Graph Fraud Detection Based on Accessibility Score Distributions

Minji Yoon ✉

Carnegie Mellon University
minjiy@cs.cmu.edu

Abstract. Graph fraud detection approaches traditionally present frauds as subgraphs and focus on characteristics of the fraudulent subgraphs: unexpectedly high densities or sparse connections with the rest of the graph. However, frauds can easily circumvent such approaches by manipulating their subgraph density or making connections to honest user groups. We focus on a trait that is hard for fraudsters to manipulate: the unidirectionality of communication between honest users and fraudsters. We define an accessibility score to quantify the unidirectionality, then prove the unidirectionality induces skewed accessibility score distributions for fraudsters. We propose SkewA, a novel fraud detection method that measures the skewness in accessibility score distributions and uses it as an honesty metric. SkewA is (a) robust to frauds with low density and various types of camouflages, (b) theoretically sound: we analyze how the unidirectionality brings skewed accessibility score distributions, and (c) effective: showing up to 95.6% accuracy in real-world data where all competitors fail to detect any fraud.

1 Introduction

Various online platforms allow people to share their thoughts and recommend products and services to each other. Users rely on reviews with the belief they are written by disinterested people, thus more objective and unbiased. Fraudsters exploit people's trust on these platforms and derive benefits from fake followers and reviews. These frauds hinder and mislead people's decision making, thus detecting these actions is crucial for companies and customers alike.

Various graph-based approaches have been proposed to detect frauds. Most of them [2,5,7] focus on dense interconnections among fraudsters (dense subblock/subtensor/subgraph). Another popular approach focuses on the isolation of fraud communities [1,11]. However, those methods have vulnerabilities. To evade the density-based methods, frauds generate a number of bot accounts, make their subgraph sparse, and their density low. To circumvent the isolation-based algorithms, frauds camouflage themselves as honest users by writing reviews on normal products or hijacking honest accounts.

In this paper, we focus on a characteristic that is hard for frauds to manipulate: the unidirectionality of communication between honest users and fraudsters. Honest users rarely communicate with fraudsters while fraudsters write reviews

or follow honest users for camouflage. This unidirectionality is generated by honest users, thus hard for fraudsters to manipulate like densities or connections. To quantify the unidirectionality, we first define accessibility scores that estimate how easily other nodes can access a given node (Section 4.1). Fraudsters show skewed accessibility score distributions — high accessibility scores from each other but low accessibility scores from honest users (Section 4.2). We prove this skewness in the accessibility score distributions theoretically and empirically (Section 4.3). Finally, we propose SkewA, a novel approach to detect frauds. SkewA defines a novel metric for honesty that measures the skewness then spots frauds with lowest honesty scores (Section 4.4). Through extensive experiments, we demonstrate the superior performance of SkewA over existing methods.

The main contributions of this paper are as follows:

- **Insight:** The unidirectionality of communication results in skewness in accessibility score distributions for fraudsters: high scores on fraud groups and low scores on honest groups.
- **Robustness:** SkewA is based on the unidirectionality generated by honest users, thus hard for fraudsters to manipulate.
- **Theoretical guarantees:** SkewA proves how the skewed accessibility score distributions are generated and preserved under camouflages.
- **Effectiveness:** SkewA presents up to 95.6% accuracy in public benchmarks, where all competitors fail to detect any fraud.

Reproducibility: our code is publicly available [1].

2 Related Work

Graph fraud detection algorithms could be classified into supervised and unsupervised methods based on whether a method requires labels of fraudulent or benign users/products. See [1] for an extensive survey.

Supervised methods model a fraud detection task as a binary classification problem for nodes on graphs. [3,16] leverage either labeled normal nodes or labeled fraudulent nodes. They exploit random walks to propagate the initial normalness/badness scores to the remaining nodes. [6,14,15] leverage both fraudulent and normal users. [6] is based on random walks, while [15] exploits pairwise Markov Random Field (pMRF). GANG [14] leverages pMRF and Loopy Belief Propagation to detect fraudsters.

Unsupervised methods measure suspicious scores based on graph topology. [13] factorizes the adjacency matrix and flags edges, which introduce high reconstruction error as outliers. SpokEN [11] and [12] focuses on singular vectors of a graph, which are clearly separated when plotted against each other. Fraudar [5] adapts the theoretical perspective to fraud detection and camouflage resistance and achieves meaningful bounds for applications. DeFraudar [4] presents six

[1] `https://github.com/minjiyoon/PKDD21-SkewA`

Table 1. Table of symbols.

Symbol	Definition
G	Bipartite graph $G = (V, E)$
n_1, n_2	Numbers of products and users in G
m	Number of edges in G
$\tilde{\mathbf{A}}_{\mathbf{C}}$	$(n_1 \times n_2)$ column-normalized adjacency matrix
$\tilde{\mathbf{A}}_{\mathbf{R}}$	$(n_2 \times n_1)$ column-normalized adjacency matrix
c	Restart probability of RWR
$\mathbf{b}$	$(n_1 \times 1)$ starting vector of RWR

Table 2. Comparison between methods.

Property \ Method	GANG [14]	HoloScope [9]	SpokEN [11]	DeFraudar [4]	Fraudar [5]	**SkewA**
Unsupervised		✓	✓	✓	✓	✓
Robust to density						✓
Camouflage-resistant	✓	?		?	✓	✓
Theoretical guarantees					✓	✓

fraud indicators that measure the spamicity of a group. HoloScope [9] penalizes nodes with many connections from other nodes based on the unidirectional communication between fraudulent and honest users.

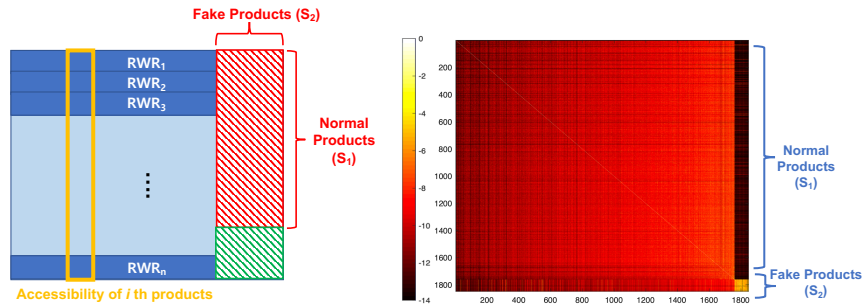
Several methods have used PageRank or Random Walk to detect frauds. However, most of them [6,15] are supervised learning requiring labels to assign initial scores to propagates. One of our design goals is to avoid the requirement for sources other than the graph topology to measure anomalousness. [9] and [14] exploit the unidirectional communication between honest users and frauds, but both lack theoretical guarantees on how their metric preserves the unidirectionality under fraud's camouflage. In this paper, we propose an unsupervised fraud detection method SkewA with theoretical analysis on robustness to fraud's camouflage. Table 2 compares SkewA to existing methods.

3 Preliminaries

We review Random Walk with Restart (RWR) [10] which is used in accessibility score computation then describe how to compute RWR in a bipartite graph.

3.1 Random Walk with Restart

RWR measures each node's relevance w.r.t. a seed node s in a graph. It assumes a random walker starting from s, who traverses edges in the graph with probability



(a) Theoretical accessibility matrix (b) Experimental accessibility matrix

Fig. 1. In an RWR matrix stacking n RWR row vectors, each column corresponds to an accessibility column vector.

$1-c$ and occasionally restarts at the seed node s with probability c. Then the frequency of visitation of the walker on each node becomes its relevance score w.r.t. the seed node. From [17], the RWR score vector $\mathbf{r}_{\text{RWR}}$ is presented as $\mathbf{r}_{\text{RWR}} = c\sum_{i=0}^{\infty}\left((1-c)\tilde{\mathbf{A}}\right)^i\mathbf{b}$ where $\tilde{\mathbf{A}}$ is the column-normalized adjacency matrix, c is the restart probability and $\mathbf{b}$ is the seed vector with the seed node's index s set to 1 and others to 0. If $0 < c < 1$ and $\tilde{\mathbf{A}}$ is irreducible and aperiodic, $\mathbf{r}_{\text{RWR}}$ is guaranteed to converge to a unique solution [8].

3.2 RWR for Bipartite Graphs

In a bipartite graph, we have two adjacency matrices, $\mathbf{A_C}$ and $\mathbf{A_R}$, which are transpose to each other. $\mathbf{A_C}$ puts products in its rows and users in its columns, while $\mathbf{A_R}$ puts users in its rows and products in its columns. $\mathbf{A_C}(i,j)$ and $\mathbf{A_R}(j,i)$ are set to 1 when j-th user writes a review on i-th product and 0 otherwise. Then $\tilde{\mathbf{A}}_\mathbf{C}$ and $\tilde{\mathbf{A}}_\mathbf{R}$ become column-normalized $(n_1 \times n_2)$ and $(n_2 \times n_1)$ matrices where n_1 and n_2 denote the total numbers of products and users, respectively. One iteration in RWR computation in a unipartite graph is divided into two sub-steps in a bipartite graph. From the original equation, we replace $\tilde{\mathbf{A}}$ with $\tilde{\mathbf{A}}_\mathbf{C}\tilde{\mathbf{A}}_\mathbf{R}$. By multiplying with $\tilde{\mathbf{A}}_\mathbf{R}$, scores are propagated from products to users. Then, by multiplying with $\tilde{\mathbf{A}}_\mathbf{C}$, the scores are propagated from the user nodes back to the product nodes. Other components are identical to the regular RWR computation.

4 Proposed Method

On a review website, fraudsters write a number of reviews on normal products to disguise themselves as honest users. In contrast, normal users purchase and review fake products only accidentally. When abstracting this phenomenon to a user-product bipartite graph, fraudulent user nodes are connected to normal

product nodes, while honest user nodes rarely make connections to fake product nodes. This **unidirectionality of communication** is decided by honest users; thus, frauds cannot manipulate or dissimulate it. Based on this unidirectionality, we propose a robust fraud detection method SkewA.

To quantify the unidirectionality, we first define accessibility scores for each node as how easily other nodes could reach to the node (Section 4.1). Then we show how the unidirectionality of communication leads to the skewed accessibility score distributions for fraudsters (Sections 4.2 and 4.3). Finally, we propose our novel algorithm SkewA to detect frauds (Section 4.4).

4.1 Accessibility

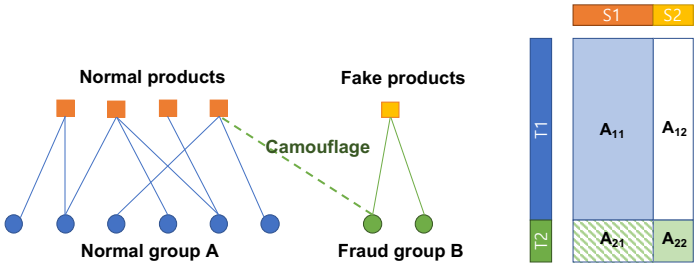
RWR scores with seed node i measure how easily the seed node i could reach other nodes. The scores are measured in the perspective of the seed node; thus easily manipulated by the seed node by adding edges to target nodes to increase their RWR scores. Here we define accessibility scores that measure how easily other nodes could reach the seed node i. The accessibility scores appear to be identical to the RWR scores at first glance. However, the probability of crossing an edge (i, j) from node i is different from the probability of crossing the same edge from the node j. When source node i has a larger number of out-edges than target node j, the probability of crossing the edge (i, j) is smaller since a random walker has more options to choose. This results in the different RWR and accessibility scores for each target node given the same seed node. Contrary to RWR scores, accessibility scores are estimated by target nodes and hard for the seed node to control. This explains why we choose accessibility scores as a measurement for detecting frauds.

Definition 1 (Accessibility score vector). *In an n-dimensional accessibility score vector of node i, the j-th component contains the probability that a random walker starts from node j and reaches node i.*

Accessibility score computation is based on RWR computation. We vertically stack n RWR row vectors with n different seed nodes (Figure 1(a)). Then the i-th column in this $(n \times n)$ matrix becomes an accessibility score vector for node i, presenting how easily other nodes could reach to node i. We exploit that the accessibility score matrix is the transpose of the RWR score matrix.

4.2 Skewness in Accessibility Score Distributions

In Figure 2(a), a graph is partitioned into two disjoint groups, the honest group A and the fraud group B. The honest group A has the most nodes and edges of the graph. Then, fraudsters in B add a few edges towards the normal group A to camouflage themselves as honest users (green dashed line). With the camouflage edges, crossing these two communities becomes possible for a random walker. However, the possibilities for the walker to pass from A to B is still small: a larger number of edges in A implies more options for the random walker to choose; thus,



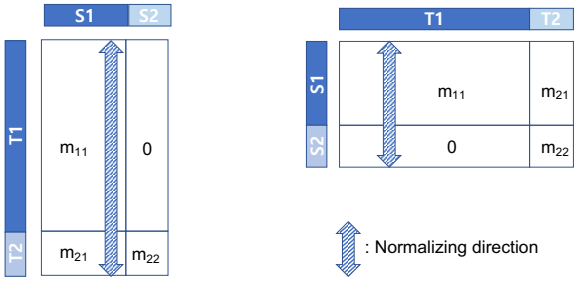
(a) Bipartite graph with frauds (b) Adjacency matrix

Fig. 2. User-product bipartite graph: an edge is generated when a user writes a review on a product.

the random walker starting from A is more likely to select the honest edges (blue line) than the camouflage edges (green dashed line). In short, honest users in the group A are less likely to reach out to a fraudster in the group B, resulting in low accessibility scores. In contrast, fraud colleagues in group B access to the target fraudster easily with help of dense interconnection (green line). Then the fraud colleagues have high accessibility scores. This pattern results in the skewness in the accessibility score distributions for the fraudster: low scores from the honest group while high scores from the fraudulent group. On the other hand, honest users have weak skewness in accessibility score distributions. A random walker starting from fraud group B is more likely to choose the camouflage edges (green dashed line) than a walker starting from group A because group B has fewer inter-connected edges than group A. This pattern brings the moderate accessibility scores from B to A, thus less skewed distributions for honest users.

4.3 Theoretical Analysis

In this Section, we prove how skewness is generated in accessibility score distributions of frauds and preserved under the camouflage of the frauds. In Figure 2(b), S_1 (orange part in X-axis) indicates the normal products while S_2 (yellow part in X-axis) denotes the fake products for which fraudsters write fake reviews. T_1 (blue part in Y-axis) denotes the honest users while T_2 (green part in Y-axis) denotes the fraudsters. In an $(n_2 \times n_1)$ adjacency matrix $\mathbf{A}$, $\mathbf{A}_{11}$ (blue part in the matrix) corresponds to edges (reviews) between honest users and normal products. $\mathbf{A}_{22}$ (plain green part in the matrix) contains edges from fraudsters to their target products; we call these edges fake edges. $\mathbf{A}_{22}$ is dense due to a large number of fake reviews. $\mathbf{A}_{21}$ (hatched green part) corresponds to camouflage edges from fraudsters to normal products. Finally, $\mathbf{A}_{12}$ contains reviews written by honest users on fake products. $\mathbf{A}_{12}$ has almost no edge since honest users purchase fake products only accidentally. m_{ij} denotes the total number of edges in the sub-block $\mathbf{A}_{ij}$ where $i, j \in 0, 1$.



(a) A_R (b) A_C

Fig. 3. A_R and A_C are column-normalized adjacency matrices of a bipartite graph.

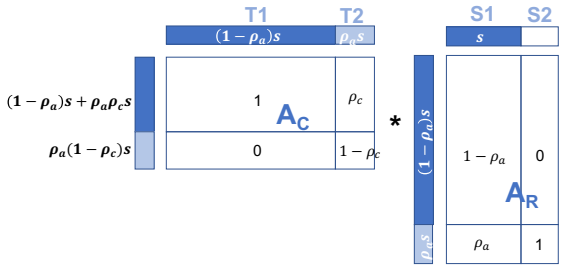
We analyze accessibility score distributions based on the RWR computation — accessibility score vectors are computed from columns of the corresponding RWR matrix. When a vector is multiplied with a column-normalized adjacency matrix, the amount of scores in the input vector is preserved in the output vector. Based on this characteristic, we model the ratio of propagated scores in the output vector as follows:

Assumption 1 (Ratio of Propagated Scores) *I denotes a group of nodes in an input vector, while O_1 and O_2 denote two disjoint groups in an output vector. The numbers of edges from I to O_1 and O_2 are m_1 and m_2, respectively. When I with total scores s is multiplied with a column-normalized matrix, O_1 receives $\frac{m_1}{m_1+m_2}s$ while O_2 receives the remaining $\frac{m_2}{m_1+m_2}s$.*

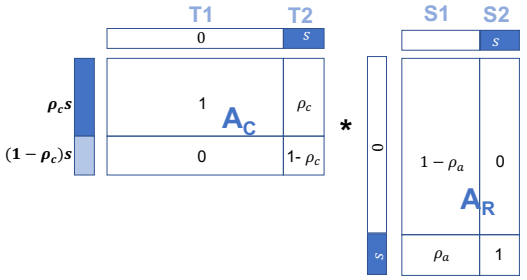
Based on Assumption 1, when S_1 starts with total scores s, T_1 receives $\frac{m_{11}}{m_{11}+m_{21}}s$ while T_2 receives the remaining $\frac{m_{21}}{m_{11}+m_{21}}s$ (Figure 3(a)). Similarly, when S_2 starts with total scores s, T_1 receives $\frac{m_{12}}{m_{12}+m_{22}}s$ while T_2 receives the remaining $\frac{m_{22}}{m_{12}+m_{22}}s$. However, since $m_{12} \approx 0$ (honest users rarely purchase fake products), T_2 receives the whole score s from S_2. Under the same assumption, when T_1 starts with total scores s, S_1 receives $\frac{m_{11}}{m_{11}+m_{12}}s$ while S_2 receives the remaining $\frac{m_{12}}{m_{11}+m_{12}}s$ (Figure 3(b)). However, since $m_{12} \approx 0$, S_1 receives the whole score s from T_1. Similarly, when T_2 starts with total scores s, S_1 receives $\frac{m_{21}}{m_{21}+m_{22}}s$ while S_2 receives the remaining $\frac{m_{22}}{m_{21}+m_{22}}s$.

We show the effectiveness of Assumption 1 empirically on real-world data in Section 5. In the following Section, we analyze the ratio of propagated scores after two sub-steps of RWR computation varying the location of a seed node. We define two ratio parameters: the ratio of camouflage edges to honest edges $\rho_a = \frac{m_{21}}{m_{11}+m_{21}}$, and the ratio of camouflage edges to fake edges $\rho_c = \frac{m_{21}}{m_{21}+m_{22}}$.

Seed Node from Normal Products (S_1): In Figure 4(a), by multiplying with A_R, score s from S_1 is propagated into T_1 and T_2 with scores $(1-\rho_a)s$ and $\rho_a s$, respectively. Then these scores are propagated back to group S_1 and S_2 by multiplying with A_C. All scores $(1-\rho_a)s$ in group T_1 are propagated into only group S_1, while score $\rho_a s$ in group T_2 is divided into $\rho_a\rho_c s$ and $\rho_a(1-\rho_c)s$ and



(a) Score Propagation from Nomal Products



(b) Score Progatation from Fake Products

Fig. 4. Two sub-steps in score propagation.

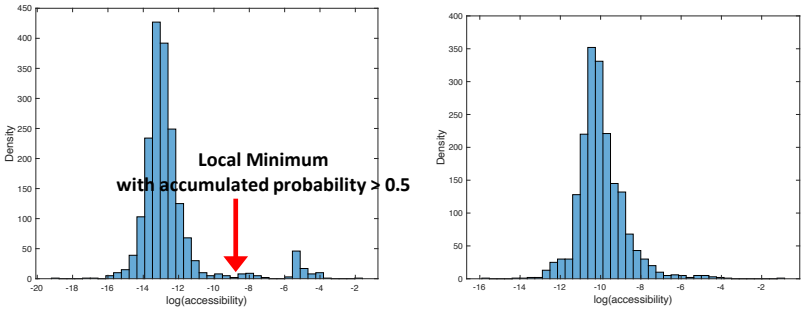
propagated into group S_1 and S_2, respectively. In short, score s starting from normal product group $S1$ will be propagated into S_1 with $(1-\rho_a)s+\rho_a\rho_c s$ and S_2 with $\rho_a(1-\rho_c)s$ after two sub-steps in one iteration of RWR computation.

Seed Node from Fake Products (S_2): In Figure 4(b), by multiplying with A_R, score s from S_2 is propagated into only T_2. Then, by multiplying with A_C, the score s in T_2 is propagated back to S_1 and S_2 with $\rho_c s$ and $(1-\rho_c)s$, respectively. In summary, score s starting from the fake products $S2$ is propagated into S_1 with the score $\rho_c s$ and S_2 with the score $(1-\rho_c)s$ after one iteration of RWR computation.

Ratio of Propagated Scores after One RWR Iteration: Score $s_1(k)$ and $s_2(k)$ denote scores propagated into group S_1 and S_2 at the k-th iteration of RWR computation. When the seed node is located at S_1, $s_1(0)=1$ and $s_2(0)=0$. Otherwise, $s_1(0)=0$ and $s_2(0)=1$. We present $s_1(k)$ and $s_2(k)$ in the iterative equation forms as follows:

Theorem 2 (Ratio of Propagated Scores). *Given ratio of camouflage edges to honest edges ρ_a and ratio of camouflage edges to fake edges ρ_c, scores propagated into group S_1 and S_2 at the k-th iteration of RWR computation are:*

$$s_1(k) = (1-\rho_a)s_1(k-1) + \rho_a\rho_c s_1(k-1) + \rho_c s_2(k-1)$$
$$s_2(k) = \rho_a(1-\rho_c)s_1(k-1) + (1-\rho_c)s_2(k-1)$$



(a) Fraudster (b) Honest user

Fig. 5. Probability density function of accessibility scores.

Proof. $s_1(k)$ *and* $s_2(k)$ *are the sum of scores propagated from* $s_1(k-1)$ *and* $s_2(k-1)$ *to each group, respectively. We simply apply the same rule as above.*

Camouflage edges generated by frauds are much fewer than the total number of edges in real-world graphs, thus $\rho_a = \frac{m_{21}}{m_{11}+m_{21}}$ has small values ($\rho_a << 1$). Then Theorem 2 is approximated as follows:

$$s_1(k) \approx s_1(k-1) + \rho_c s_2(k-1)$$
$$s_2(k) \approx (1-\rho_c) s_2(k-1)$$

When a seed node is located in S_1 ($s_1(0) = 1, s_2(0) = 0$), S_2 rarely receives scores ($s_2(k) \approx 0$). In other words, the accessibility scores from S_1 to S_2 are small. On the other hand, when a seed score is located in S_2 ($s_1(0) = 0, s_2(0) = 1$), S_2 receives large scores, resulting in high accessibility scores from S_2 to S_2. Then the fraud group (S_2) has skewed accessibility score distributions: small scores from the honest group (S_1) while large scores from the fraud group (S_2).

Real-world Graphs: We reproduce our theoretical analysis on the Tripadvisor dataset. We inject a fraudulent block with size of 5% of total users and products. We inject fake edges randomly to the block with 5% density, then add camouflage edges amounting to 10% of the fake edges. Figure 1(b) shows the resulting accessibility score matrix. The last 90 columns correspond to the accessibility score vectors of the injected fraud group and show clear skewness: low scores (dark-colored) for normal products and high scores (blight-colored) for fake products as we analyzed. Figure 5 shows two sampled distributions from the same dataset. In a fraudster's distribution (Figure 5(a)), the neighbor group has high scores around e^{-5}, while the stranger group has low scores around e^{-13}. On the other hand, the distribution of an honest user (Figure 5(b)) is less skewed with majority gathered around e^{-10}. This shows the effectiveness of our theoretical analysis on the real-world graph — for fraudulent nodes, skewness in the accessibility distribution between two groups is apparent; for honest nodes, there is no clear disparity in accessibility scores between the neighbor and stranger groups.

4.4 SkewA

Based on skewness in accessibility score distributions, we propose a fraud detection method SkewA. SkewA first divides a graph into two groups, neighbor and stranger groups for each node. Then SkewA defines a novel honesty metric which measures how accessibility scores are distributed across the neighbor and stranger groups. SkewA spots fraudsters with the lowest honesty scores.

Algorithm 1: SkewA

Input: A bipartite graph G, Top k
Output: k fraudsters
Compute accessibility score matrix $\mathbf{A}_{acc}$;
Compute $\alpha = log(\frac{m}{n_1})$;
foreach *column vector* $\mathbf{a}$ *in* $\mathbf{A}_{acc}$ **do**
  ComputeHonesty($\mathbf{a}$, α)
return k *nodes with lowest honesty scores*

Algorithm 2: ComputeHonesty

Input: Accessibility score vector $\mathbf{a}$, parameter α
Output: Honesty score s_{honest}
Find local minimum in pdf;
Divide into S_1 and S_2 by the local minimum;
Compute sum and variance of S_1 and S_2;
$s_{\text{honest}} = (\text{var}_1\text{var}_2)^{\frac{\alpha}{2}}(\text{sum}_2)^{-\frac{2}{\alpha}}$;
return s_{honest}

Clustering We divide nodes into the neighbor and stranger groups based on the probability density function (pdf) of the accessibility score distribution (Figure 5). We first find local minimums in pdf whose accumulated probabilities from zero are larger than 0.5 then choose the one who has the smallest accessibility score. Based on the local minimum, we partition nodes into two groups, those accessibility scores are less or greater than the score of the minimum, then classify them as stranger and neighbor groups, respectively. We exploit that the neighbor group has high accessibility scores, while the stranger group has low scores. We consider the local minimums whose accumulated probabilities are larger than 0.5 because the neighbor group is smaller than half of the graph.

Metric for Honesty Given the stranger and neighbor groups, we measure sum_1, var_1 and sum_2, var_2 denoting sum and variance of stranger and neighbor groups, respectively. We define a metric for honesty as follows:

$$honesty = (var_1 var_2)^{\frac{\alpha}{2}}(sum_2)^{-\frac{2}{\alpha}} \tag{1}$$

where α is defined as $log(\frac{m}{n_1})$, the ratio of the number of edges to the number of product nodes. The lower the honesty score, the more likely a node is to be a fraud. We describe each component in the honesty metric.

$\boldsymbol{var_1 var_2}$ has small values for frauds. Accessibility scores from honest users toward a fraudster are all small, resulting in small values of var_1. Accessibility scores among fraud colleagues are similar with each other due to dense interconnections, resulting in small var_2. In contrast, honest users has variable accessibility scores across the graph, resulting in large values of var_1 and var_2.

Isolated honest users who have few connections with the rest part of the graph have small accessibility scores for all nodes, resulting in small values of $var_1 var_2$. To deal with isolated users, we introduce the second term.

$\boldsymbol{sum_2}$ has large values with frauds. Dense interconnections in the fraud group result in high accessibility scores among them. In contrast, the isolated honest users have small-sized neighbor groups, resulting in small sum_2. sum_1 is not a good metric for honesty — both fraudsters and isolated honest users have small sum_1 with low accessibility scores for the stranger group.

Parameter $\boldsymbol{\alpha} = log(\frac{m}{n_1})$ regulates the effects of sum_2 and $var_1 var_2$ on the honesty estimation. The density of a graph $(\frac{m}{n_1 n_2})$ is a good indicator of the number of isolated users in the graph — when a graph has low density, it implies that there are many isolated users. With more isolated honest users, we need to put more priority on sum_2 than $var_1 var_2$.

Algorithm Algorithm 1 describes how we spot frauds based on the skewness in accessibility score distributions. We first compute an accessibility score matrix $\mathbf{A}_{acc}$ and the parameter α. Then we measure the honesty score based on Equation 1 in Algorithm 2. Finally, SkewA chooses top-k nodes with the lowest honesty scores as fraudsters.

5 Experiments

In this Section, we evaluate the performance of SkewA compared to state-of-the-art fraud detection methods. We aim to answer the following questions:

- **Q1. Robustness to sparse frauds:** Does SkewA outperform state-of-the-art competitors under various densities of frauds? (Section 5.2)
- **Q2. Camouflage-resistance:** How accurately does SkewA detect frauds under various types of camouflages? (Section 5.3)
- **Q3. Effects of camouflage ratio:** How does the camouflage ratio affect on the performance of SkewA? (Section 5.4)
- **Q4. Effectiveness of theoretical analysis:** Does our analysis on the accessibility score distributions coincide with the real-world datasets? (Section 5.5)

5.1 Setup

We implement SkewA in C++; all experiments are carried out on a 2.2 GHz Intel Core i7 Macbook Pro, 16GB RAM.

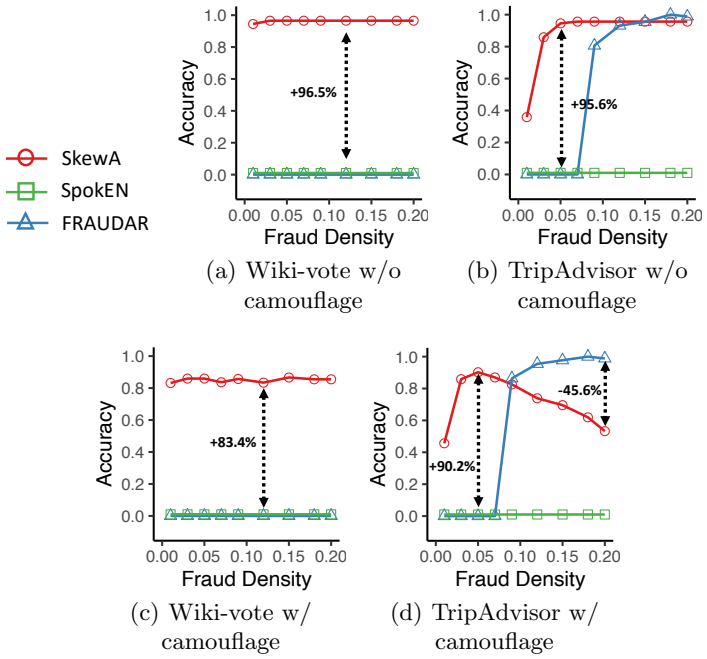


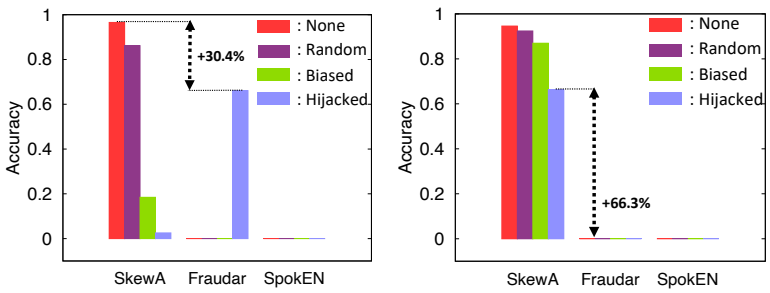
Fig. 6. Robustness to sparse and camouflaged frauds.

Dataset: we use two real-world datasets, Wiki-vote and TripAdvisor[2]. Wiki-vote is a who-trust-whom voting bipartite graph with $16K$ nodes ($8K$ for source and $8K$ for target) and $103K$ edges. TripAdvisor is a bipartite review graph with $147K$ nodes ($145K$ for users and $2K$ for products) and $176K$ edges. Parameter $\alpha = log(\frac{num.edges}{num.products})$ is set approximately with 1 and 2 on the Wiki-vote and TripAdvisor datasets, respectively.

Fraud injection: we inject a fraudulent block into each dataset. The numbers of fraudsters and fake products are 5% of total users and total items, respectively. The density of the block is set to 5%, and the corresponding number of edges are randomly generated among them. We inject four types of camouflage scenarios: 1) fraud with no camouflage, 2) random camouflage, 3) biased camouflage, and 4) hijacked accounts. In scenario 2), frauds write reviews on randomly chosen normal products. In scenario 3), frauds write reviews on normal products chosen with probability proportional to each product's degree. Finally, in scenario 4), frauds hijack honest accounts randomly and add reviews on fake products. The number of camouflage edges is decided by the camouflage ratio ρ_c (ratio of camouflage edges to fake edges). In our experiment, ρ_c is set to 0.1.

Baseline: we compare SkewA to state-of-the-art fraud detection methods, FRAUDAR [5] and SpokEN [11] described in Section 3.

[2] `http://snap.stanford.edu/data/`



(a) Wiki-vote (b) TripAdvisor

Fig. 7. Robustness to various camouflages of frauds.

5.2 Robustness to sparse frauds

We examine the robustness of SkewA varying density of frauds from 1% to 20%. We inject $1st$ and $2nd$ camouflage scenarios ('No Camo' and 'Random Camo') on the datasets. We compute honesty scores by each method, then choose bottom-k honest nodes where k is the number of injected frauds.

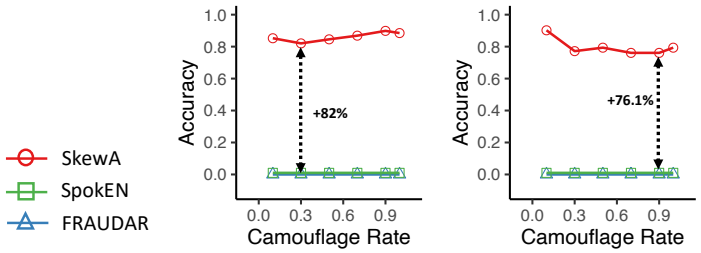
In Figure 6, SkewA shows consistently high accuracy under various densities of frauds on both datasets, while FRAUDAR and SpokEN barely detect frauds. FRAUDAR shows high accuracy only with high-density frauds on the TripAdvisor dataset. Since FRAUDAR focuses on dense subgraphs to detect fraud groups, sparse graphs (e.g., TripAdvisor) which make dense fraudulent subgraphs more noticeable are helpful for FRAUDAR. SpokEN relies on SVD to detect frauds, thus it is vulnerable to low-density and camouflages of frauds.

SkewA's accuracy decreases at a high density of frauds on the TripAdvisor dataset. TripAdvisor dataset has more isolated honest users with its low density. Then, high-density frauds result in higher var_2 (variance among colleagues) than var_2 of the isolated honest users. With lower var_2 than frauds, the isolated honest users has lower honest scores then become false positives. Overall, SkewA shows consistently high accuracy across all settings.

5.3 Camouflage-resistance

In this Section, we demonstrate the camouflage-resistance of SkewA. We change the camouflage scenarios: 1) 'No Camo', 2)'Random Camo', 3) 'Baised Camo', and 4) 'Hijacked'. Other settings are same as described in Section 5.1.

In Figure 7, SkewA is resistant to various types of camouflage attacks, while FRAUDAR and SpokEN miss most of the frauds. One exception is on the Wiki-vote dataset with the 'Hijacked' scenario where FRAUDAR shows high accuracy. On the Wiki-vote dataset, which has high density, frauds are likely to hijack honest users that are part of dense subgraphs. Then FRAUDAR, which focuses on dense subgraphs, is more likely to detect the frauds.



(a) Wiki-vote (b) TripAdvisor

Fig. 8. Robustness to camouflage ratios.

SkewA shows low accuracy in the 'Biased' and 'Hijacked' scenarios. In the 'Biased' scenario, a fraud makes connections to popular nodes which are connected with most honest nodes. Then any honest node connected to the popular nodes can reach the fraud groups easily through the popular nodes. Then accessibility scores from honest users to fraudsters increase, resulting in less skewed distributions. In the 'Hijacked' scenario, hijacked accounts are originally honest ones, thus already connected to other honest users. This brings high accessibility scores from honest users to fraudsters, resulting in less skewed accessibility score distributions. However, SkewA still spots some skewness in accessibility score distributions, showing higher accuracy than its competitors.

5.4 Effects of camouflage ratio

We discuss the effects of the camouflage ratio on the performance of SkewA. The camouflage ratio denotes the ratio between the number of camouflage edges and the number of fake edges. We vary the camouflage ratio from 0.1 to 1.0 under the same experimental setting described in Section 5.1. The camouflage type is set with the 'Random Camo' scenario. In Figure 8, as the camouflage ratio increases, SkewA shows consistently high accuracy, while FRAUDAR and SpokEN fail to detect frauds. SkewA exploits the unidirectionality of communication between frauds and honest users, thus not affected by the camouflage ratio.

5.5 Effectiveness of theoretical analysis

Theorem 2 describes the ratio of propagated scores into an honest group and a fraud group. Based on this theorem, we find out the skewness in accessibility score distributions of fraudsters. Here, we verify the effectiveness of Theorem 2 empirically on the TripAdvisor dataset. We compute the sum of scores propagated into each group based on Theorem 2 and compare with the experimental values. Under the same experimental setting described in Section 5.1, we notate the ratio of camouflage edges to fake edges as ρ_c and vary ρ_c from 0.1 to 1.0. Then the ratio of camouflage edges to honest edges ρ_a is decided by ρ_c and other parameters. The camouflage type is set with the 'Random Camo' scenario.

Table 3. $\frac{\sum_i s_1(i)}{\sum_i s_2(i)}$ on the TripAdvisor dataset: we compute ratio of sums of propagated scores into honest group S_1 and fraud group S_2 varying the seed node location.

	Theoretical Ratio		**Experimental Ratio**	
Seed Location	S_1	S_2	S_1	S_2
$\rho_c = 0$ $\rho_a = 0$	∞	0	∞	0
$\rho_c = 0.1$ $\rho_a =$ 2.2e−4	2288.4	0.25	177.1	0.36
$\rho_c = 0.3$ $\rho_a =$ 6.6e−4	1028	0.79	153.1	0.91
$\rho_c = 0.5$ $\rho_a =$ 1.1e−3	789	1.36	148.9	1.38
$\rho_c = 0.7$ $\rho_a =$ 1.6e−3	688.8	1.94	142.5	1.73
$\rho_c = 1$ $\rho_a =$ 2.2e−3	614.9	2.83	135.7	1.96

Score $s_1(k)$ and $s_2(k)$ denote scores propagated into the honest group S1 and the fraud group S2 at the k-th propagation step, respectively. We measure the sum of scores $\sum_i s_1(i)$ and $\sum_i s_2(i)$ propagated into each group and compute the ratio ($\frac{\sum_i s_1(i)}{\sum_i s_2(i)}$). In Table 3, the theoretical ratio and the experimental ratio show similar tendencies. When a seed node is chosen from normal product group S_1, fake product group S_2 receives only small amounts of scores, resulting in high ratios. This coincides with the skewed accessibility score distributions of fraudsters — low accessibility scores from normal users to fraudsters. On the other hand, when the seed node is chosen from S_2, S_1 receives moderate amounts of scores, leading to low ratios. This shows the weak skewness in the accessibility score distributions of normal users.

The differences between theoretical and experimental ratios come from dead-ends in real-world graphs. Scores could not be propagated further on dead-end nodes, and this leads to the score leak. Differences between theoretical ratios and experimental ratios are much smaller when a seed is located in S_2. The fraud group has fewer dead-ends than the honest group since they intentionally create accounts to make as many connections as possible for frauds. When scores are started from S_1, scores are more likely to meet dead-ends (then diminished) and it leads to a larger gap between theoretical and experimental values.

Similar tendencies in theoretical and empirical ratios prove our analysis on the accessibility score distributions is effective on the real-world datasets.

6 Conclusion

In this paper, we propose a novel algorithm SkewA for graph fraud detection. Due to the unidirectionality of communication between frauds and honest users, fraudsters show skewness in the accessibility score distributions . SkewA measures honesty based on this skewness. SkewA presents up to 95.6% accuracy in the public benchmarks where all competitors fail to detect any fraud. Future works include ensembling SkewA with density-focused fraud detection meth-

ods. The ensemble will make SkewA more robust to adversarial attacks with a high density of frauds.

References

1. Akoglu, L., Tong, H., Koutra, D.: Graph based anomaly detection and description: a survey. Data Mining and Knowledge Discovery **29**(3) (2015)
2. Beutel, A., Xu, W., Guruswami, V., Palow, C., Faloutsos, C.: Copycatch: stopping group attacks by spotting lockstep behavior in social networks. In: Proceedings of the 22nd international conference on World Wide Web (2013)
3. Cao, Q., Sirivianos, M., Yang, X., Pregueiro, T.: Aiding the detection of fake accounts in large scale social online services. In: Presented as part of the 9th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 12) (2012)
4. Dhawan, S., Gangireddy, S.C.R., Kumar, S., Chakraborty, T.: Spotting collective behaviour of online frauds in customer reviews. arXiv preprint arXiv:1905.13649 (2019)
5. Hooi, B., Song, H.A., Beutel, A., Shah, N., Shin, K., Faloutsos, C.: Fraudar: Bounding graph fraud in the face of camouflage. In: KDD (2016)
6. Jia, J., Wang, B., Gong, N.Z.: Random walk based fake account detection in online social networks. In: 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). IEEE (2017)
7. Jiang, M., Cui, P., Beutel, A., Faloutsos, C., Yang, S.: Catchsync: catching synchronized behavior in large directed graphs. In: KDD (2014)
8. Langville, A.N., Meyer, C.D.: Google's PageRank and beyond: The science of search engine rankings. Princeton university press (2011)
9. Liu, S., Hooi, B., Faloutsos, C.: Holoscope: Topology-and-spike aware fraud detection. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (2017)
10. Pan, J.Y., Yang, H.J., Faloutsos, C., Duygulu, P.: Automatic multimedia cross-modal correlation discovery. In: KDD (2004)
11. Prakash, B.A., Seshadri, M., Sridharan, A., Machiraju, S., Faloutsos, C.: Eigenspokes: Surprising patterns and community structure in large graphs (2010)
12. Shah, N., Beutel, A., Gallagher, B., Faloutsos, C.: Spotting suspicious link behavior with fbox: An adversarial perspective. In: ICDM (2014)
13. Tong, H., Lin, C.Y.: Non-negative residual matrix factorization with application to graph anomaly detection. In: SDM (2011)
14. Wang, B., Gong, N.Z., Fu, H.: Gang: Detecting fraudulent users in online social networks via guilt-by-association on directed graphs. In: 2017 IEEE International Conference on Data Mining (ICDM). IEEE (2017)
15. Wang, B., Zhang, L., Gong, N.Z.: Sybilscar: Sybil detection in online social networks via local rule based propagation. In: IEEE INFOCOM 2017-IEEE Conference on Computer Communications. IEEE (2017)
16. Yang, C., Harkreader, R., Zhang, J., Shin, S., Gu, G.: Analyzing spammers' social networks for fun and profit: a case study of cyber criminal ecosystem on twitter. In: Proceedings of the 21st international conference on World Wide Web (2012)
17. Yoon, M., Jung, J., Kang, U.: Tpa: Fast, scalable, and accurate method for approximate random walk with restart on billion scale graphs. In: ICDE (2018)