Group-based Fraud Detection Network on e-Commerce Platforms

Jianke Yu[1,2], Hanchen Wang[1,3], Xiaoyang Wang[4], Zhao Li[5,6], Lu Qin[3], Wenjie Zhang[4], Jian Liao[2], Ying Zhang[1,3]

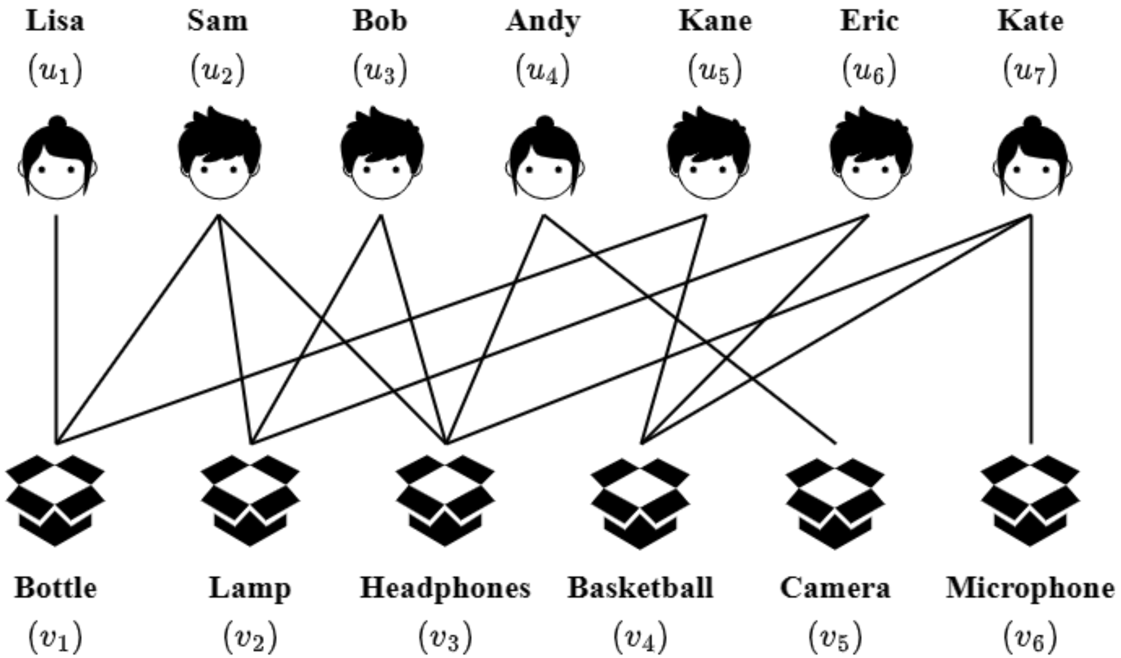
[1] Zhejiang Gongshang University, [2] Alibaba Group, [3] University of Technology Sydney, [4]University of New South Wales, [5] Zhejiang University, [6] Hangzhou Link2Do Technology

KDD 2023

Background

Attributed Bipartite Graph

An attributed bipartite graph is a type of graph which consists of two sets of vertices that are linked by edges. The vertices have additional attributes, making this graph particularly useful for **representing information in the field of e-commerce**.



Background

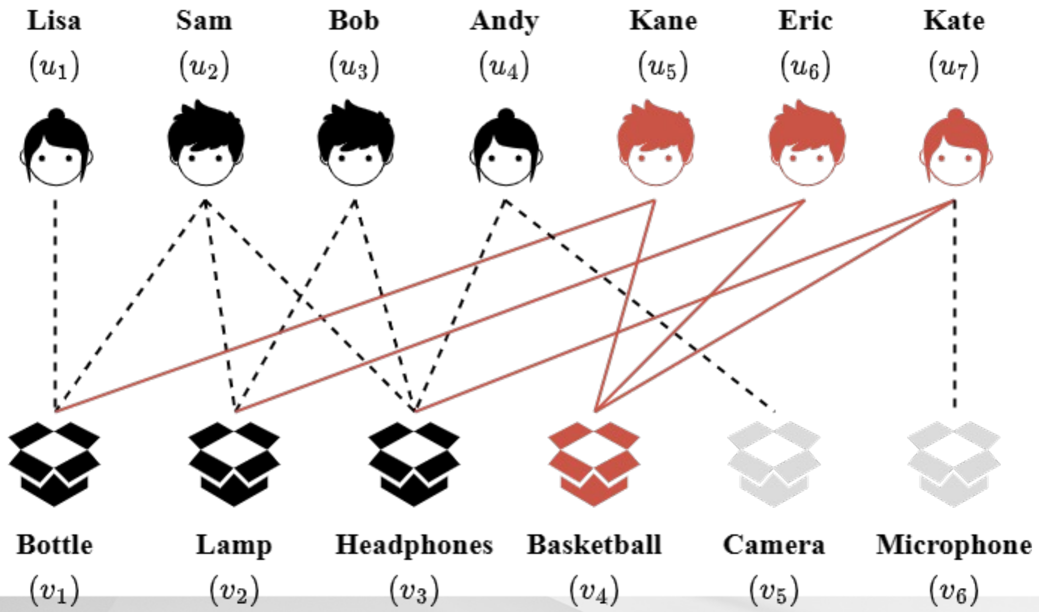
Group-based Frauds on Attributed Bipartite Graphs

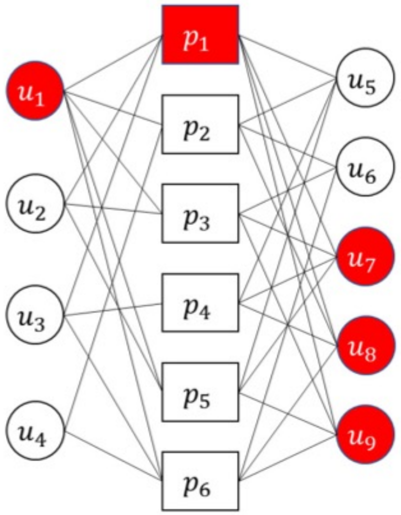
Group-based fraud is becoming increasingly rampant:

“Ride Item’s Coattails” attack (edge classification)

Sockpuppet-based Targeted Attack on Reviewing Systems

(STARS attack) (vertex classification)





Rating	Score	Rating	Score
(u_1, p_1)	1	(u_5, p_5)	0.5
(u_1, p_2)	-1	(u_6, p_2)	0
(u_1, p_3)	0.5	(u_6, p_3)	1
(u_1, p_5)	0.5	(u_6, p_6)	1
(u_1, p_6)	0.5	(u_7, p_1)	1
(u_2, p_1)	0	(u_7, p_3)	1
(u_2, p_3)	1	(u_7, p_4)	1
(u_2, p_5)	1	(u_7, p_6)	0.5
(u_3, p_1)	0	(u_8, p_1)	1
(u_3, p_4)	-0.5	(u_8, p_2)	0
(u_3, p_6)	0.5	(u_8, p_4)	0.5
(u_4, p_2)	-1	(u_8, p_6)	1
(u_4, p_6)	1	(u_9, p_1)	1
(u_5, p_1)	-1	(u_9, p_3)	0.5
(u_5, p_2)	-1	(u_9, p_5)	0.5
(u_5, p_4)	0	(u_9, p_6)	0.5

Image source: STARS: Defending against Sockpuppet-Based Targeted Attacks on Reviewing Systems

Background

SOTA method for “Ride Item’s Coattails” attack

RICD ((α, k1, k2)-biclique): **fraud detection method** for “Ride Item’s Coattails” attack. Can only utilize structural information.

Tianchi competition winner’s algorithm: **classification method**. Can only use attribute information.

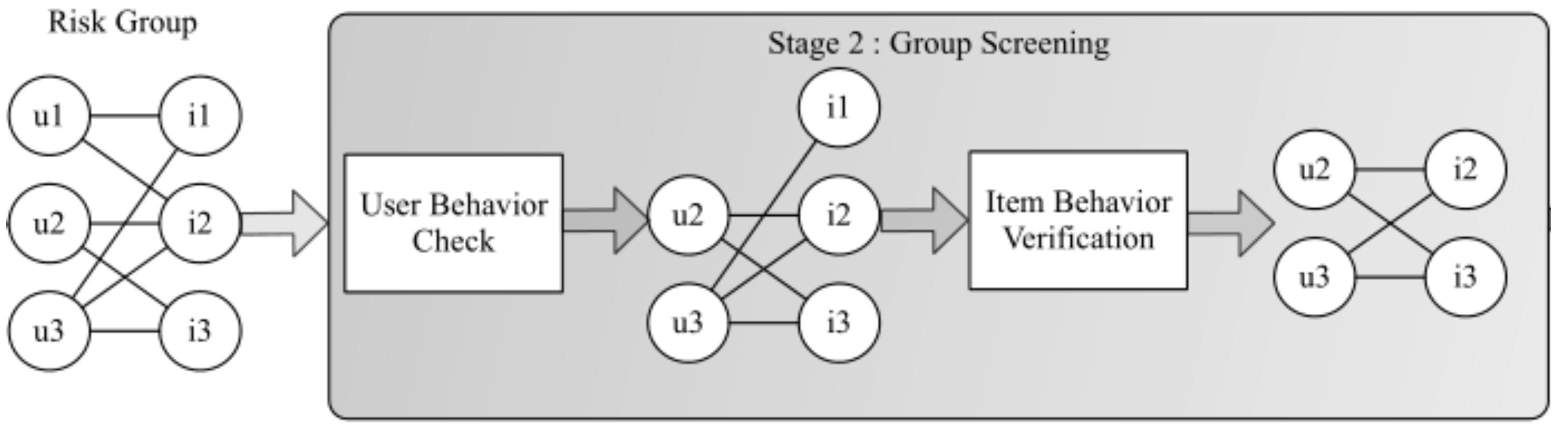


Image source: Large-scale Fake Click Detection for E-commerce Recommendation Systems

Background

SOTA method for STARS attack

RTV: fraud detection method for Sockpuppet-based Targeted Attack on Reviewing Systems (STARS).
Unable to make good use of label information.

	Algorithm RTV **Input:** Rating graph $G = (\mathcal{U} \cup \mathcal{P}, \mathcal{R}, \mathrm{sc})$, weights $\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma_1, \gamma_2, \gamma_3, \gamma_4$, threshold ϵ **Output:** $\mathrm{fair}(u)\ \forall u \in \mathcal{U}$, $\mathrm{good}(p)\ \forall p \in \mathcal{P}$, $\mathrm{rel}(u,p)\ \forall (u,p) \in \mathcal{R}$
1	**for each** $u \in \mathcal{U}$, $\mathrm{fair}_0(u) \leftarrow \mathrm{norm}(u)$
2	**for each** $p \in \mathcal{P}$, $\mathrm{good}_0(p) \leftarrow \mathrm{norm}(p)$
3	**for each** $(u,p) \in \mathcal{R}$, $\mathrm{rel}_0(u,p) \leftarrow \mathrm{norm}(u,p)$
4	$\mu_f \leftarrow \frac{\sum_{u \in \mathcal{U}} \mathrm{fair}_0(u)}{\lvert\mathcal{U}\rvert}$, $\mu_g \leftarrow \frac{\sum_{p \in \mathcal{P}} \mathrm{good}_0(p)}{\lvert\mathcal{P}\rvert}$
5	$t \leftarrow 1$
6	**for each** $u \in \mathcal{U}$, $\mathrm{fair}_t(u) \leftarrow$ value computed as specified in Section 4.1, with $\mathrm{rel}(u,p) = \mathrm{rel}_{t-1}(u,p)$
7	**for each** $p \in \mathcal{P}$, $\mathrm{good}_t(p) \leftarrow$ value computed as specified in Section 4.1, with $\mathrm{rel}(u,p) = \mathrm{rel}_{t-1}(u,p)$
8	**for each** $(u,p) \in \mathcal{R}$, $\mathrm{rel}_t(u,p) \leftarrow$ value computed as specified in Section 4.1, with $\mathrm{fair}(u) = \mathrm{fair}_t(u)$
9	$\Delta \leftarrow \max\left(\sum_{u \in \mathcal{U}} \lvert\mathrm{fair}_t(u) - \mathrm{fair}_{t-1}(u)\rvert, \sum_{p \in \mathcal{P}} \lvert\mathrm{good}_t(p) - \mathrm{good}_{t-1}(p)\rvert, \sum_{(u,p) \in \mathcal{R}} \lvert\mathrm{rel}_t(u,p) - \mathrm{rel}_{t-1}(u,p)\rvert\right)$
10	**if** $\Delta > \epsilon$ **or** $t = 1$ **then** $t \leftarrow t + 1$ and go to Line 6
11	**return** $\mathrm{fair}_t(u)\ \forall u \in \mathcal{U}$, $\mathrm{good}_t(p)\ \forall p \in \mathcal{P}$, $\mathrm{rel}_t(u,p)\ \forall (u,p) \in \mathcal{R}$

Image source: STARS: Defending against Sockpuppet-Based Targeted Attacks on Reviewing Systems

Background

Existing methods

Classification Methods:

- Imbalanced labeled vertices, community information.

Cohesive Subgraph Mining Methods:

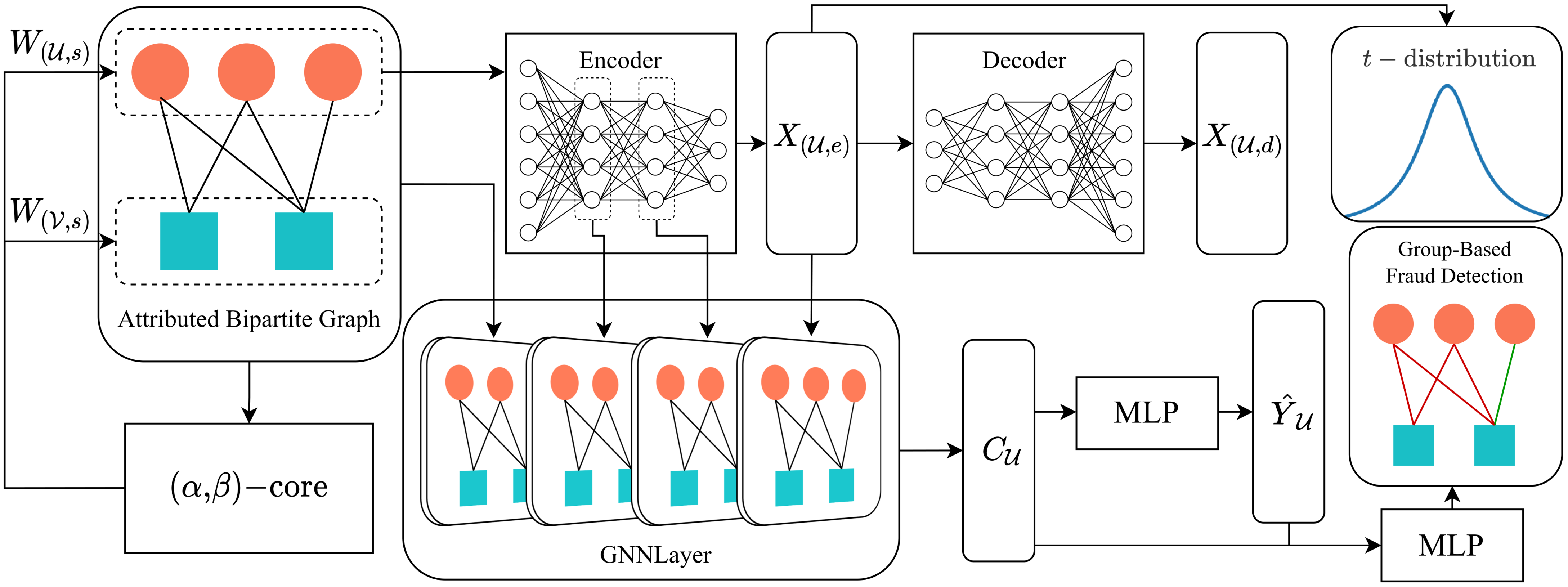
- Attribute and label information, suffer from NP-completeness.

Fraud Detection Methods:

- Global topological and attribute information, label information, manual parameter setting.

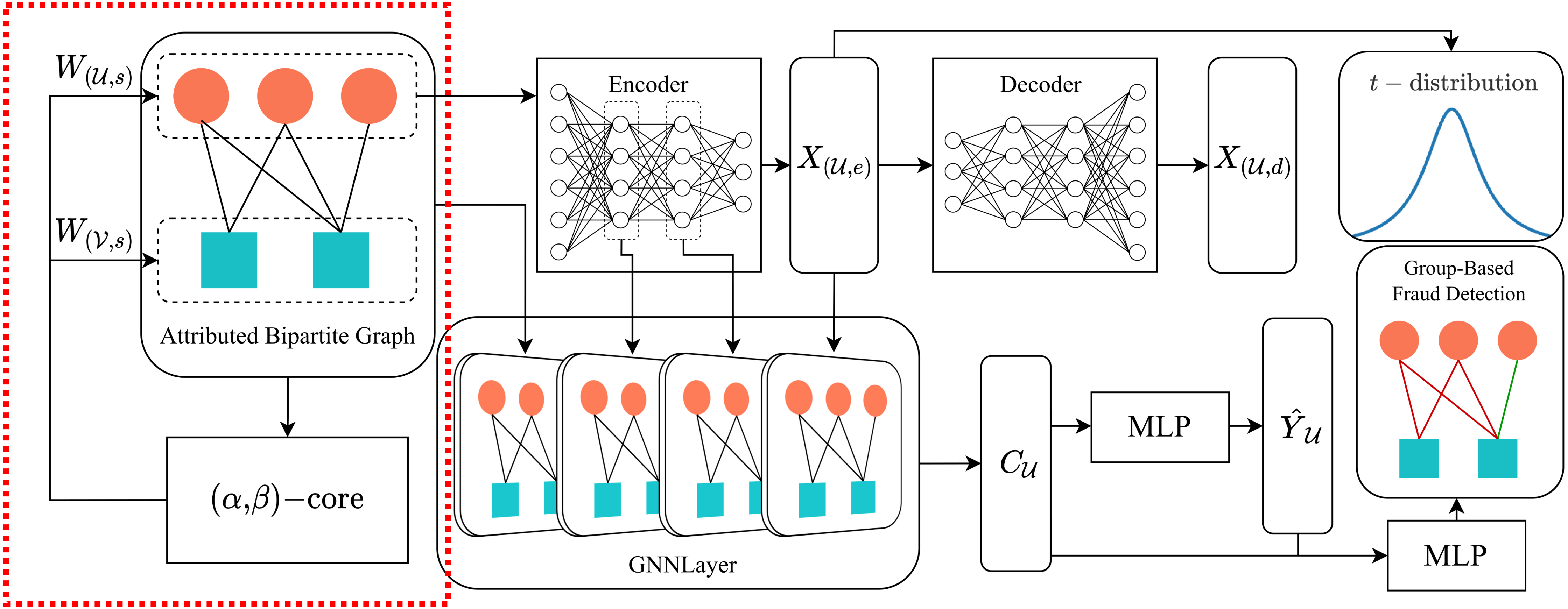
Overview

Group-based Fraud Detection method: GFDN



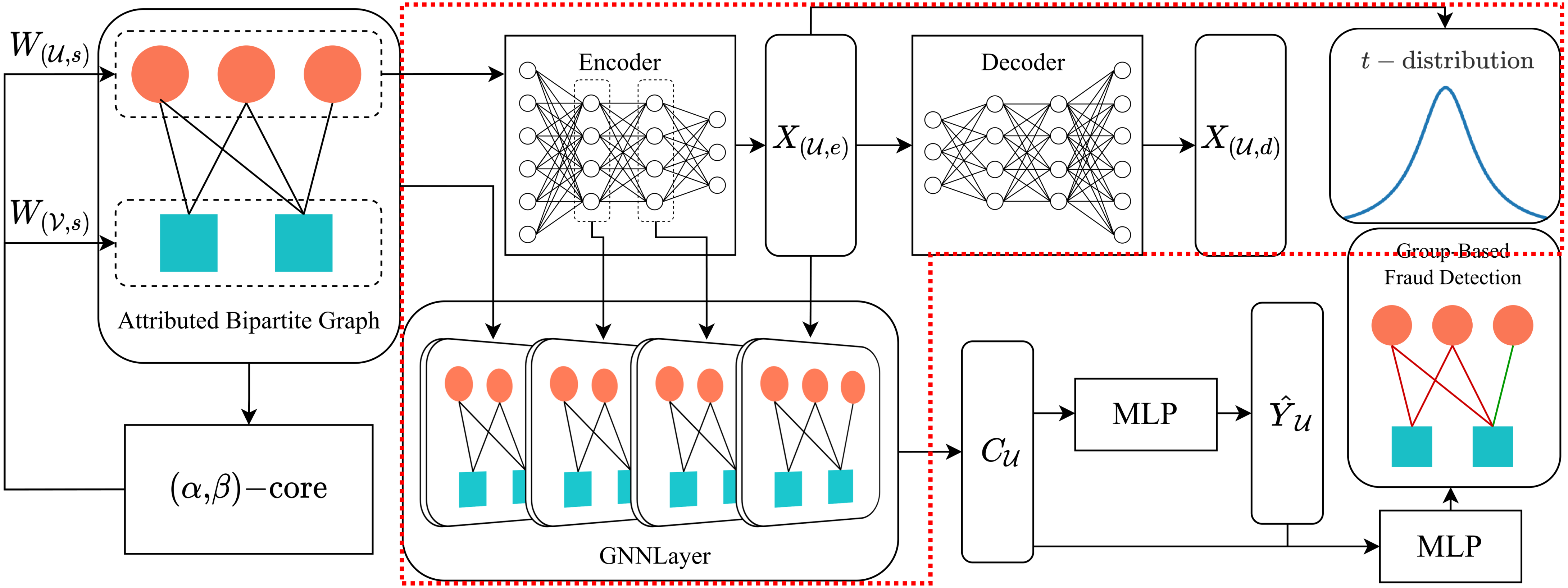
Overview

Group-based Fraud Detection method: GFDN



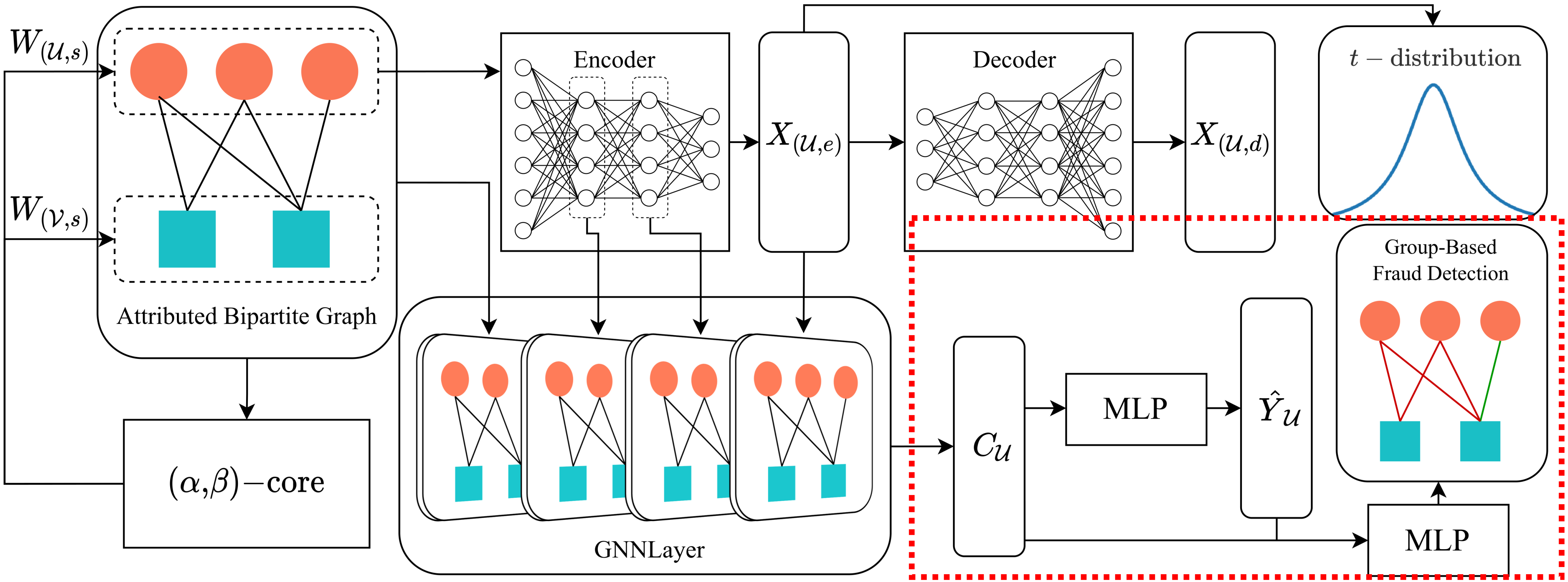
Overview

Group-based Fraud Detection method: GFDN



Overview

Group-based Fraud Detection method: GFDN



GFDN

Structural Feature Initialization

(α, β)-core:
Given a bipartite graph G and integers α, $\beta \in Z^+$, (α, β)-core of G is denoted as G ′ which consists of two vertex sets U′ ⊆ U and V′ ⊆ V.
The (α, β)-core G ′ is a maximal bipartite subgraph induced by U′ ∪ V′ from G in which all the vertices in U′ have degrees at least α and all the vertices in V′ have degrees at least β.

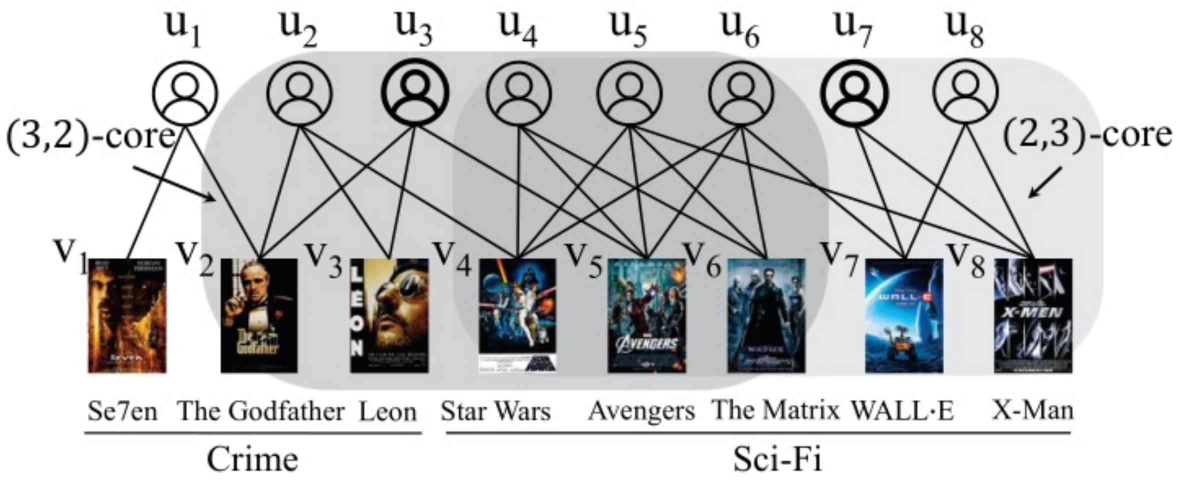


Image source: Efficient (α, β)-core Computation: an Index-based Approach

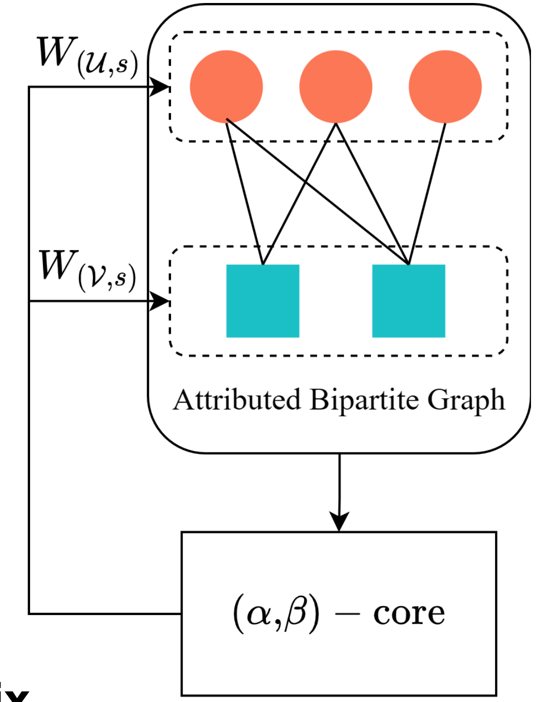
GFDN

Structural Feature Initialization

GFDN will generate structural features for vertices based on their existence in different (α, β)-core.

$$\hat{X}_{(\mathcal{U},s)} = \boldsymbol{X}_{(\mathcal{U},s)} \odot (\boldsymbol{I}_{\mathcal{U}} \boldsymbol{W}_{(\mathcal{U},s)}),\ \hat{X}_{(\mathcal{V},s)} = \boldsymbol{X}_{(\mathcal{V},s)} \odot (\boldsymbol{I}_{\mathcal{V}} \boldsymbol{W}_{(\mathcal{V},s)})$$

Structural Features **Element-wise Product** **All-ones Vector** **Weight Matrix**

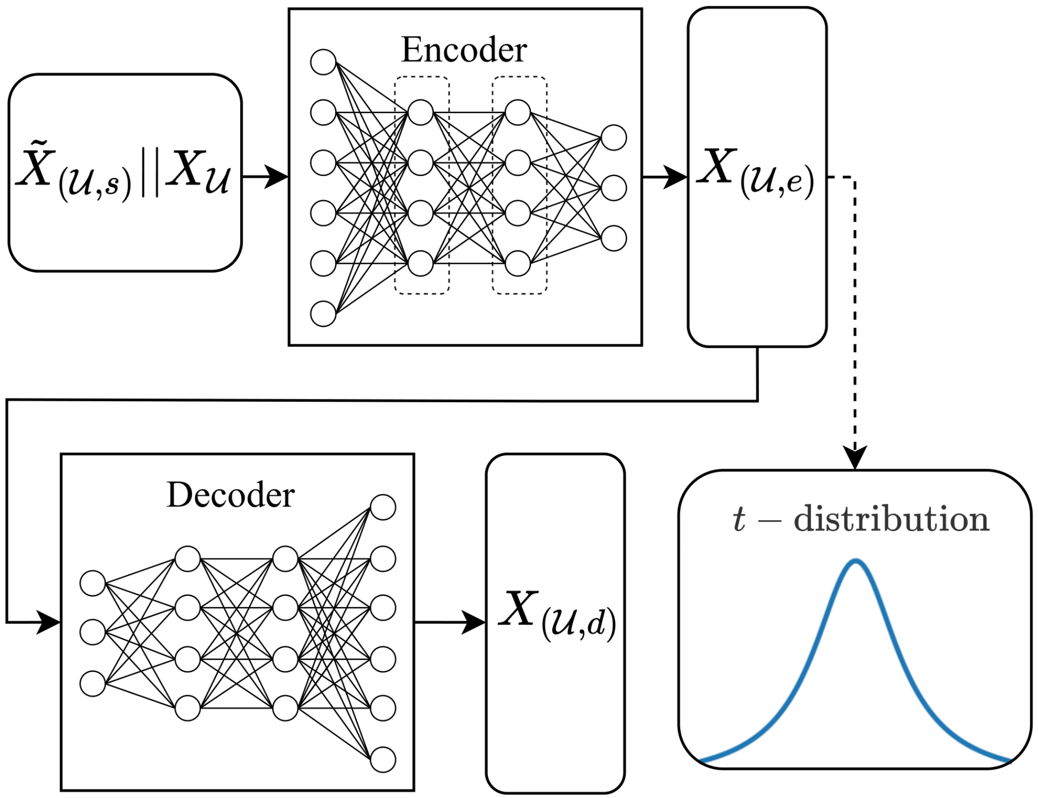


GFDN

Fraudster Community Detection

BDCN - Autoencoder:
Autoencoder in Bipartite Deep Clustering Network (BDCN) can:
1. preserving both structural and attribute information from the input features.
2. Generate high-quality community representation for customer vertices.

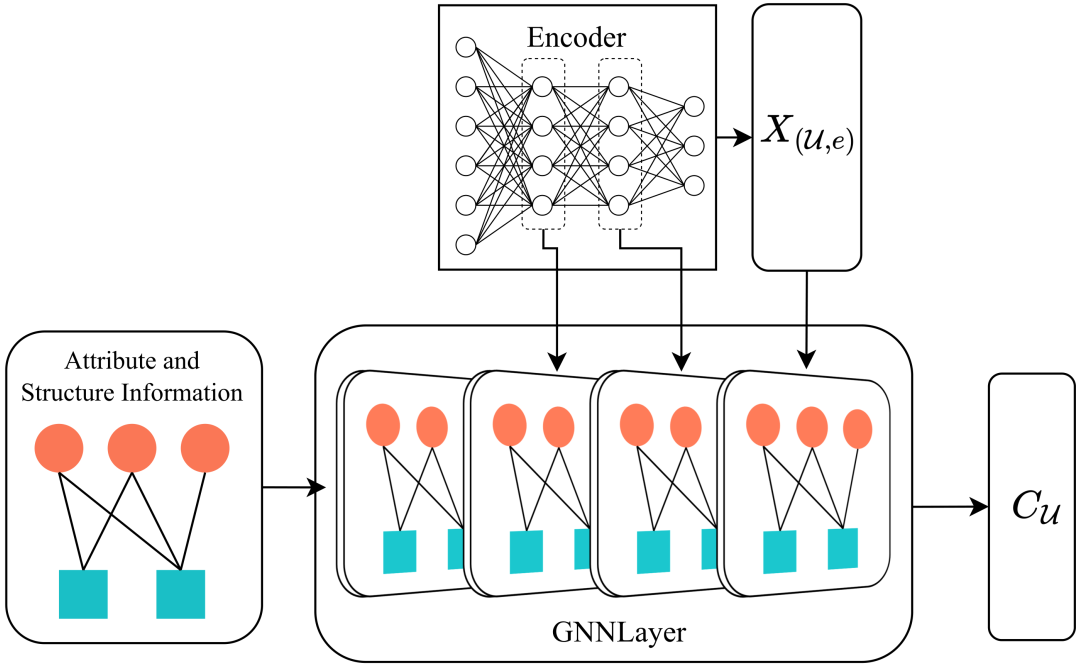
It can achieve self-supervised fraud **community detection** using a loss function measures with Student's t-distribution kernel.



GFDN

Fraudster Community Detection

BDCN - GNN:
GNN in BDCN can aggregate on attribute bipartite graph and preserve the attribute information and structural information of the graph well. The output of each encoding layer will be used.

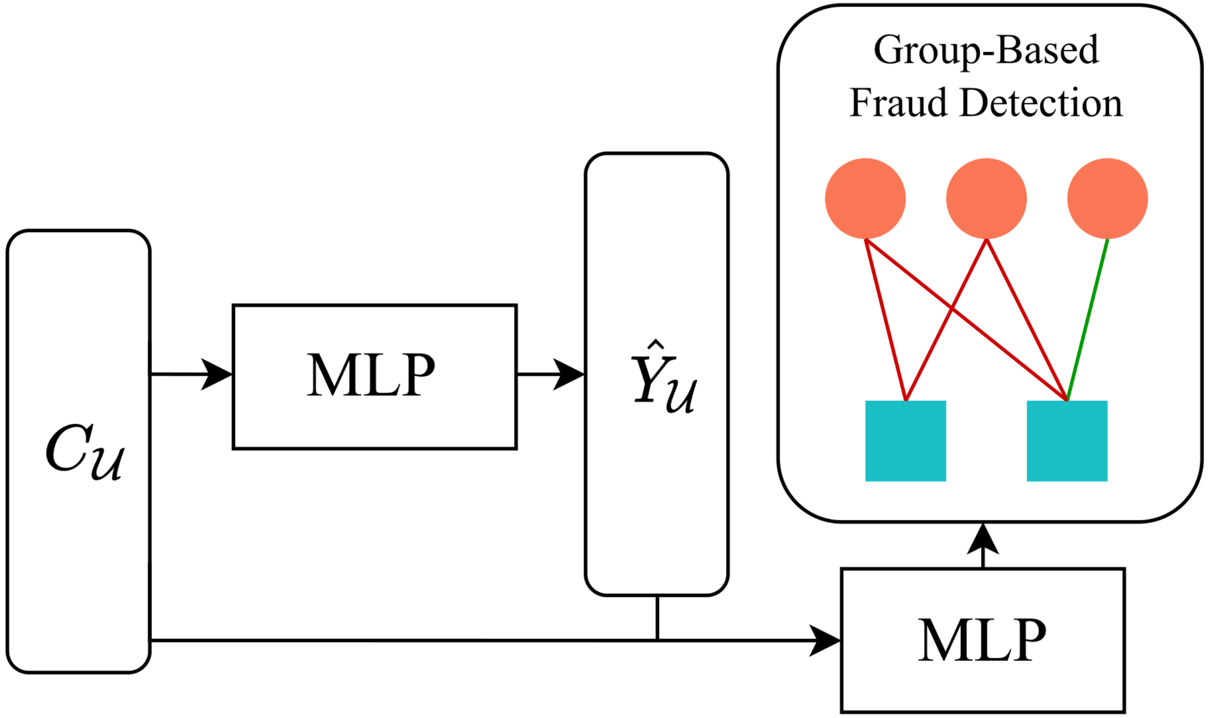


GFDN

Training Objective

"Ride Item's Coattails" Attack:
In "Ride Item's Coattails" attack, not all edges related to fraudsters necessarily have attack implications. GFDN will perform **multi-task training** on this issue, predicting both **fraudsters** and **fraudulent attack**.

STARS Attack:
STARS attack detection aims to **detect fraudsters**, in which case GFDN only needs to perform the vertex classification task.



GFDN

Training Objective

The final loss function will be composed of the loss functions of the aforementioned training objectives, including reconstruction of **autoencoder**, **community prediction**, **fraudster prediction**, and **fraudulent attack prediction**. The sum of the weights of all parts of them is 1.

$$\mathcal{L} = \omega_{ae}\underbrace{\mathcal{L}_{ae}}_{\text{Autoencoder}} + \omega_c\underbrace{\mathcal{L}_c}_{\text{Community}} + \omega_l\underbrace{\mathcal{L}_l}_{\text{Fraudster}} + \omega_e\underbrace{\mathcal{L}_e}_{\text{Fraudulent}}$$

Experiments

Experimental Setup

- **Dataset**
  - 4 real-life datasets.
- **Compared methods**
  - 5 learning-based methods.
  - 2 pattern-based methods.
  - 4 fraud detection methods.
  - A naïve model and four ablated GFDNs
- **Parameter settings**
  - The number of GNN layer: 4.
  - The number of community: 32.
  - Hidden dimension: 128.
  - The selected GNN is GraphSAGE.
- **Implementation**
  - Structure information extraction: C++
  - Other Parts of the Model :Python + Pytorch Geometric.

Table 1: Datasets for "Ride Item's Coattails" Attack Detection

Dataset	$\|\mathcal{E}\|$	$\|\mathcal{U}\|$	$\|\mathcal{V}\|$	% Fraudulent	% Legitimate
TB	3,085,653	996,090	381,611	0.62%	3.53%
TC	1,050,000	532,345	239,840	2.86%	11.43%

Table 2: Datasets for STARS Attack Detection

Dataset	$\|\mathcal{E}\|$	$\|\mathcal{U}\|$	$\|\mathcal{V}\|$	% Fraudulent	% Legitimate
Alpha	24,186	3,286	3,754	3.10%	4.20%
OTC	35,592	4,814	5,858	3.70%	2.80%

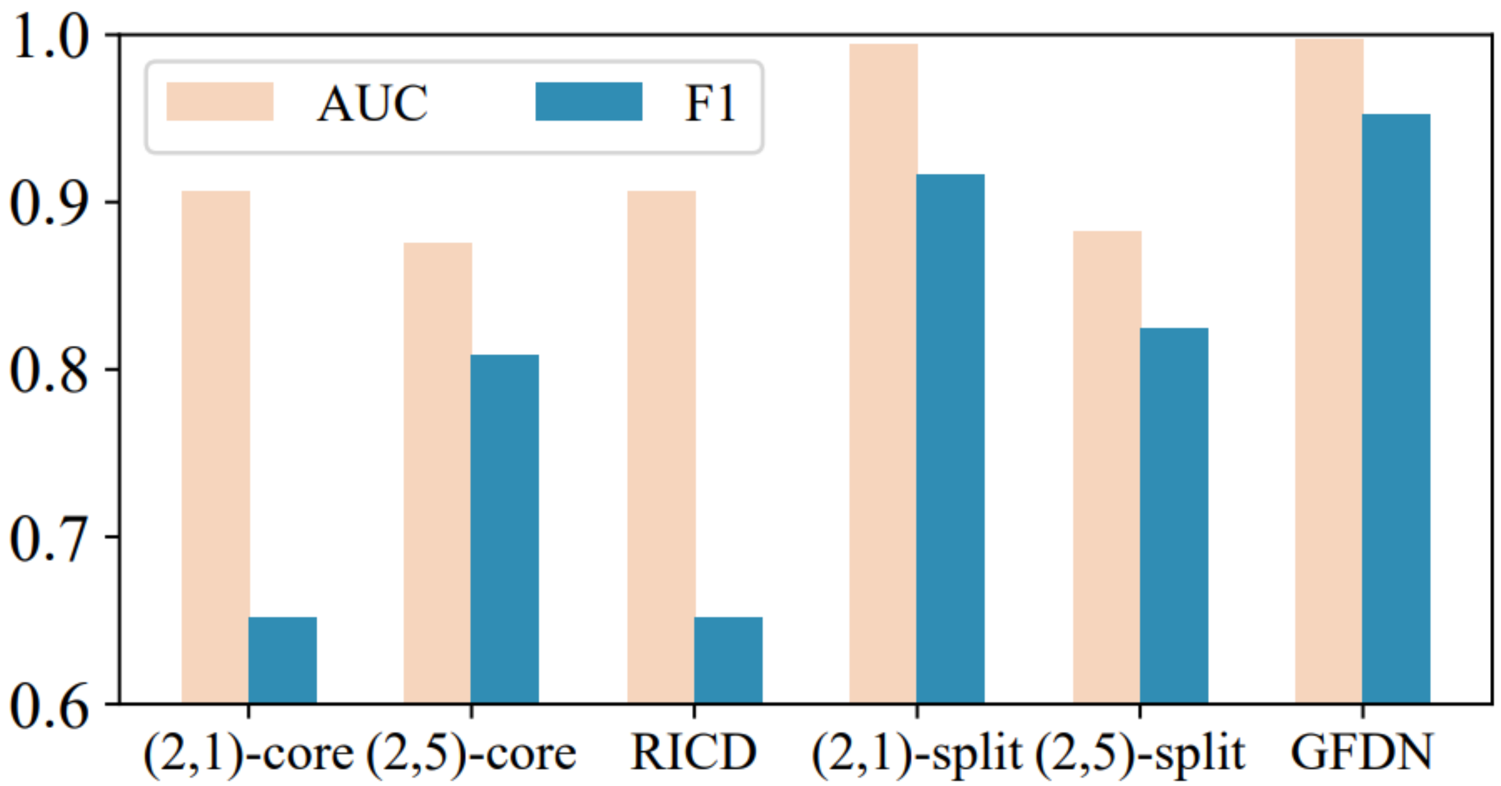
Experiments

Effectiveness Evaluation Results for “Ride Item’s Coattails” Detection

	TB Data					TC Data				
	F1	Acc	AUC	Pre	Recall	F1	Acc	AUC	Pre	Recall
LPA	0.2737	0.4627	0.5517	0.1715	0.6785	0.2056	0.4284	0.5276	0.1219	0.6557
SBGNN	0.4789	0.8228	0.7947	0.4279	0.5438	0.3676	0.8074	0.7666	0.2900	0.5018
BiGI	0.5359	0.8540	0.8491	0.5097	0.5649	0.4039	0.8292	0.8044	0.3331	0.5129
SIHG	0.6449	0.8709	0.8692	0.5470	0.7853	0.5947	0.8771	0.8985	0.4735	0.7992
Tianchi	0.6446	0.8752	0.9342	0.5606	0.7581	0.5364	0.8717	0.9107	0.4527	0.6583
RICD	0.6518	0.8405	0.9063	0.4834	**1.0000**	0.4784	0.8482	0.7474	0.3906	0.6171
(α, β)-core	0.8081	0.9449	0.8757	0.8417	0.7770	0.6348	0.8907	0.8696	0.5093	0.8423
FRAUDAR	0.2580	0.1481	0.4963	0.1483	0.9927	0.2020	0.1124	0.4981	0.1124	**0.9961**
CF1	0.2407	0.7698	0.5532	0.2371	0.2445	0.1620	0.7981	0.5253	0.1523	0.1731
CF2	0.4675	0.7603	0.7376	0.3497	0.7052	0.3588	0.6837	0.7277	0.2326	0.7844
Naive	0.8109	0.9473	0.9844	0.8736	0.7565	0.6397	0.9090	0.9516	**0.7816**	0.5414
GFDN-S	0.6867	0.9202	0.9653	0.8284	0.5864	0.6122	0.8783	0.9342	0.4780	0.8514
GFDN-F	0.9212	0.9754	0.9886	0.8821	0.9639	0.6401	0.8976	0.9287	0.5302	0.8076
GFDN-L	0.9398	0.9813	0.9964	0.9050	0.9775	0.7015	0.9192	0.9654	0.6014	0.8417
GFDN-C	0.9423	0.9821	0.9967	0.9086	0.9785	0.7048	0.9226	0.9646	0.6181	0.8198
GFDN	**0.9522**	**0.9853**	**0.9974**	**0.9254**	0.9806	**0.7226**	**0.9242**	**0.9713**	0.6154	0.8752

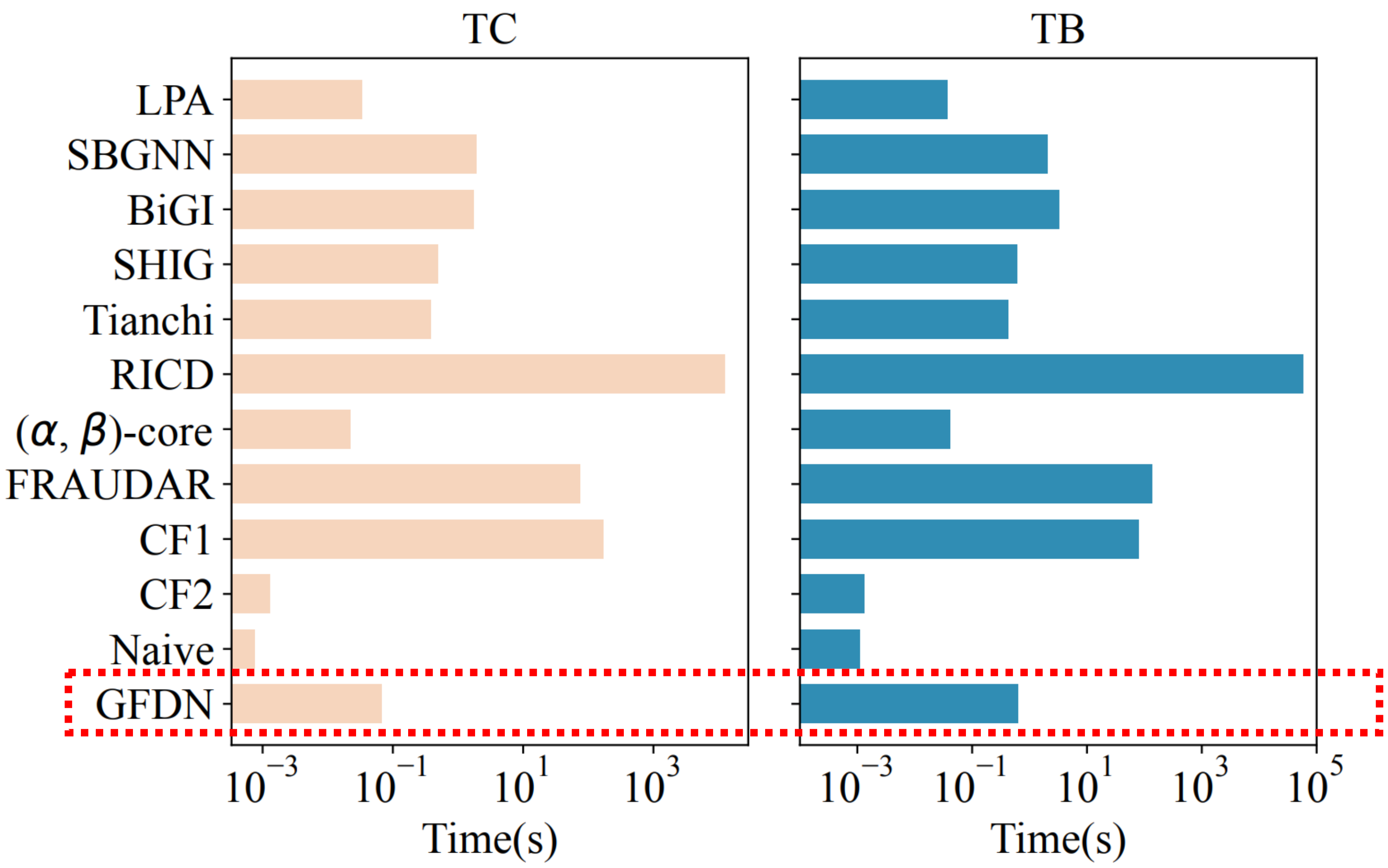
Experiments

Comparison with Pattern-based Algorithms



Experiments

Query Time Evaluation of “Ride Item’s Coattails” Detection



UTS

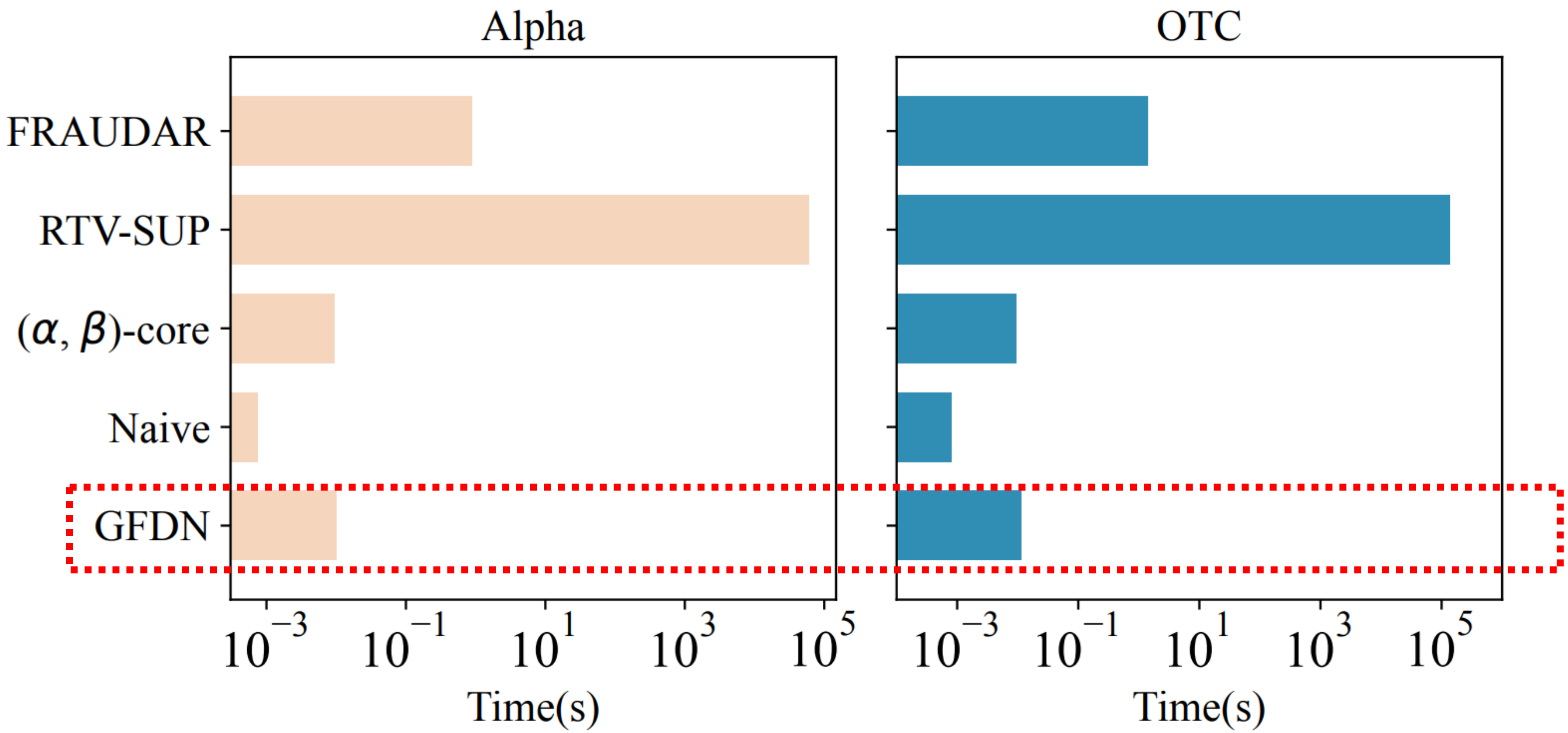
Experiments

Effectiveness Evaluation Results for STARS Detection

	Alpha					OTC				
	F1	Acc	AUC	Pre	Recall	F1	Acc	AUC	Pre	Recall
FRAUDAR	0.3800	0.2626	0.5236	0.2346	**1.0000**	0.3780	0.2547	0.5183	0.2330	**1.0000**
RTV-SUP	0.8652	**0.9452**	0.8859	**0.9747**	0.7778	0.7010	0.8082	0.8736	0.5417	0.9931
(α, β)-core	0.7857	0.8767	0.9204	0.6471	**1.0000**	0.7784	0.8711	0.9167	0.6372	**1.0000**
Naive	0.8089	0.9018	0.9789	0.7222	0.9192	0.7937	0.8978	0.9508	0.7310	0.8681
GFDN	**0.8919**	**0.9452**	**0.9913**	0.8049	**1.0000**	**0.9231**	**0.9623**	**0.9746**	**0.8571**	**1.0000**

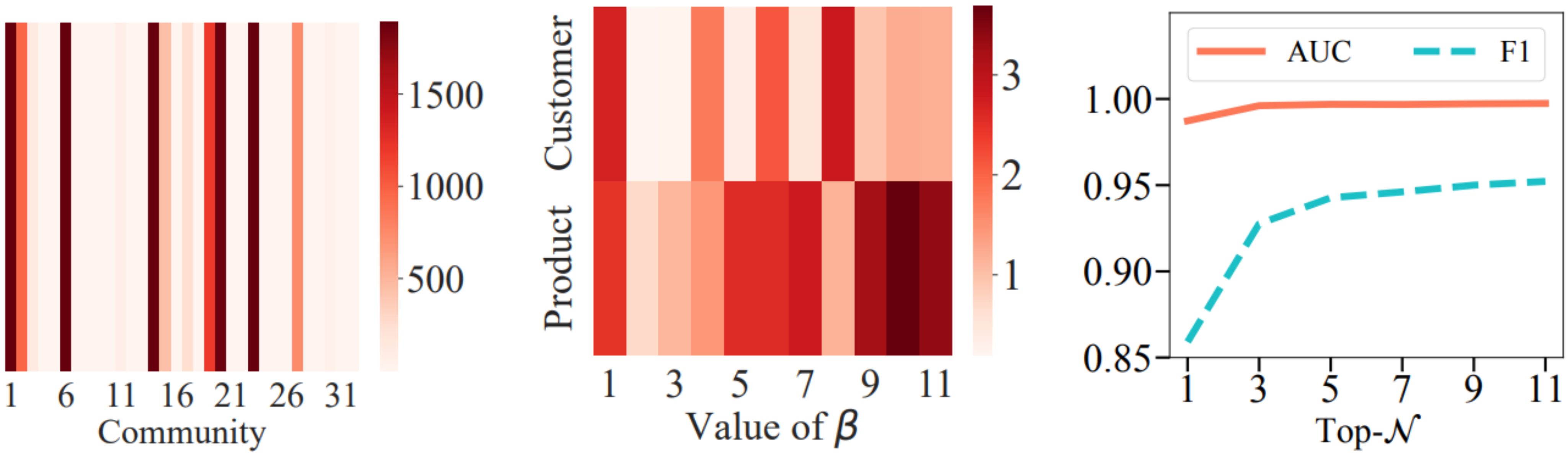
Experiments

Effectiveness Evaluation Results for STARS Detection



Experiments

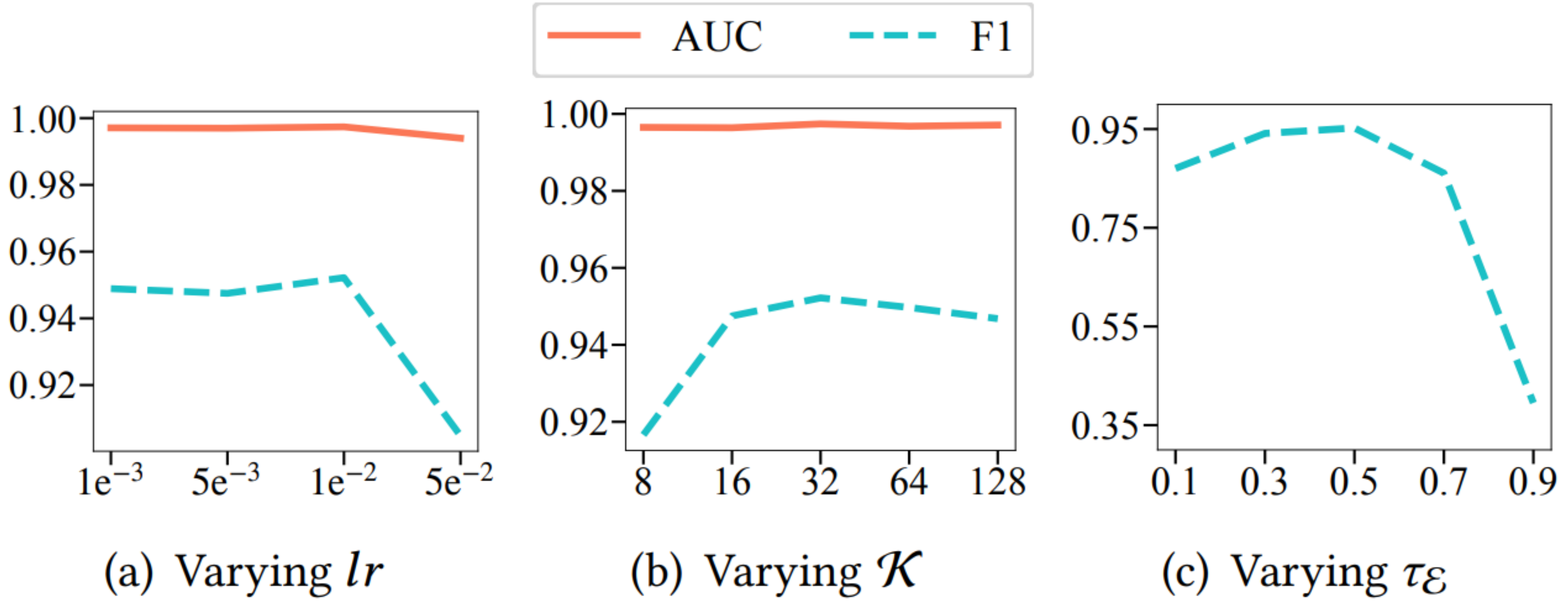
In-Depth Effectiveness Analysis of GFDN



(a) Heatmap of $\boldsymbol{C}_{\mathcal{U}}$ (b) Heatmap of $\boldsymbol{W}_{(\mathcal{U},s)}$, $\boldsymbol{W}_{(\mathcal{V},s)}$ (c) Varying number of β

Experiments

Parameter Analysis Results in GFDN



Thank you!

Jianke Yu
jiankey.zjgsu@gmail.com

Hanchen Wang
hanchen.wang@uts.edu.au