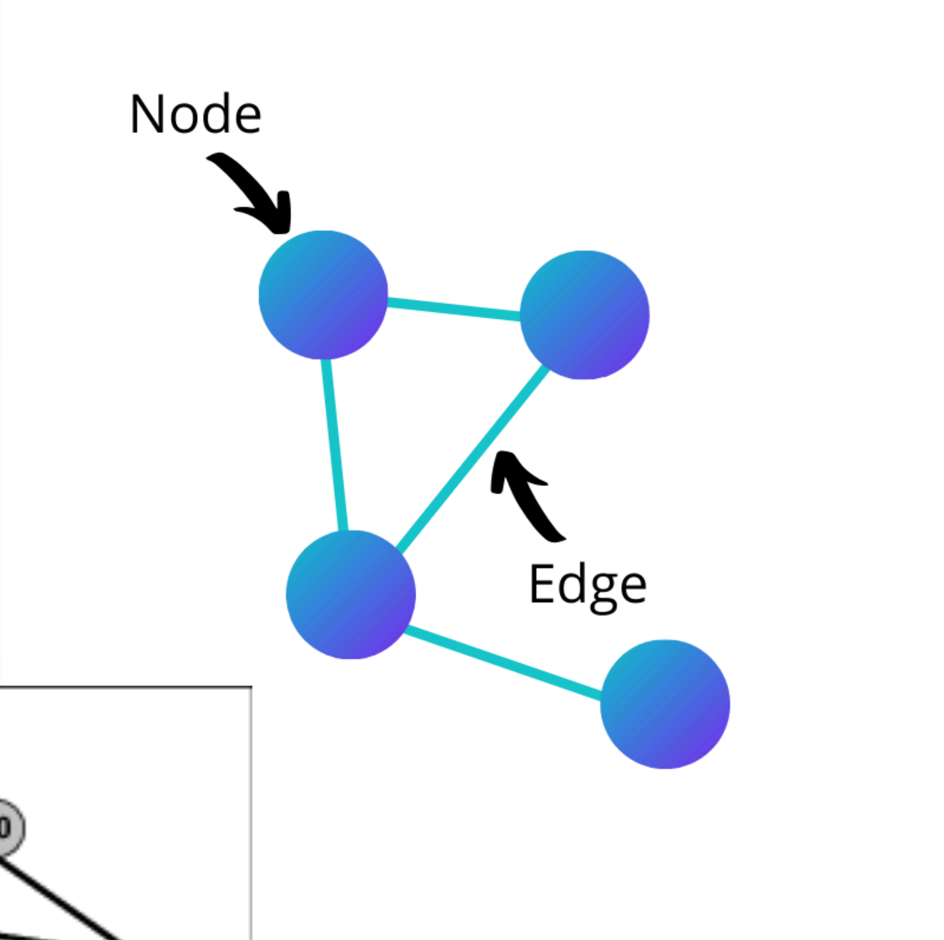


Source: https://arxiv.org/pdf/1901.03407v2.pdf

Graphs

Types of graphs



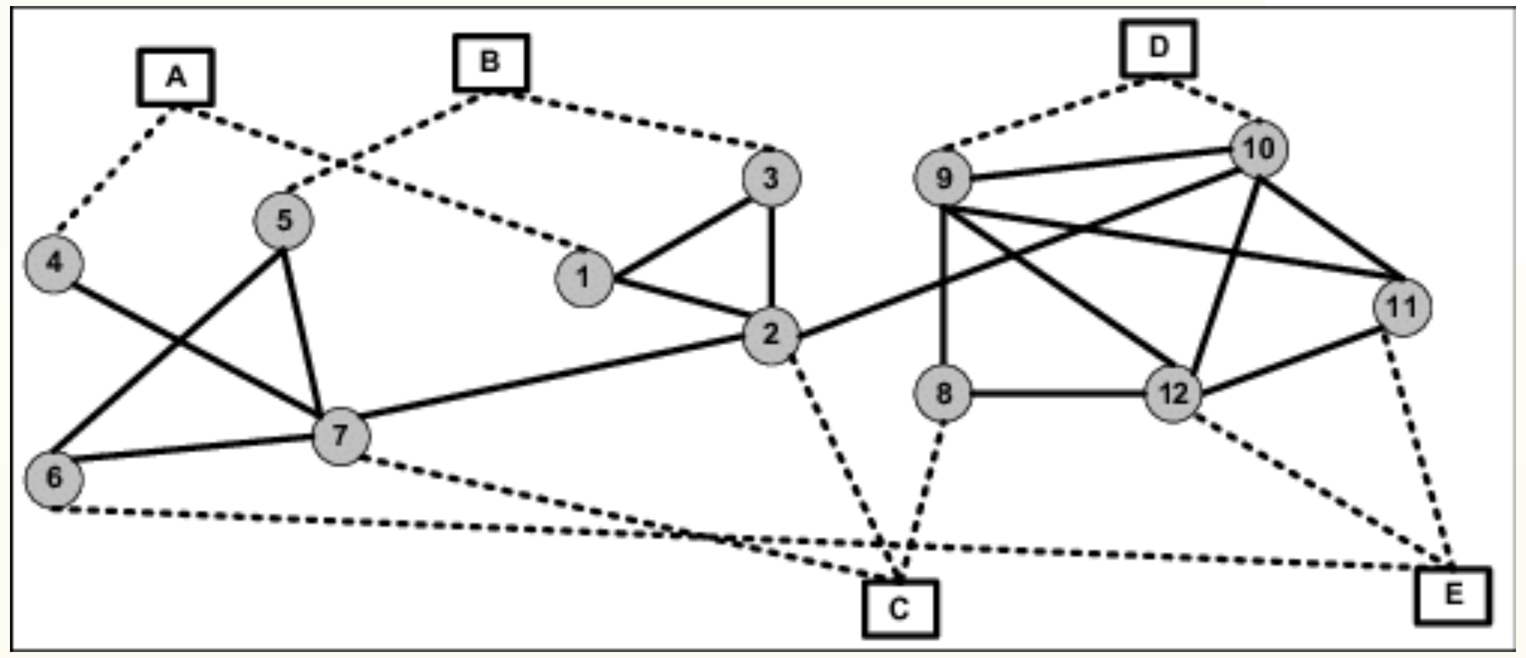


- Plain graph

  G = {V,E}

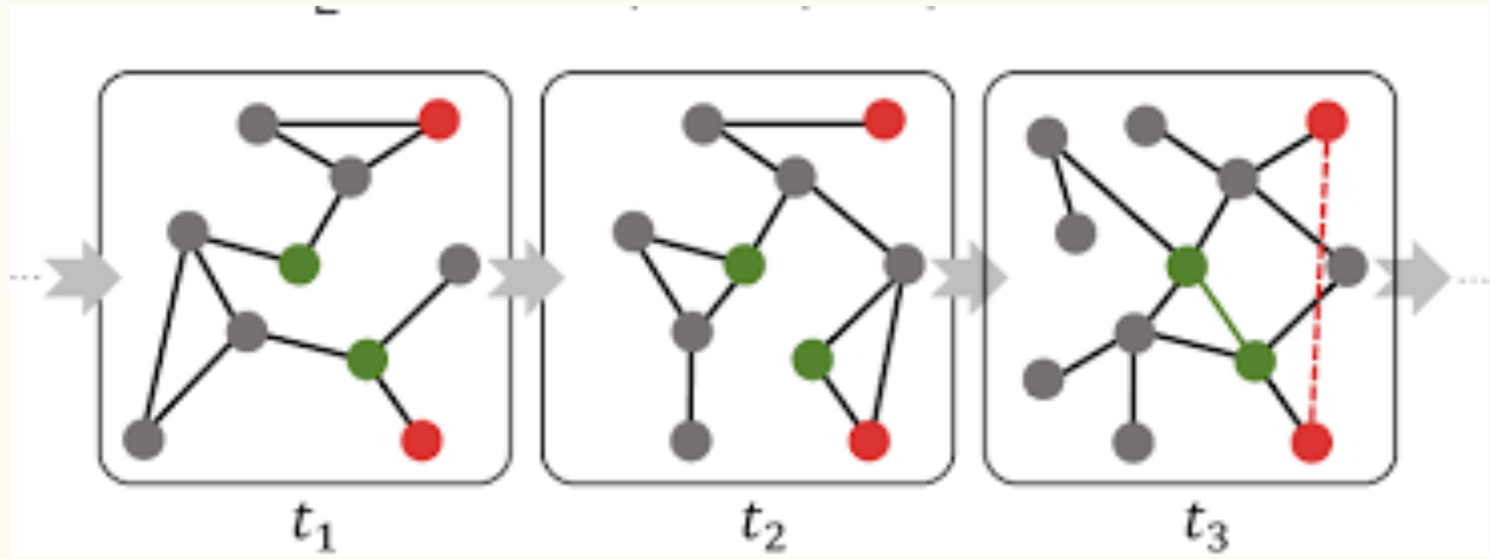
- Attributed graph

  G = {V,E,X}



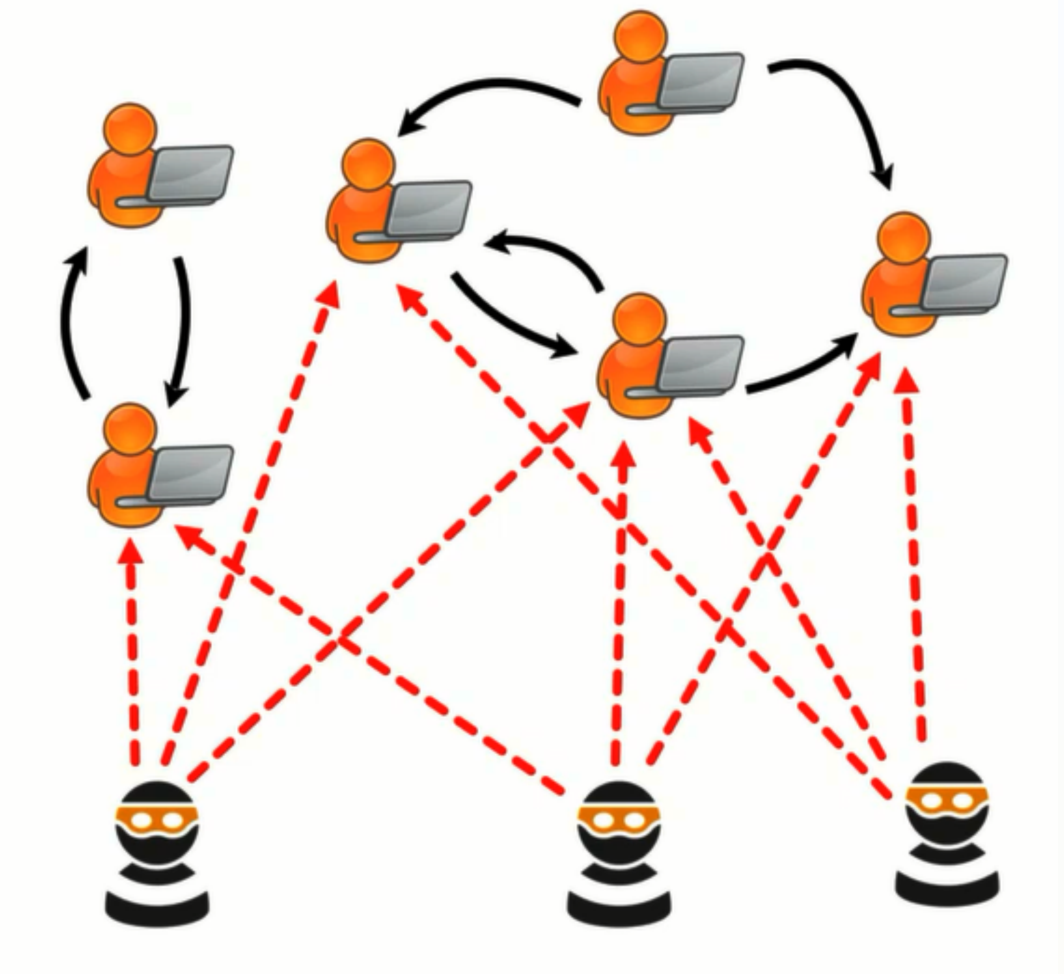
- Dynamic graph

  G = {V(t),E(t),Xv(t),Xe(t)}}



Source:https://miro.medium.com/max/1200/1*v_RunGMoLxaicBubSmKr1w.png
https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSv_0qmlmmAHLSwDj6zmze25sN609-7Yh9q9Q&usqp=CAU
https://www.researchgate.net/profile/Jun-Luo-25/publication/220895154/figure/fig1/AS:654058415222789@1532951086640/The-model-of-social-networks-w
information-In-this-example-there-are-12.png

Challenges in Graph Anomaly Detection



- Data specific challenges (Data-CHs)

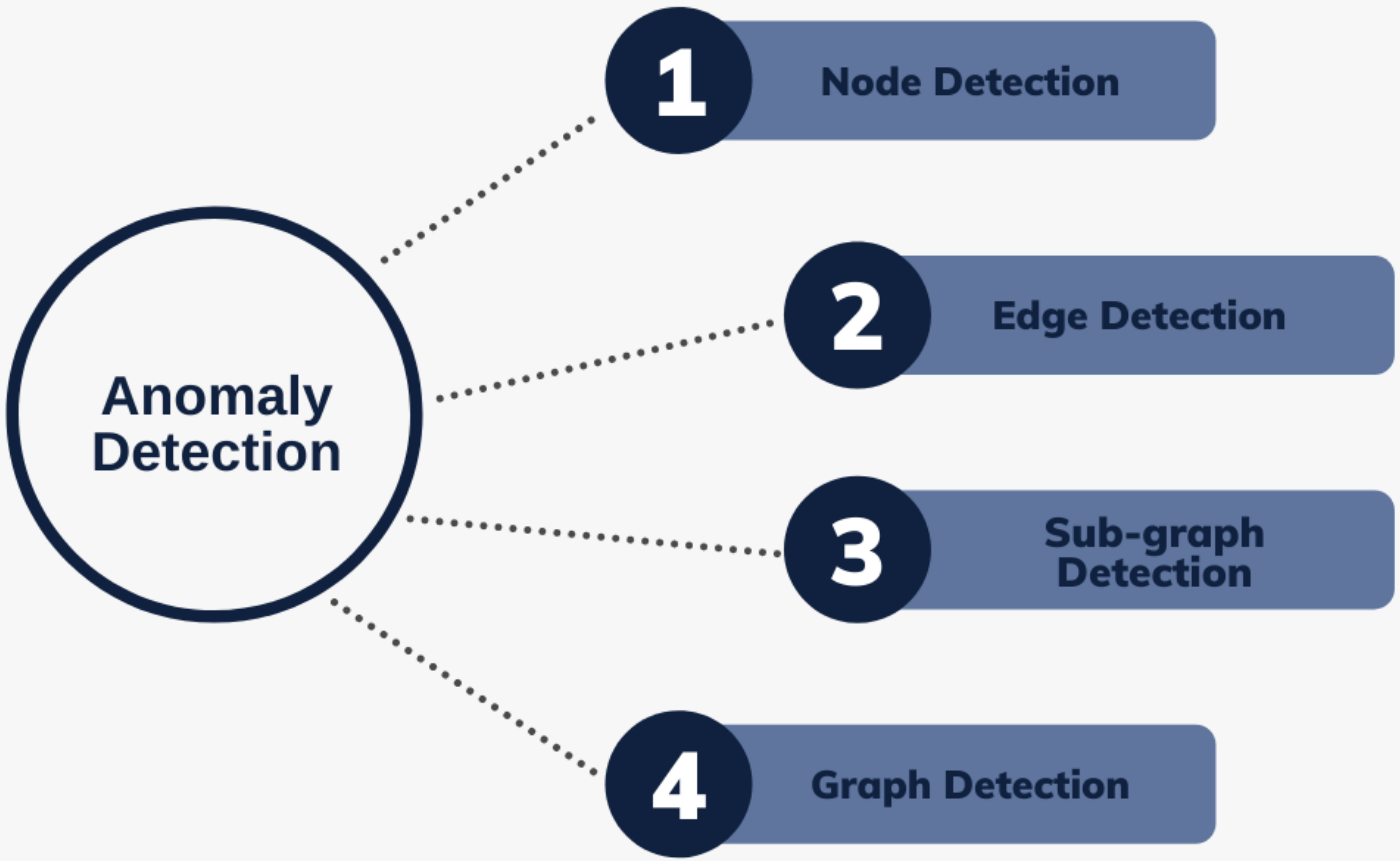
A. Scalability and dimensionality

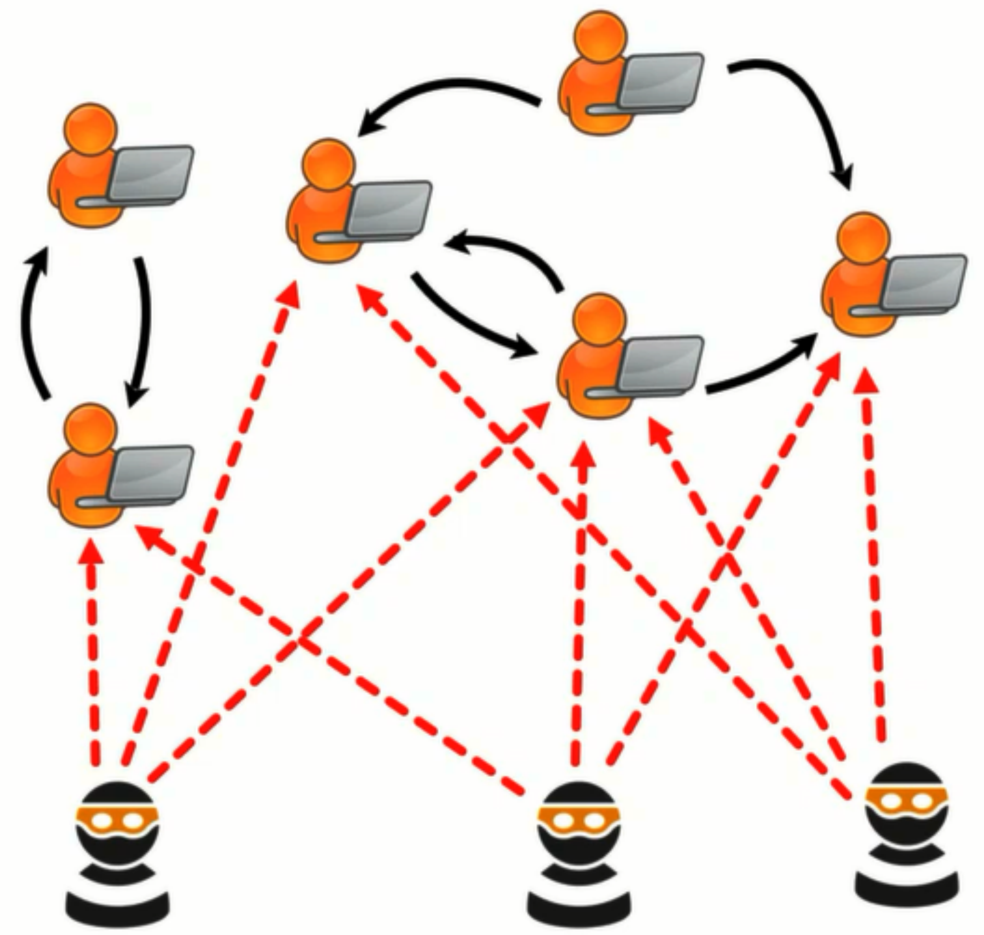
B. Class imbalance and interdependencies

- Technique specific challenges (Tech-CHs)

A. Noise and lack in labels

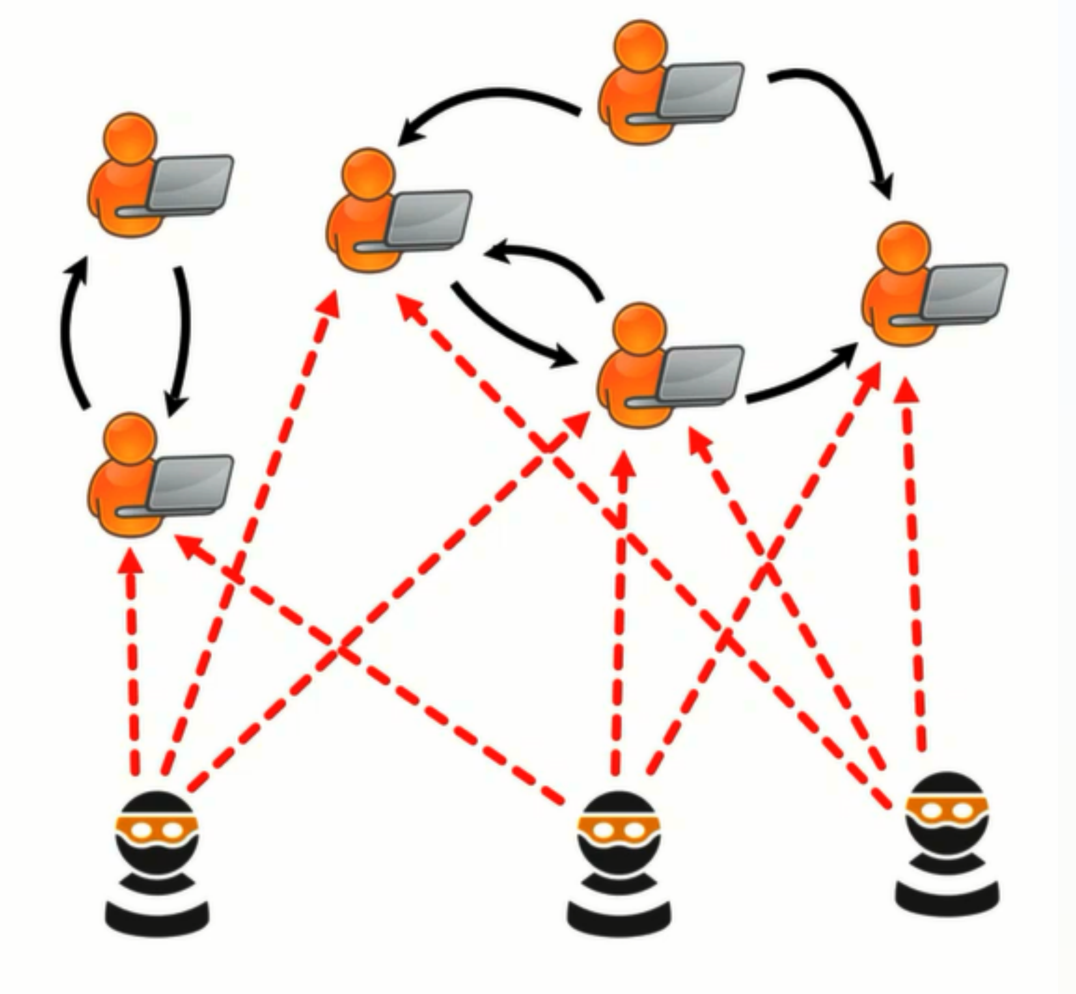
B. High training costs and interpretability


Anomaly Detection
1
Node Detection
2
Edge Detection
3
Sub-graph Detection
4
Graph Detection

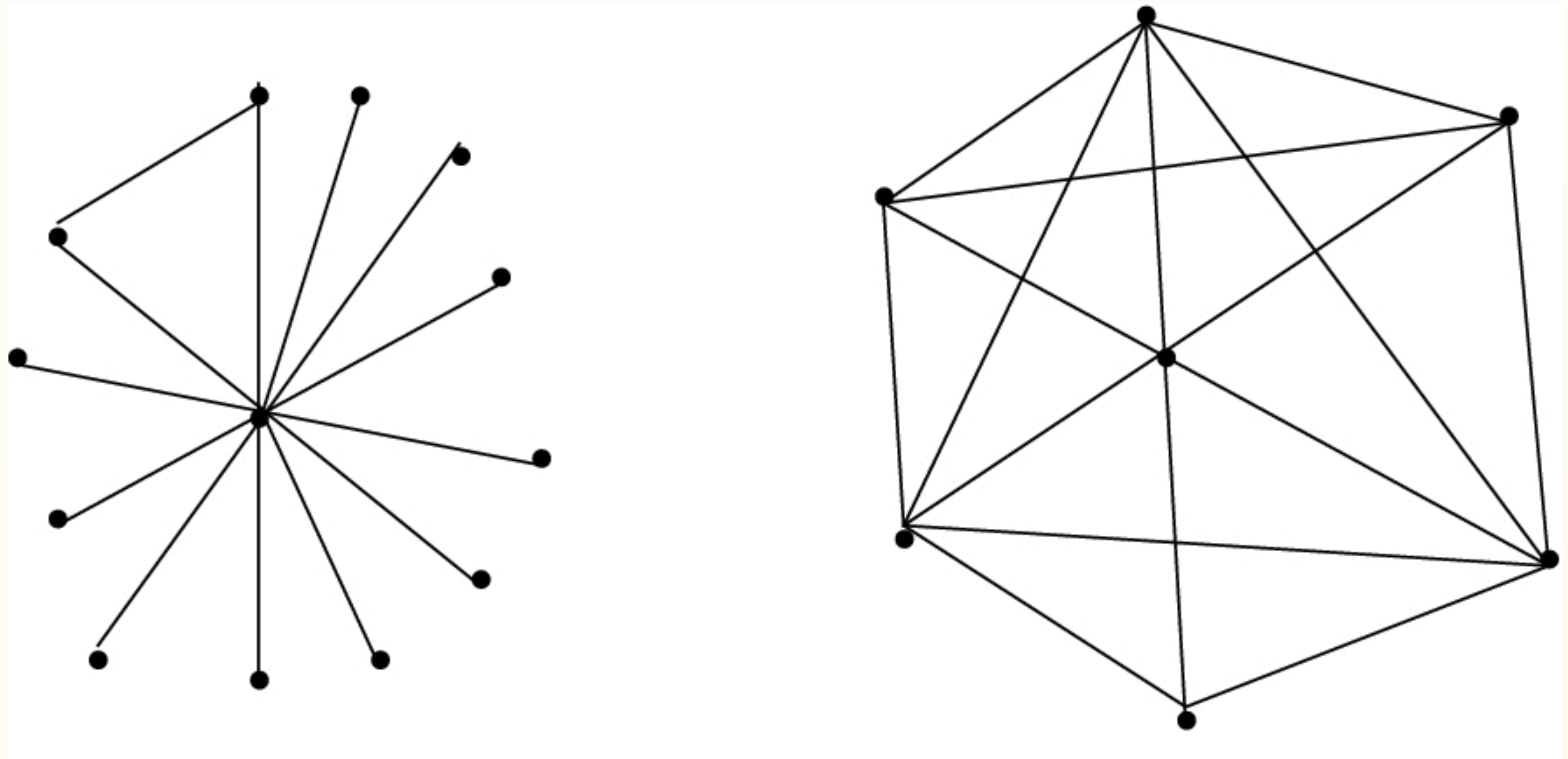


Anomalous Node Detection

ANOS ND on Plain Graphs



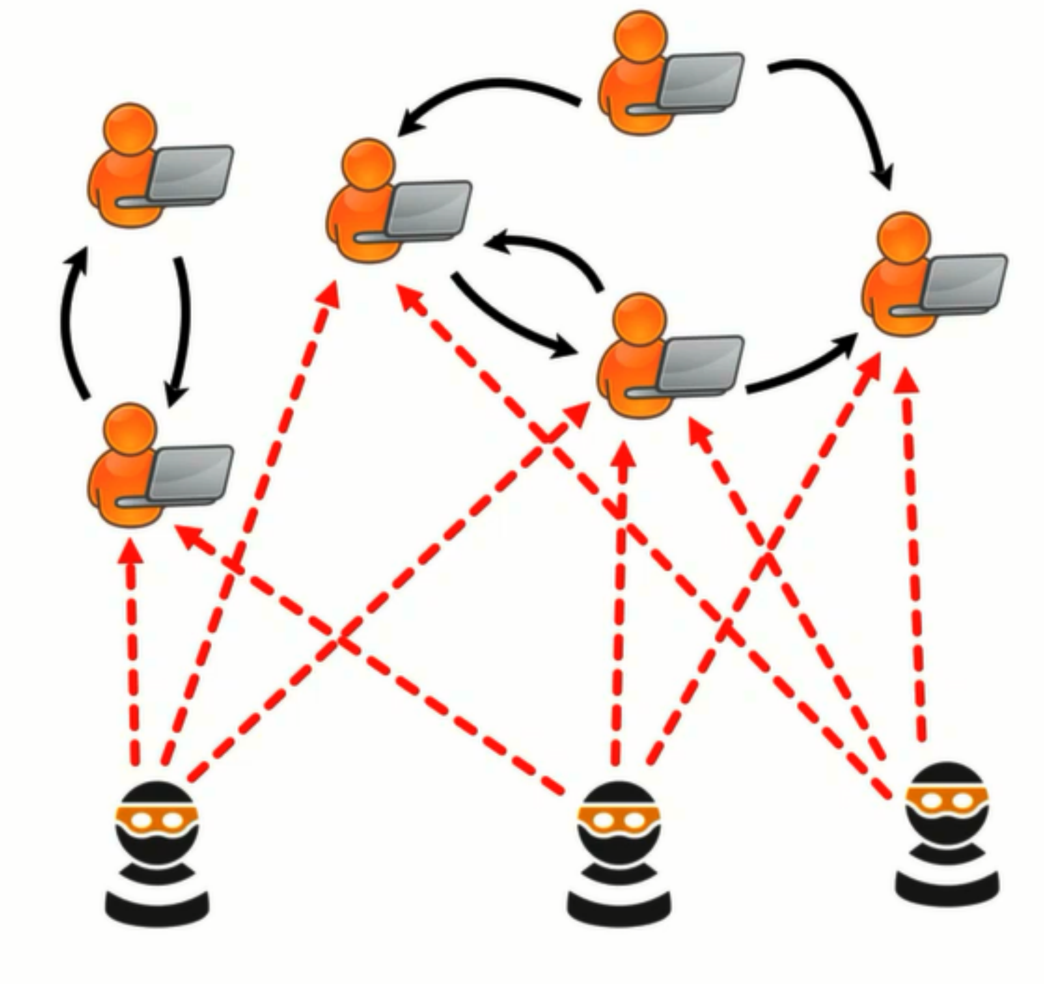
1. Traditional non-deep learning techniques (OddBall)



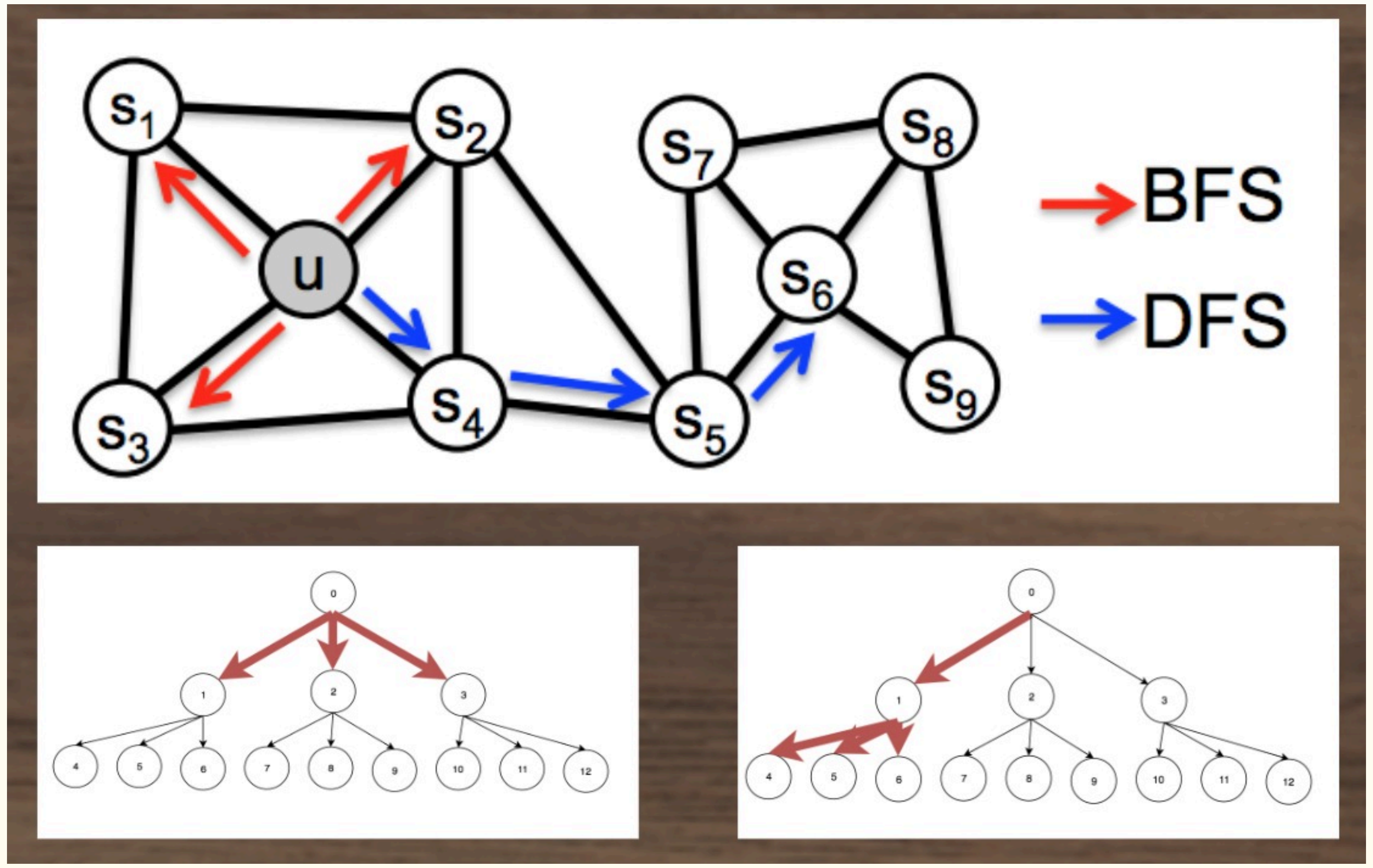
(a) Near star

(b) Near clique

Anomalous Node Detection

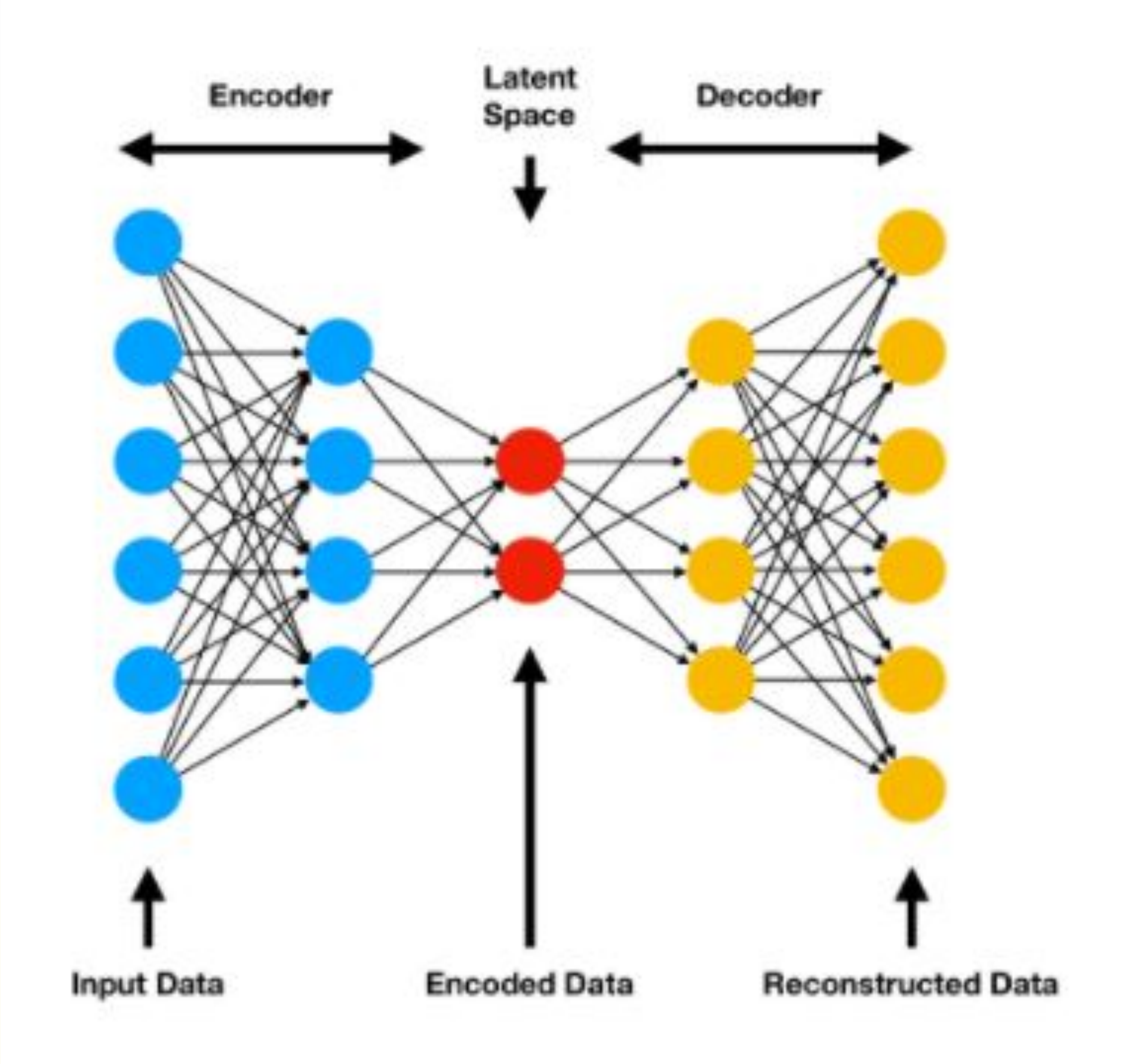


2. Network representation based techniques (Deepwalk, Node2Vec)



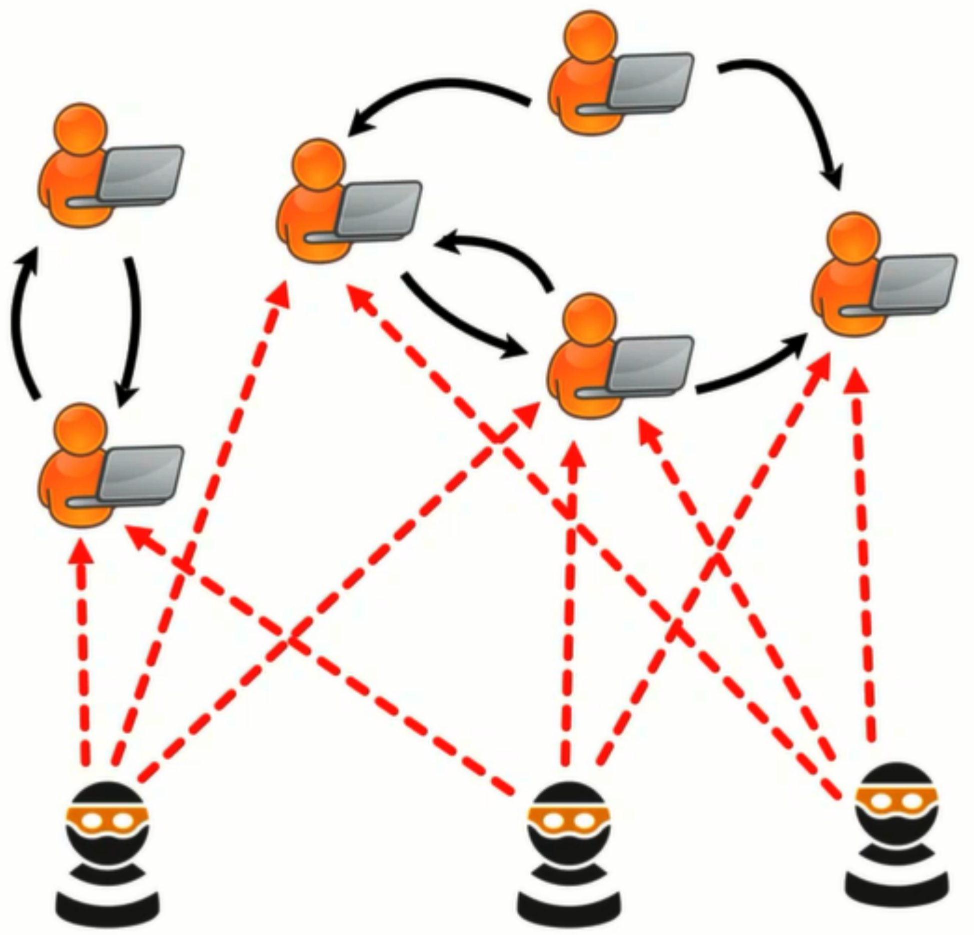
Autoencoder

- Encoder
- Latent space
- Decoder



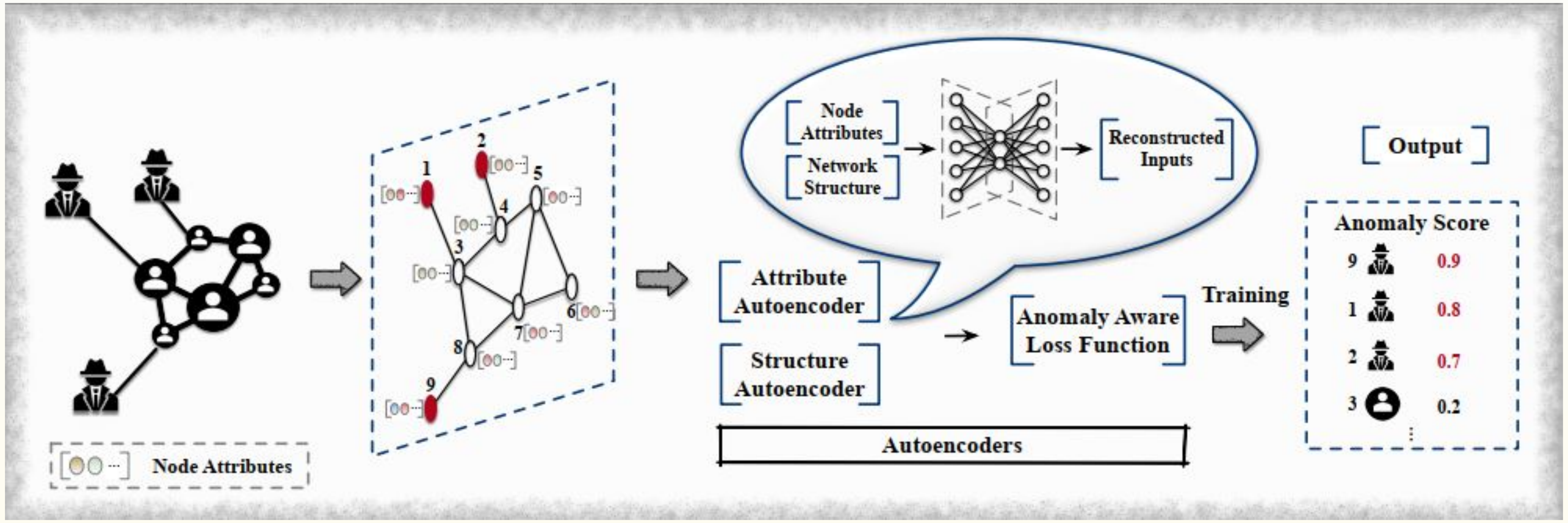
Source: https://ai-pool.com/a/s/understanding-autoencoders---an-unsupervised-learning-approach

ANOS ND on Attributed Graph



1. Deep NN Based Technique

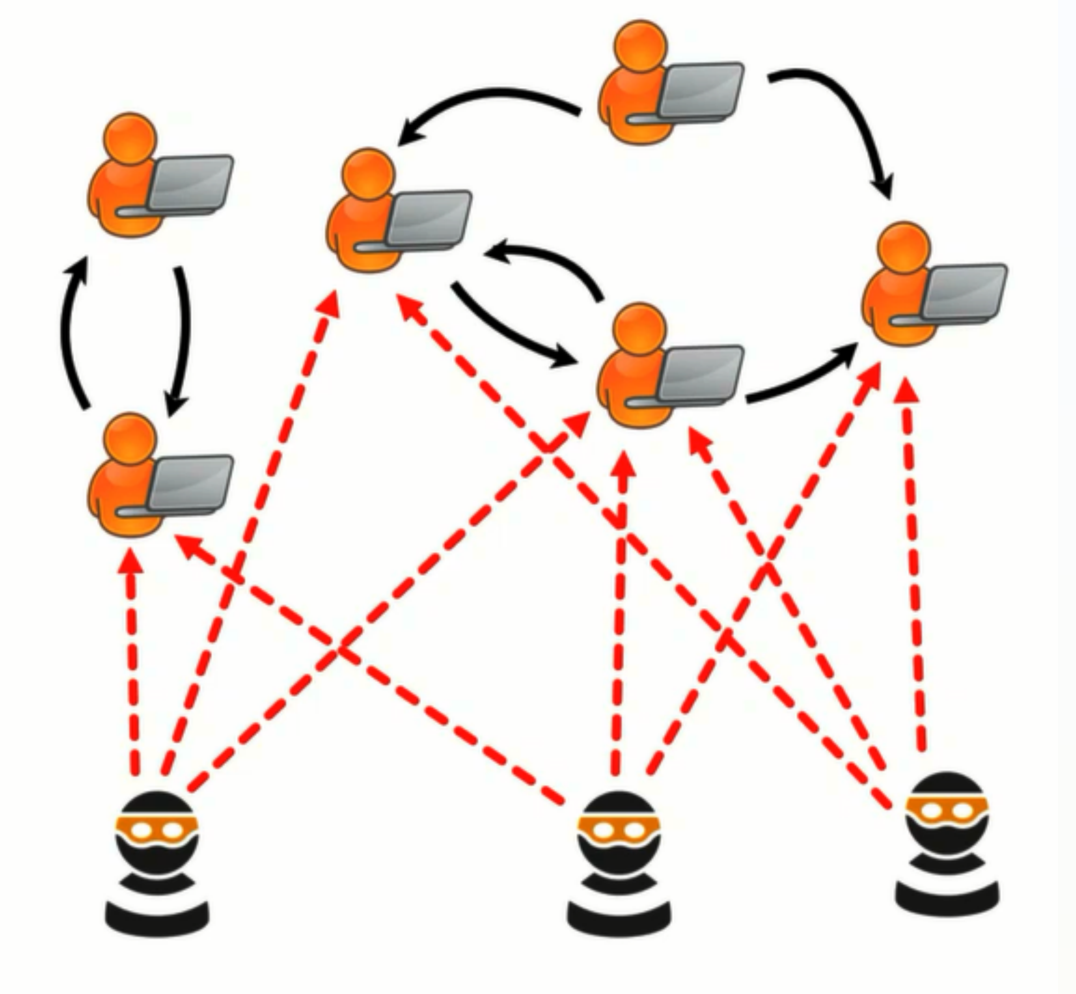
- Deep Outlier aware attributed Network Embedding (DONE)



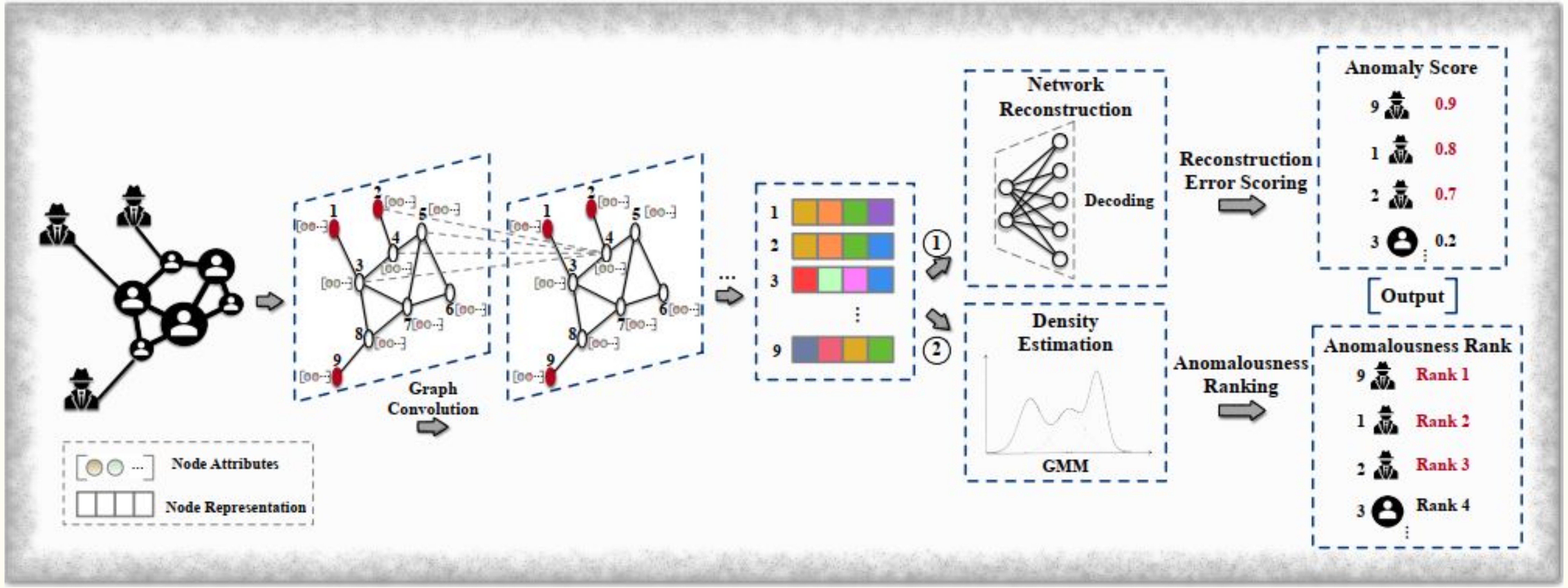
- Anomaly Score Assignment

1. Similar attributes
2. Connects
3. Different pattern

2. GCN Based Technique



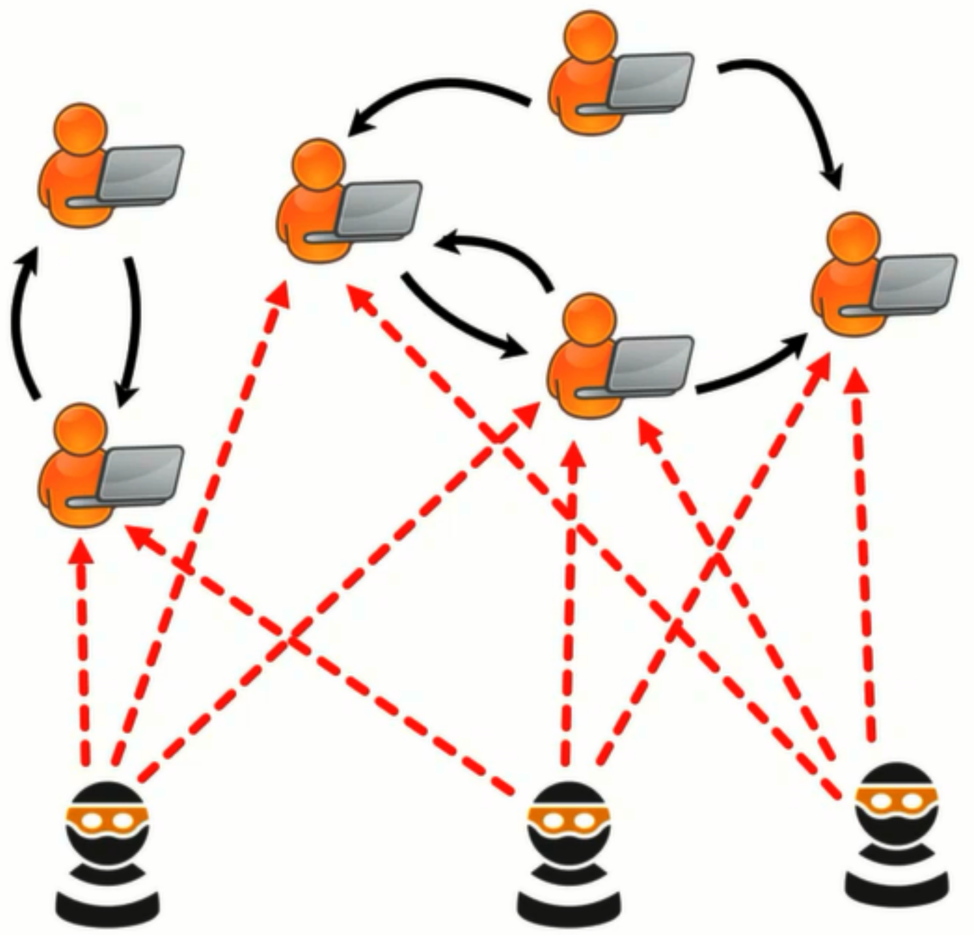
- Dominant



$$\mathcal{L}_{DOMINANT} = (1-\alpha)\mathcal{R}_S + \alpha\mathcal{R}_A$$

$$score(\mathbf{i}) = (1-\alpha)||\mathbf{a}_i - \hat{\mathbf{a}}_i||_2 + \alpha||\mathbf{x}_i - \hat{\mathbf{x}}_i||_2,$$

B. ANOS ND on Dynamic Graphs



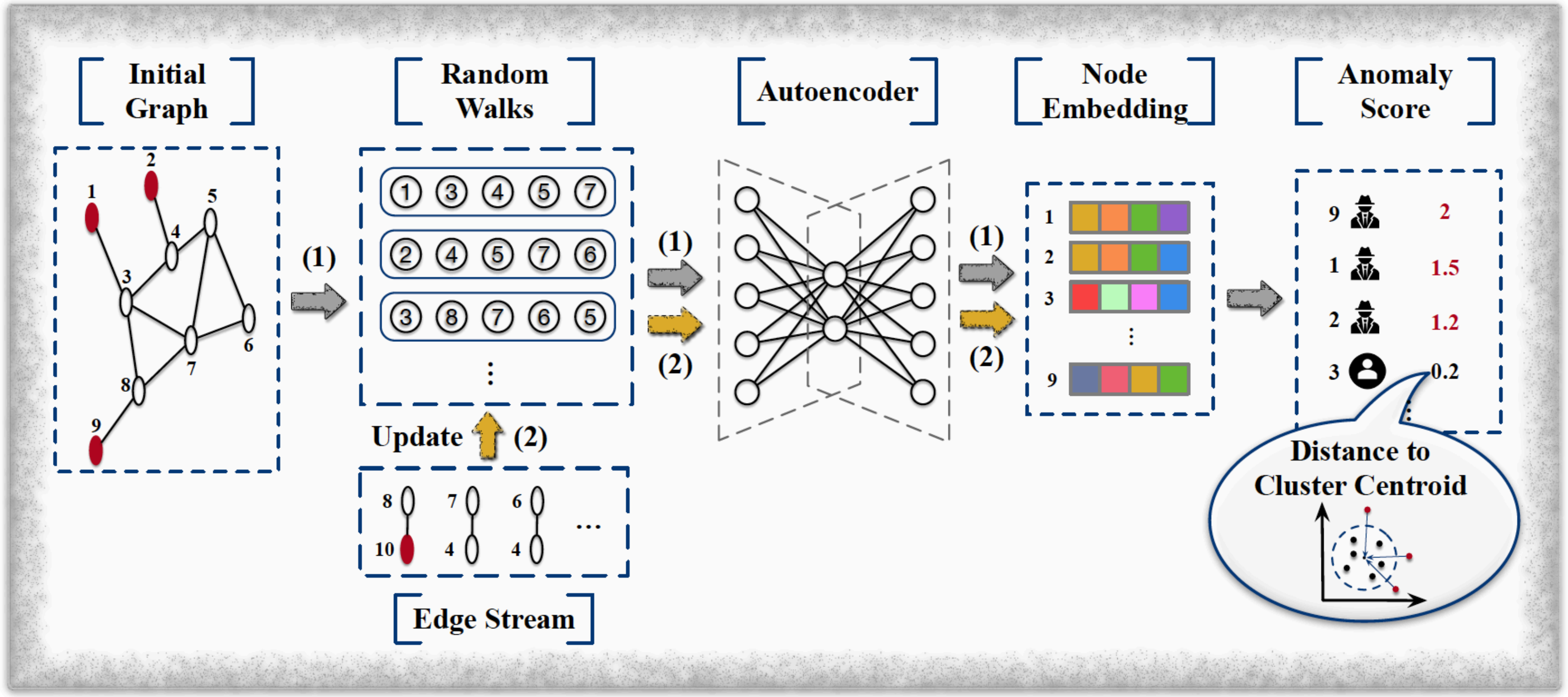
1. Network Representation Based Technique (NetWalk)

i) Random walk sequences

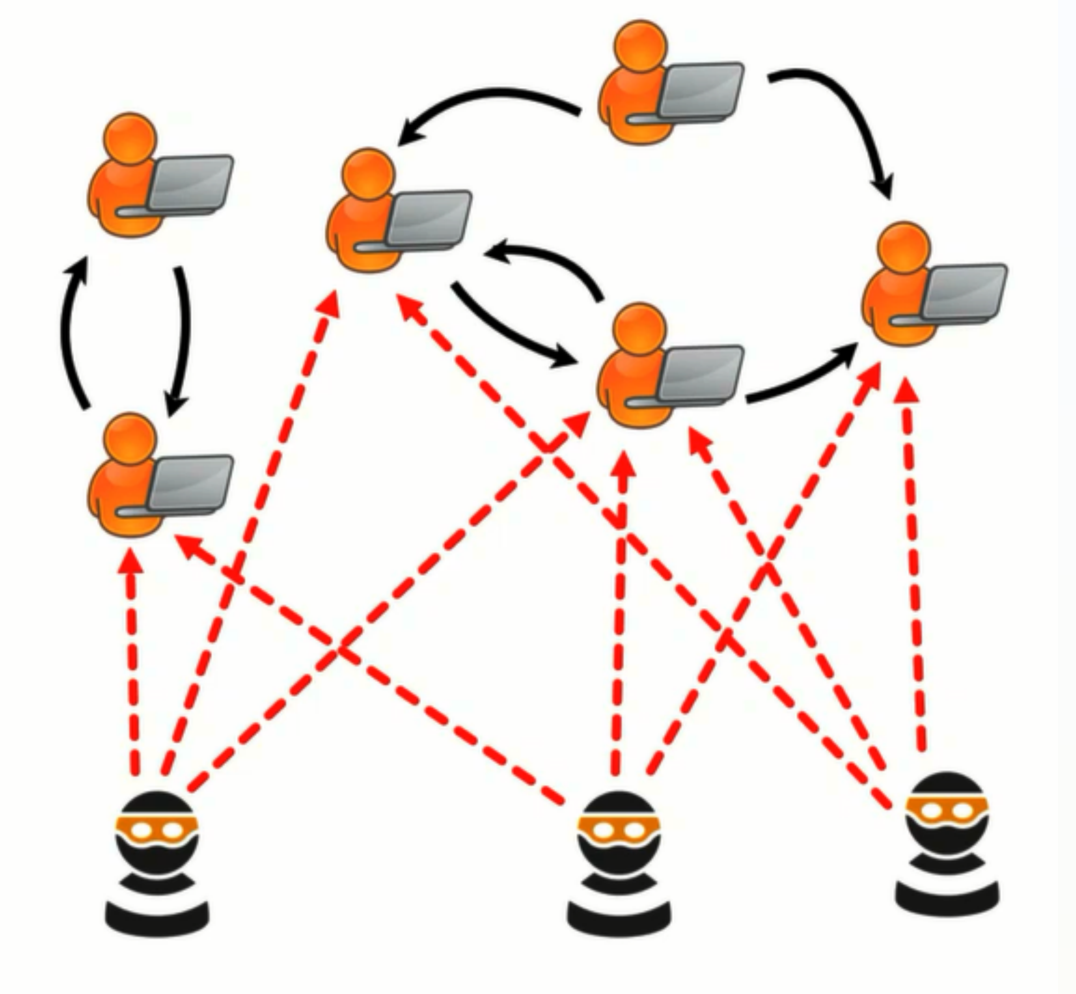
ii) Learning representations

iii) Maintaining the network representations

iv) Clustering based anomaly detection

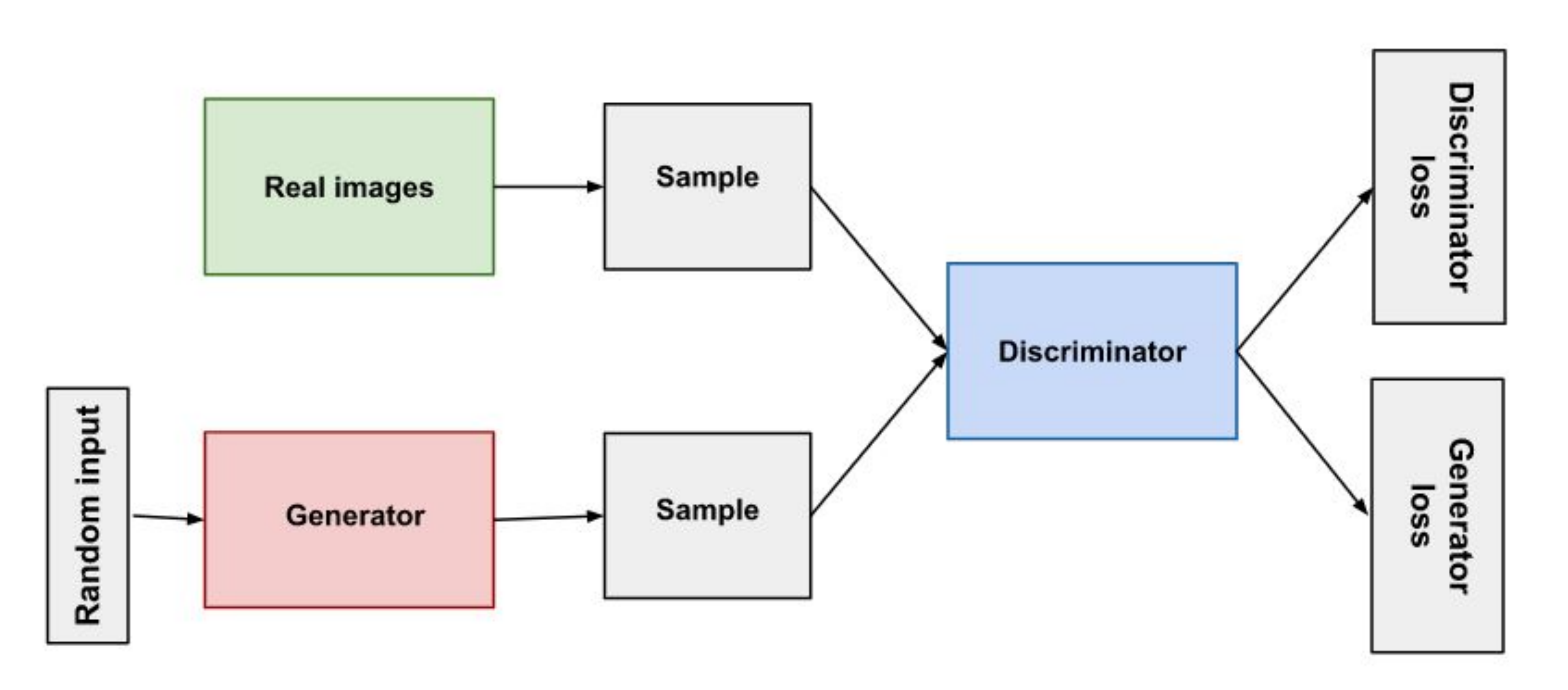


2. Generative Adversarial Networks (GAN)

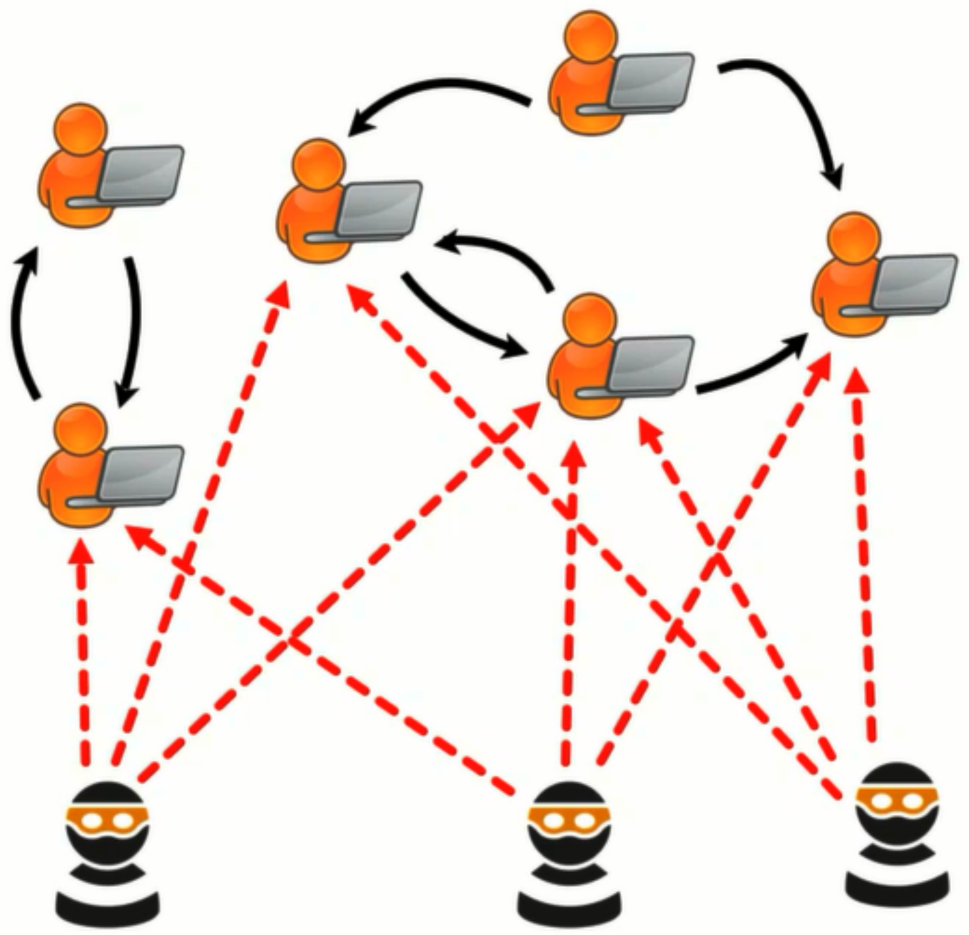


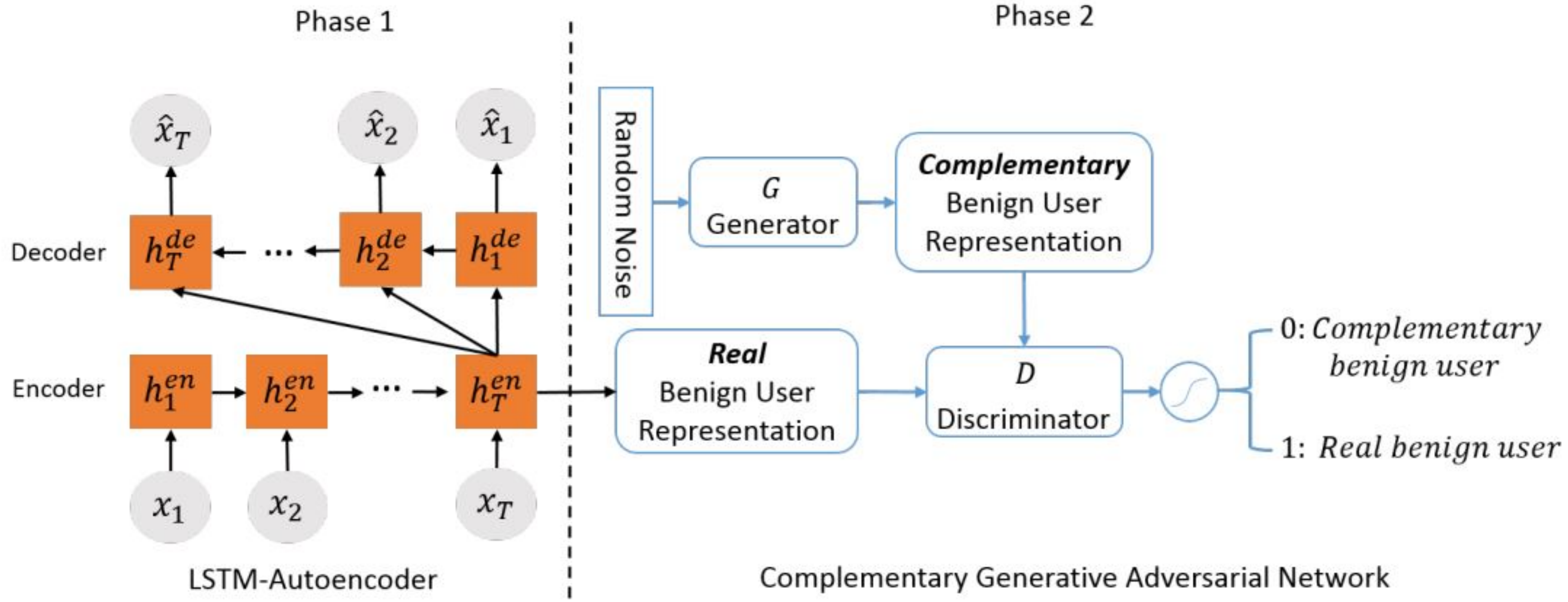
- Generator

- Discriminator



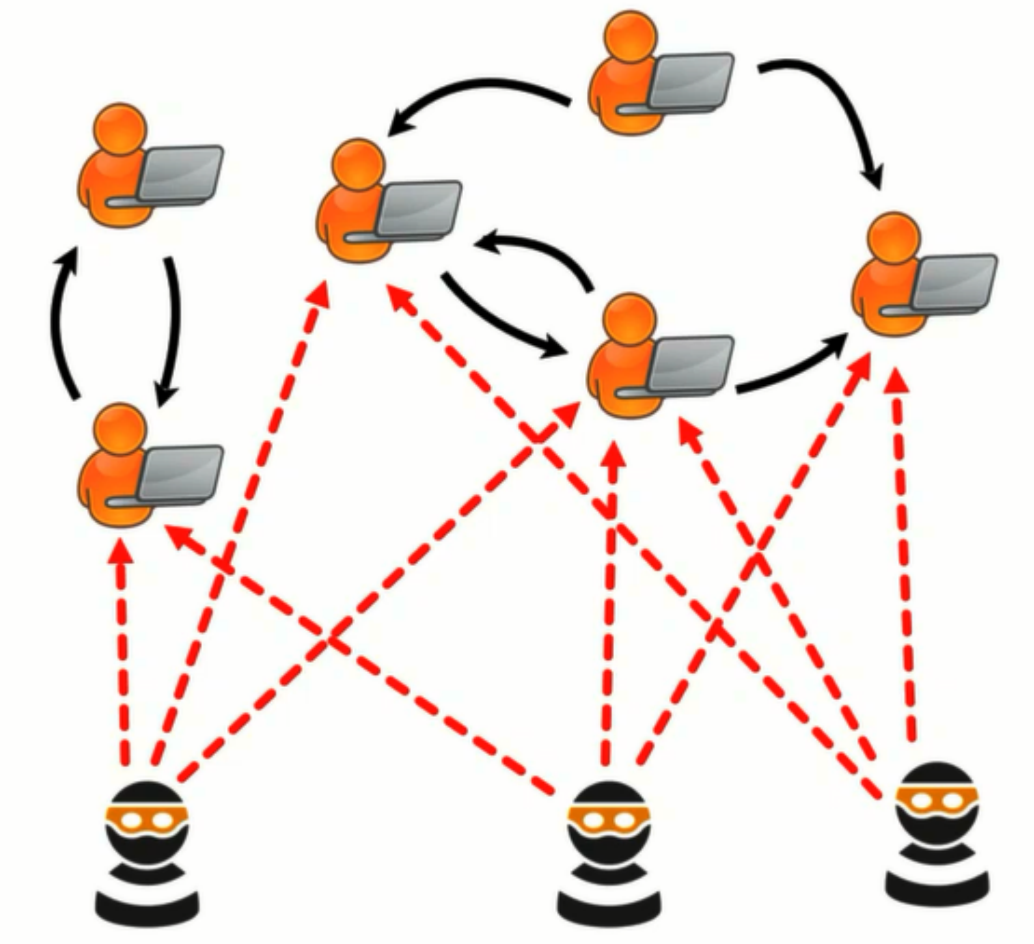
OCAN





Phase 1
Phase 2
$\hat{x}_T$
$\hat{x}_2$
$\hat{x}_1$
Decoder
h_T^{de}
h_2^{de}
h_1^{de}
Encoder
h_1^{en}
h_2^{en}
h_T^{en}
x_1
x_2
x_T
LSTM-Autoencoder
Random Noise
G Generator
Complementary Benign User Representation
Real Benign User Representation
D Discriminator
0: *Complementary benign user*
1: *Real benign user*
Complementary Generative Adversarial Network

Anomalous Edge Detection



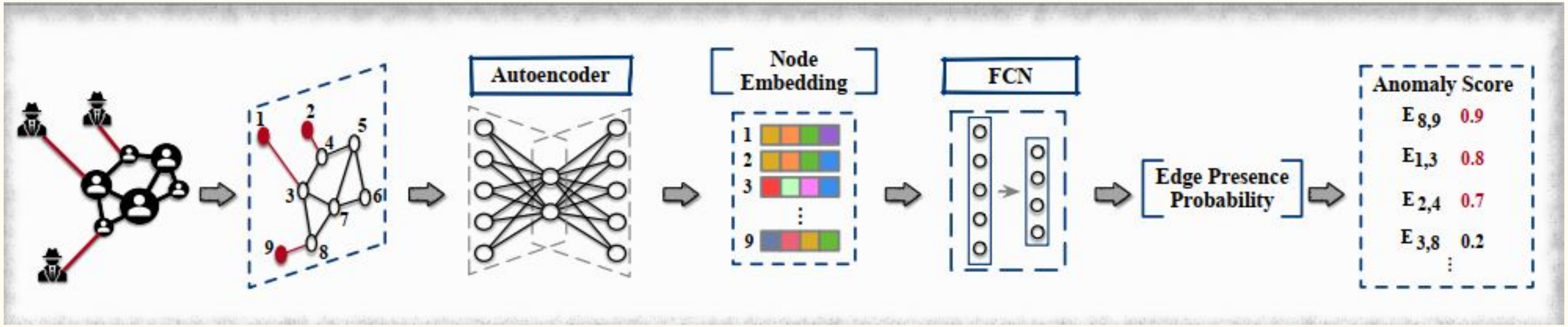
ANOS ED on Static Graphs

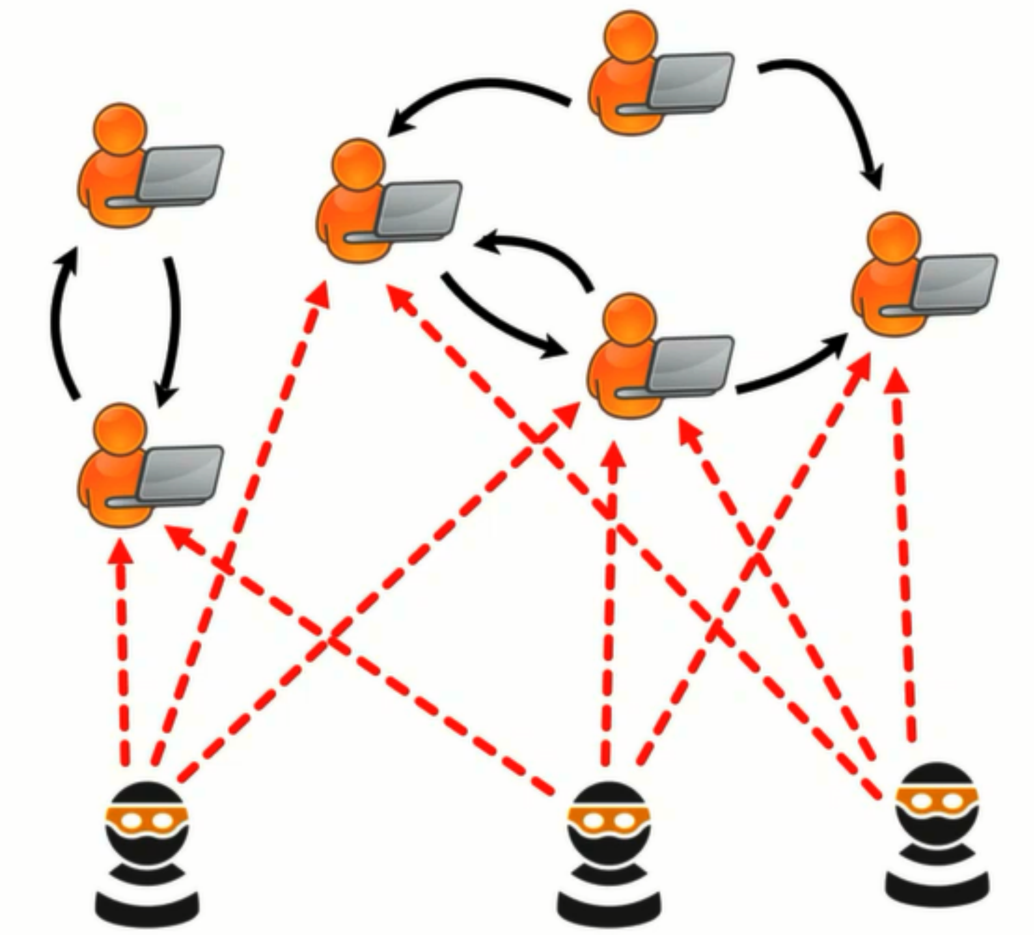
1. Deep NN Based Technique

- Unified Graph Embedding based anomalous edge Detection (UGED)

Probability of each edge: $P(v|u, N(u))$ and $P(u|v, N(v))$

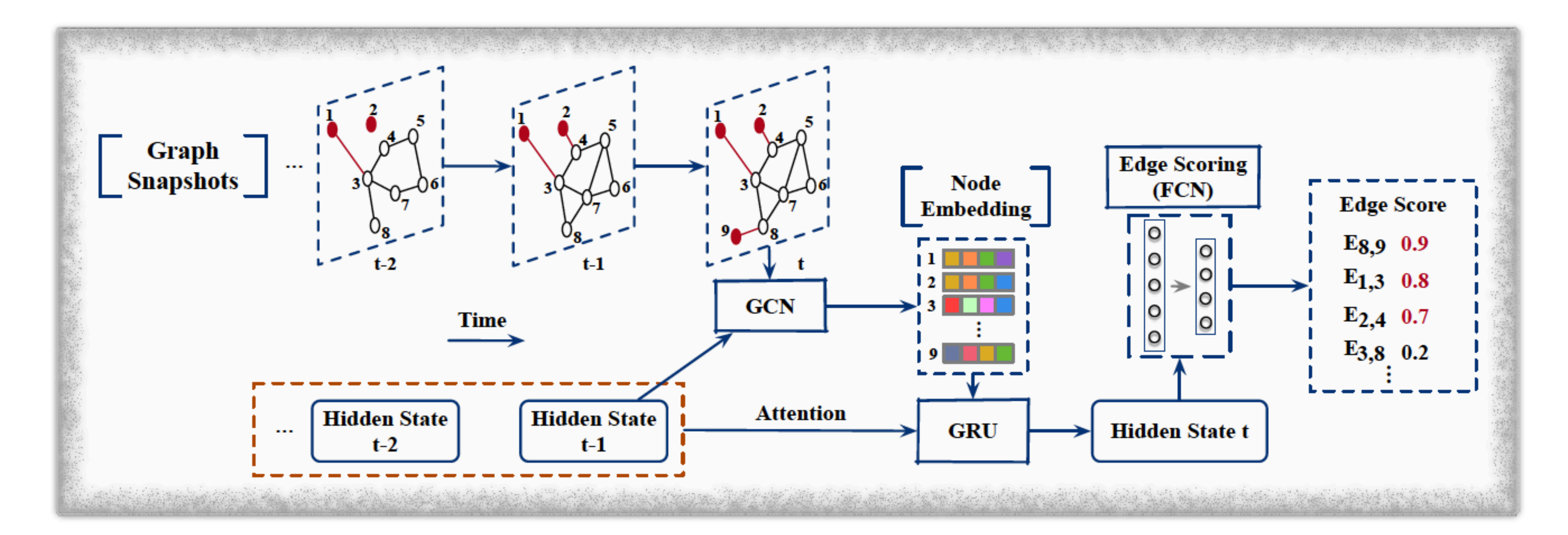
Mean aggregation: $H = h(C(u), \{C(n)|n \in N(u)\})$



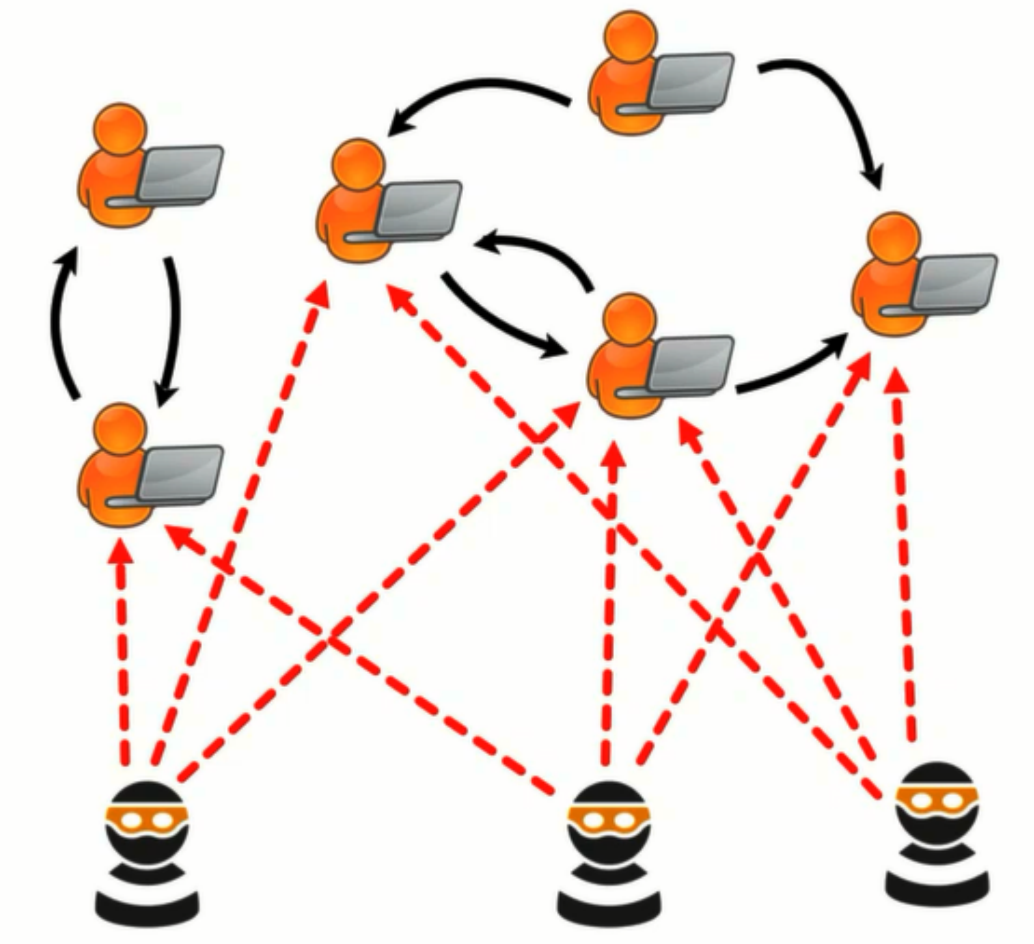


ANOS ED on Dynamic Graphs

- GCN Based Technique (AddGraph)



Anomalous Sub-Graph Detection



Deep Learning Technique

- Deep Fraud Detection (DeepFD)

i) Measures similarity: sim_{ij}

ii) User representations are trained using 3 losses

A. Reconstruction loss: $\mathcal{L}_{res}$

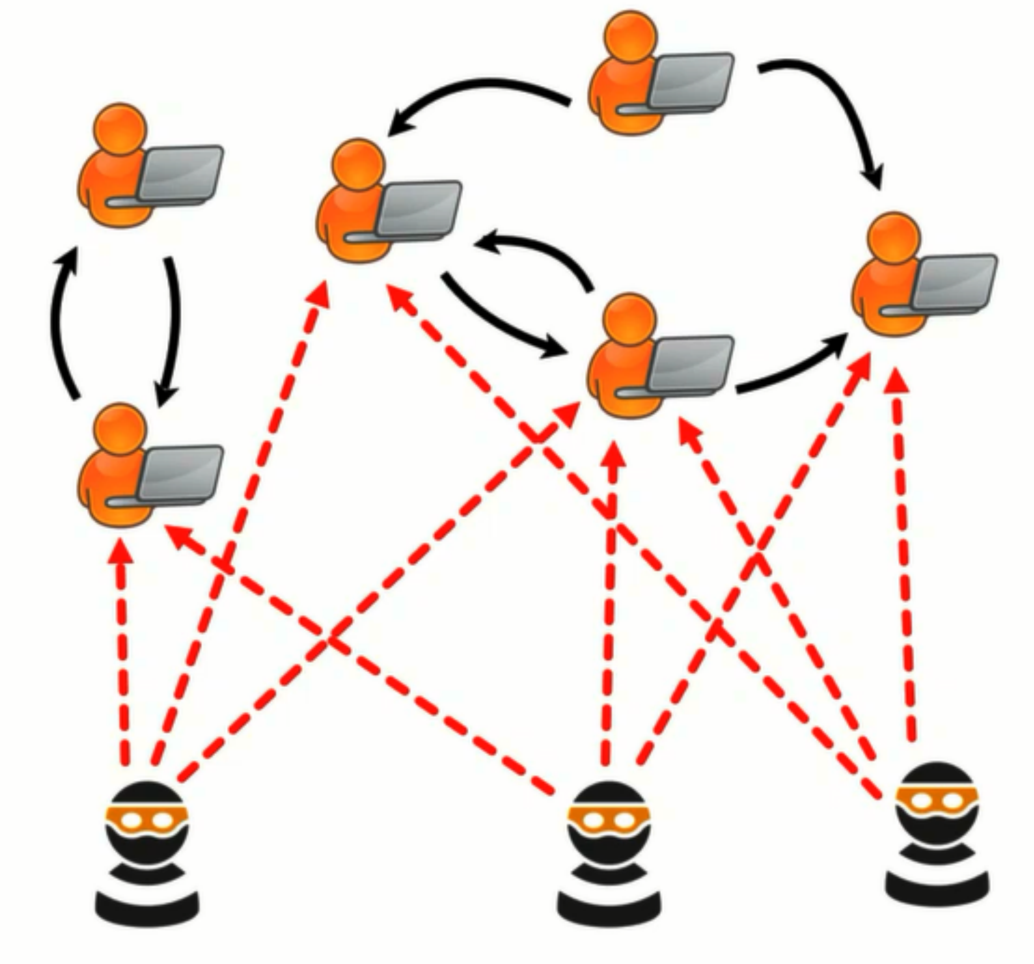
B. Similarity loss: $\mathcal{L}_{sim}$

C. Regularisation loss: $\mathcal{L}_{reg}$

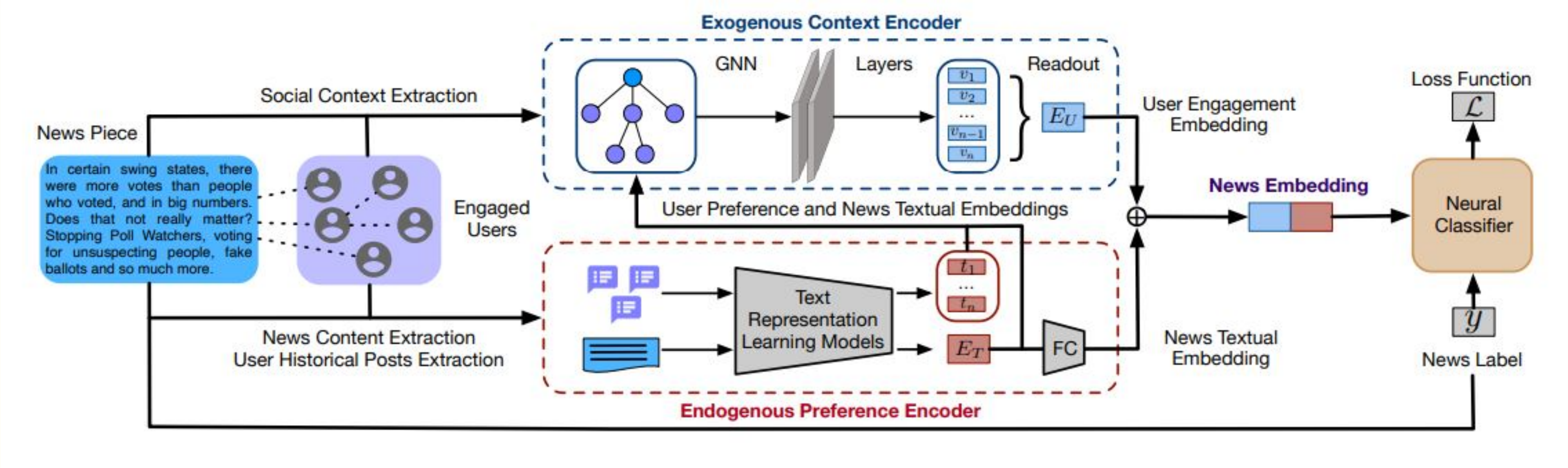
- Other methods under deep learning include FraudNE

Anomalous Graph Detection

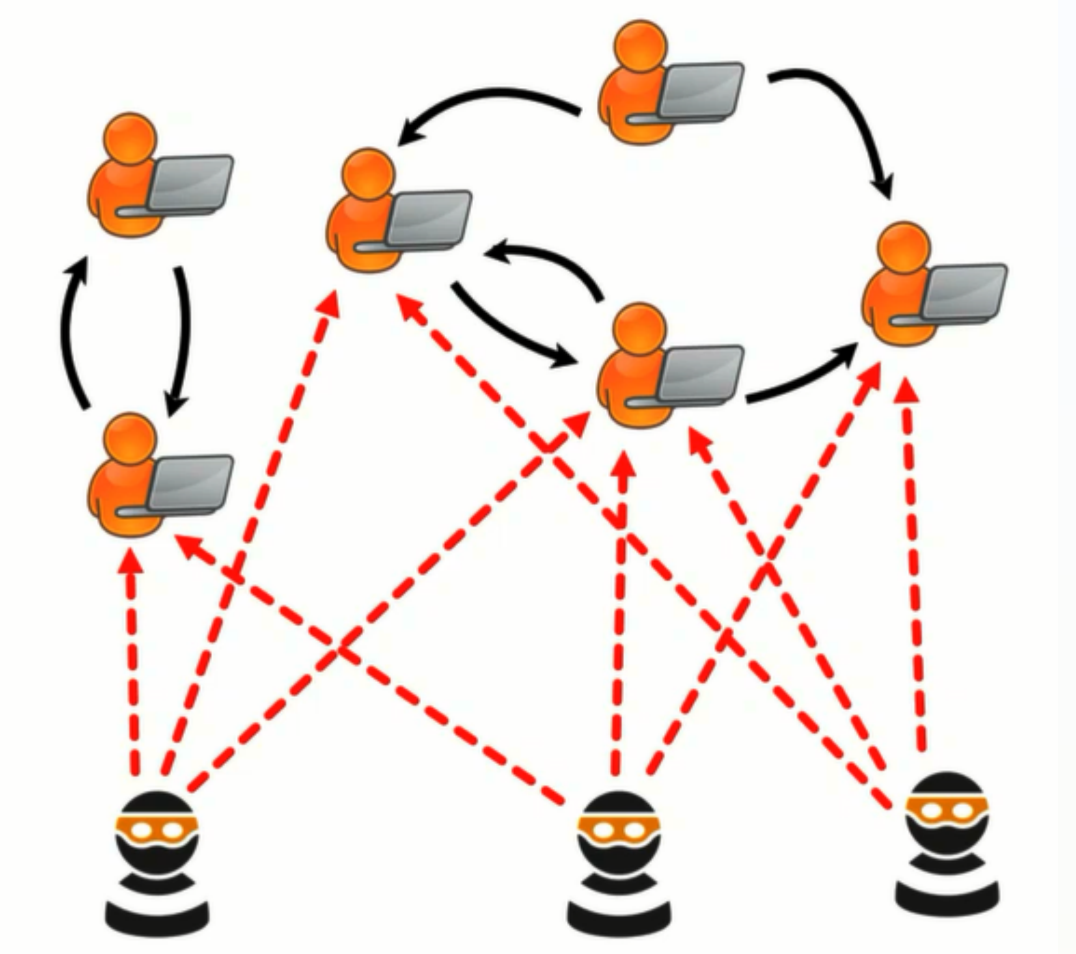
GNN Based Technique



- User Preference aware Fake news Detection (UPFD)

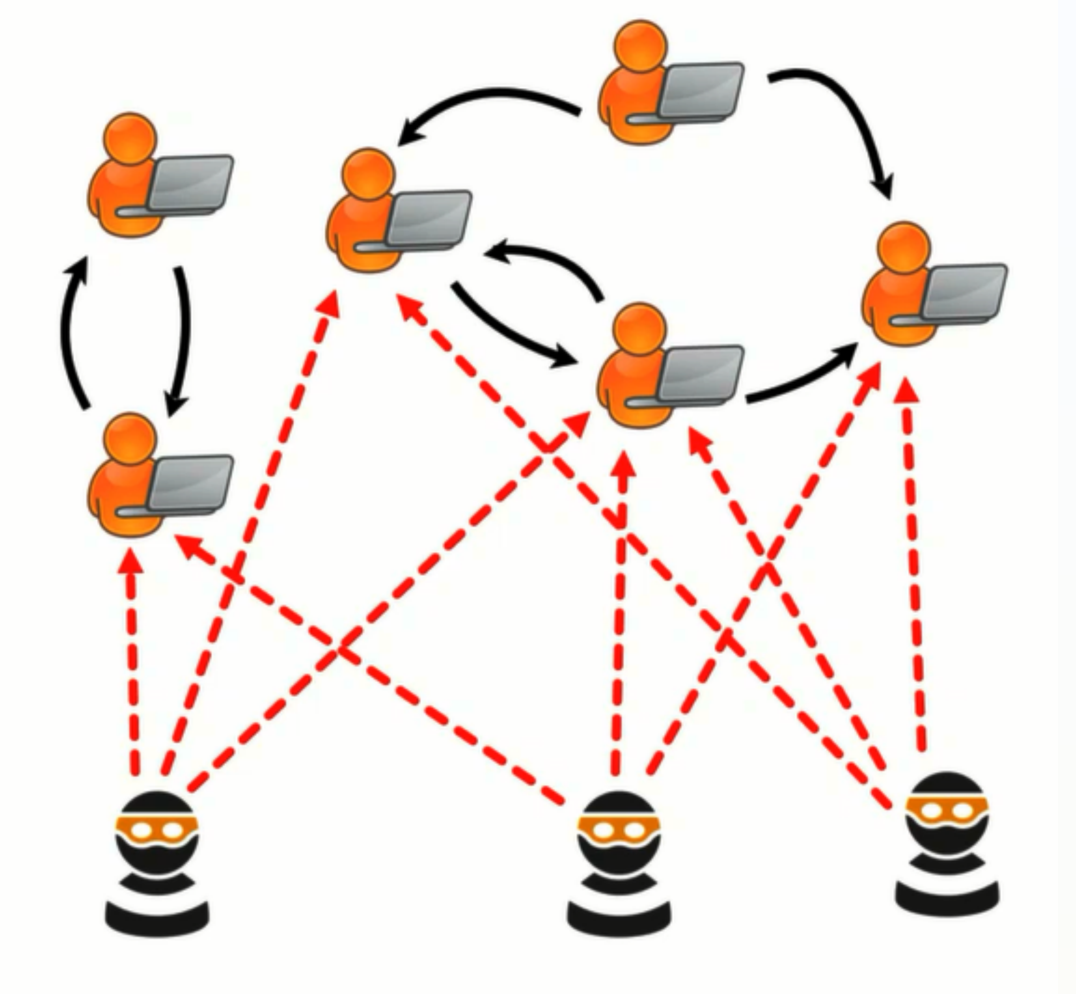


AD Performance Evaluation



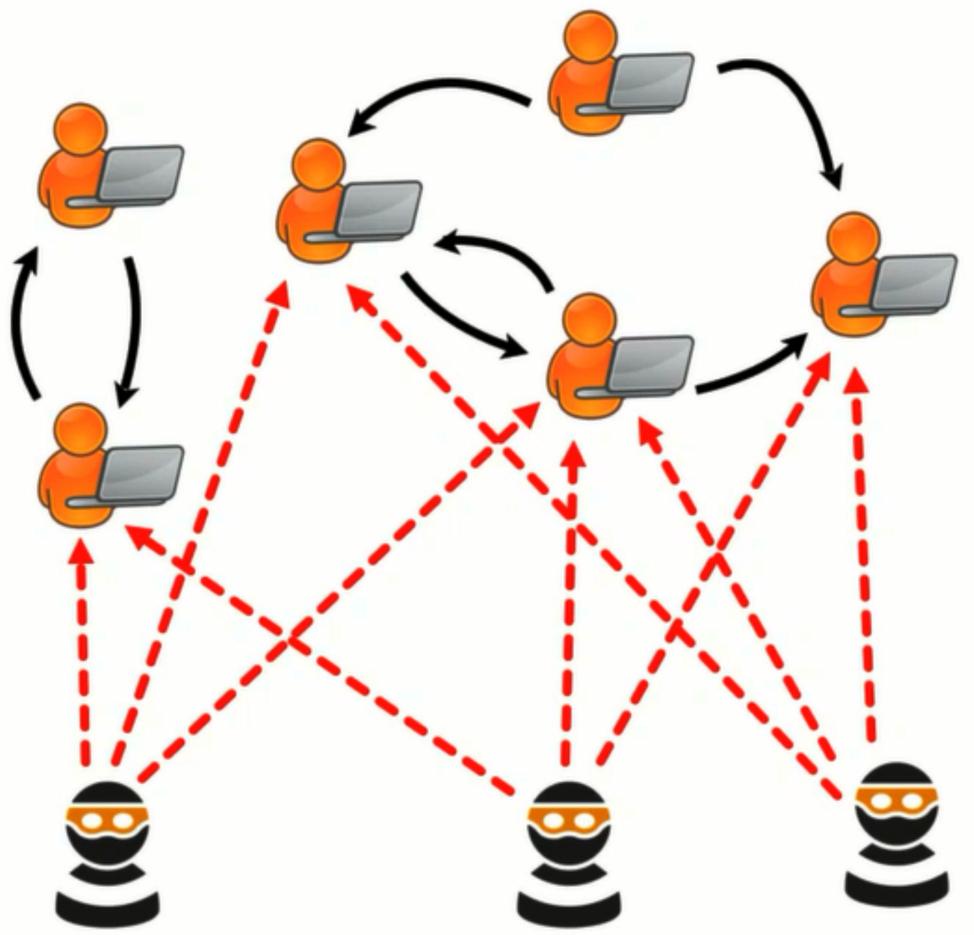
Evaluation Metric	Formula/Description
Accuracy	$\frac{tp+tn}{tp+tn+fp+fn}$
Precision	$\frac{tp}{tp+fp}$
Recall	$\frac{tp}{tp+fn}$
F1 Score	$2 * \frac{Recall*Precision}{Recall+Precision}$

Future Scope



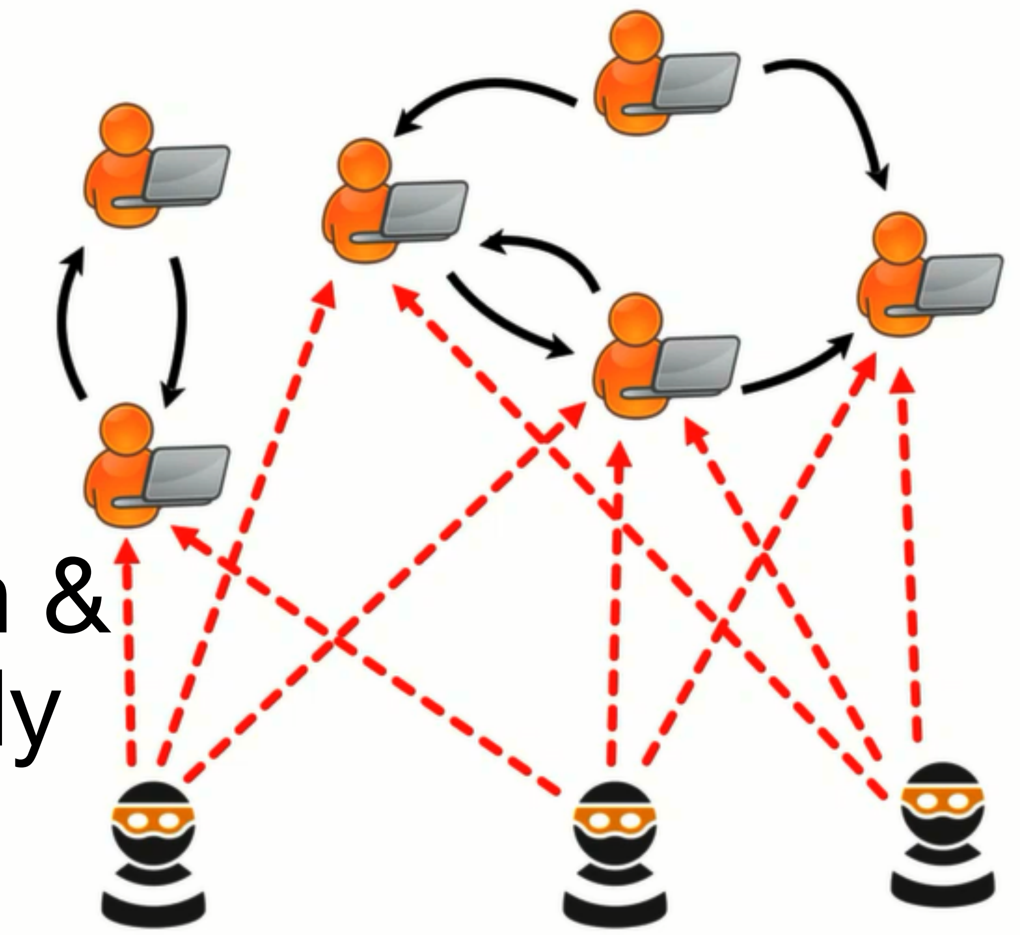
- Prior knowledge
- Two pronged approach in dynamic graphs
- Streaming heterogenous graphs
- Imbalanced graph anomaly detection

Source: https://www.youtube.com/watch?v=lF11k2s1uWw



Conclusion

References



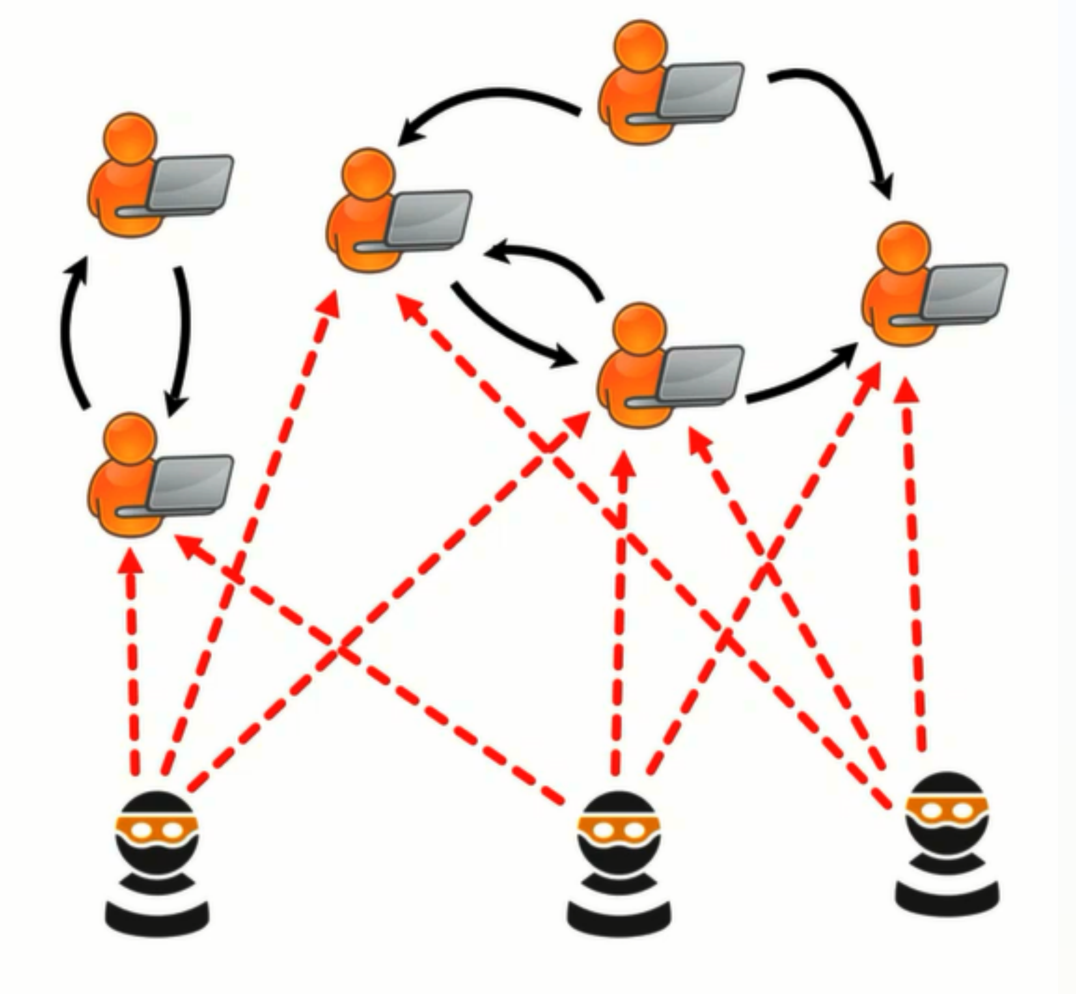
- Ma, Xiaoxiao & Wu, Jia & Xue, Shan & Yang, Jian & Zhou, Chuan & Sheng, Quan & Xiong, Hui & Akoglu, Leman. (2021). A Comprehensive Survey on Graph Anomaly Detection with Deep Learning. IEEE Transactions on Knowledge and Data Engineering. 1-1. 10.1109/TKDE.2021.3118815

- S. Bandyopadhyay, L. N, S. V. Vivek, and M. N. Murty, “Outlier resistant unsupervised deep architectures for attributed network embedding,” in Proc. ACM 13th Int. Conf. Web Search Data Mining, 2020, pp. 25–33

- L. Ouyang, Y. Zhang and Y. Wang, "Unified Graph Embedding-Based Anomalous Edge Detection," 2020 International Joint Conference on Neural Networks (IJCNN), 2020, pp. 1-8, doi: 10.1109/IJCNN48605.2020.9206720

- Y. Dou, K. Shu, C. Xia, P. S. Yu, and L. Sun, “User preference-aware fake news detection,” arXiv preprint arXiv:2104.12259, 2021.

- Yoon, Minji & Hooi, Bryan & Shin, Kijung & Faloutsos, Christos. (2019). Fast and Accurate Anomaly Detection in Dynamic Graphs with a Two-Pronged Approach. 647-657. 10.1145/3292500.3330946



Thank you