Knowledge and Information Systems (2025) 67:4459–4486
https://doi.org/10.1007/s10115-024-02327-9

REGULAR PAPER

Unleashing the power of indirect attacks against trust prediction via preferential path

Yu Bu[1] · Yulin Zhu[2] · Longling Geng[1] · Kai Zhou[1]

Received: 30 October 2023 / Revised: 25 July 2024 / Accepted: 26 December 2024 /
Published online: 6 February 2025

Abstract
Adversarial attacks in network security are a growing concern, prompting the need for innovative strategies to enhance both attack and defense mechanisms. This paper explores ways to improve adversarial attacks on the fairness and goodness algorithm (FGA) and review to reviewer (REV2), focusing on predicting trust within signed graphs. Unlike traditional time-based models, FGA and REV2 rely on iterative processes for trust propagation. By analyzing network structures, we identify *strong ties* and *weak ties* within FGA and discover *preferential paths* in REV2 that significantly impact information spread during algorithm iterations. Based on these insights, we propose a new approach called the *vicinage attack*, which enhances adversarial attacks by strategically targeting edges along these critical pathways. Our work highlights adversarial perturbation patterns that affect trust prediction on signed graphs and emphasizes their wide-reaching impact. These findings not only advance adversarial attack techniques but also deepen our understanding of trust propagation patterns. By clarifying the propagation bias in FGA and REV2, this research provides valuable insights for improving network security and developing better adversarial mitigation techniques in trust prediction.

Keywords Adversarial attack · Signed social network · Trust system · Network security · Discrete optimization

✉ Yu Bu
uuuyu.bu@connect.polyu.hk

Yulin Zhu
ylzhu@chuhai.edu.hk

Longling Geng
ll2024@outlook.com

Kai Zhou
kaizhou@comp.polyu.edu.hk

[1] Department of Computing, The Hong Kong Polytechnic University, 11 Yuk Choi Road, Hong Kong, HKSAR, China

[2] Department of Computer Science, Hong Kong Chu Hai College, 80 Castle Peak Road, Castle Peak Bay, Tuen Mun, N.T. HKSAR, China

1 Introduction

A signed graph is a widely used model to represent trust relationships among entities, where each edge is assigned a positive or negative sign to indicate the nature of interaction [1]. Positive edges typically denote positive interactions such as friendships or alliances, while negative edges signify conflicts or disagreements. This versatility finds applications across various domains, such as social networks [2, 3], sentiment analysis [4], and trust prediction [5].

For example, in a community where individuals engage in peer-to-peer transactions [6–8], each transaction involves an exchange of goods or services, and participants provide positive or negative feedback based on their experiences. A signed network can be constructed where nodes represent individuals, and edges between nodes carry signs to indicate whether the transaction feedback was positive or negative. In cryptocurrency, signed networks can provide insights into transaction dynamics between wallet addresses. Positive edges represent transactions from one wallet to another, while negative edges symbolize double-spending or fraudulent transactions. Analyzing such a signed network helps detect suspicious behaviors, track the flow of cryptocurrencies, and enhance security measures against fraud.

Understanding mutual trust information within signed graphs is crucial, and predicting trust relies on trust systems [9–11]. The fairness and goodness algorithm (FGA) is a popular trust system on signed directed networks for edge weight prediction [12]. FGA uses two metrics to characterize node behavior: goodness, which reflects how much other nodes trust a given node, and fairness, which assesses how impartially this node rates others. These concepts are defined in a mutually recursive manner that eventually converges to a unique solution. Kumar et al. demonstrated FGA's effectiveness in forecasting edge weights, indicating the trustworthiness between nodes that are not directly connected. Additionally, review to reviewer (REV2) is a network-based behavioral fraud detection algorithm [13].

REV2 has been adapted for risk rating on Ethereum [14, 15], where it quantifies the deanonymity score, life span suspiciousness, and wash suspiciousness metrics, each mapped to a standardized range of $[-1, 1]$. This transformation allows the representation of Ethereum transaction records as a weighted signed graph model. The REV2 algorithm can then assign trust scores to individual accounts and assess associated risks.

The reliability of trust prediction is critical due to the presence of adversarial attacks against trust prediction systems [16–18]. Attackers aim to manipulate trust scores using techniques like Sybil attacks [19–21], which involve tactics such as IP harvesting and deploying botnets [22]. Their primary goal is to manipulate target nodes' trust scores to evade detection by trust systems and cause confusion among users. This involves artificially inflating or deflating trust scores, effectively reversing the trustworthiness status of nodes. Studying adversarial attacks against trust prediction is essential. In the context of FGA on signed graphs, Lizurej et al. proposed an indirect Sybil attack on FGA. The objective is to decrease target nodes' trust by injecting edges into the original graph with a limited edge budget [17]. The problem involves a weighted signed network, attacker nodes S, target nodes T, an intermediary set I, a budget k, and a threshold value $r \in [-1, 1]$. Figure 1 provides details. The goal is to determine whether no more than k additional edges can decrease the trust score of each node in T below the threshold r. Attacker nodes can only affect nodes through the intermediary set I. This problem is NP-hard [17]. However, while Lizurej et al. showed indirect attacks are not effective, we find that indirect attacks can be effective due to the existence of *preferential paths*.

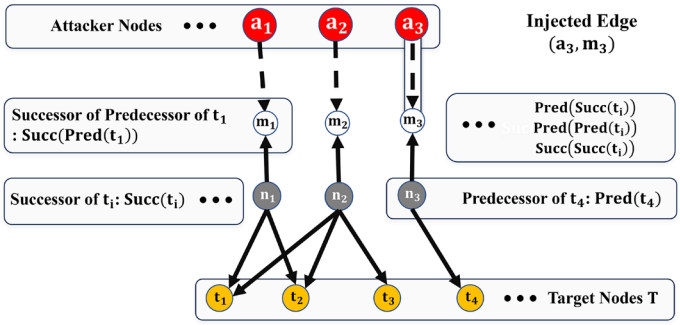


Fig. 1 The problem of Decrease Node Rating, an indirect Sybil attack on FGA, is NP-hard [17]. a_1, a_2, and a_3 are attacker nodes. m_1, m_2, and m_3 and n_1, n_2, and n_3 are intermediary nodes. t_1, t_2, t_3, and t_4 are target nodes

This paper explores adversarial attacks against trust prediction and introduces a new perspective on enhancing them. Our research focuses on the iterative propagation process within FGA and REV2. First, we aim to distinguish *strong ties* and *weak ties* in FGA from the intricate interplay of *tie structures*. Then, we extend these concepts as *preferential paths* within REV2. Our main contributions are:

- We confirm that *strong ties* and *preferential paths* are recognizable topology patterns more likely to be vulnerabilities for trust systems.
- We find a propagation bias within both FGA and REV2, where the trust metric is more likely to travel along *strong ties* or *preferential paths*.
- We are the first to highlight the role of *strong ties* and *preferential paths* in shaping the effectiveness of adversarial attacks.

Through this exploration, we aim to establish a foundation for a new paradigm in adversarial attack strategies—one that leverages the hidden influence of *strong ties* and *preferential paths* to enhance the potency of manipulative actions within complex networked environments. The remainder of this paper is organized as follows: Sect. 2 reviews existing research on trust prediction on signed graphs, status theory on signed graphs, trust and distrust propagation, and adversarial attacks on graphs. Section 3 describes the FGA and REV2 models used in our study. Section 4 outlines the specific problem statements and challenges addressed by our research. Section 5 analyzes network structures to identify strong and weak ties within FGA and preferential paths within REV2. Section 6 introduces our *vicinage*-attack approach. Section 7 presents experimental results, comparing the effectiveness of our method against baselines. Section 8 explores practical applications and broader implications. Section 9 discusses the limitations of our approach and outlines future research directions. Finally, the conclusion summarizes our key findings and their significance.

2 Related work

2.1 Trust prediction on signed graph

Hyperlink-Induced Topic Search (HITS) is a link analysis algorithm originally used to rank web pages. In signed graphs, HITS has been adapted for trust prediction by calculating

authority scores from the scaled values of hubs pointing to a node and hub scores from the scaled values of authorities linked from a node.

Several enhancements to the HITS algorithm have been made for various applications, including fraud detection, node ranking, and link prediction [23–27]. FGA builds on the HITS algorithm by introducing goodness and fairness metrics to improve trust prediction in signed graphs.

In addition to HITS-based modifications, PageRank-based algorithms have been adapted for trust prediction on signed graphs [28, 29]. REV2 has been further developed to enable more extensive propagation in complex networked environments. These advancements in HITS and PageRank-based algorithms have significantly improved the accuracy and reliability of trust prediction in signed graphs.

2.2 Status theory on signed graph

Status theory suggests that individuals within a social network occupy different levels of status, which influences their interactions with others [30]. High-status individuals are generally more trusted, while low-status individuals may face distrust. This is especially relevant in signed directed graphs, where edges represent trust (+) or distrust (-) and their direction indicates the flow of influence or information.

In signed directed graphs, high-status nodes tend to have more incoming trust edges and fewer distrust edges. Conversely, low-status nodes may have more incoming distrust edges. This pattern shows how people align their trust relationships with the perceived status of others.

Several studies have integrated status theory into trust prediction models [31, 32]. These studies show that considering social hierarchies can significantly improve the accuracy of trust predictions. In this paper, we use status theory to refine the perturbation spaces, reducing computational burden.

2.3 Trust and distrust propagation

Propagation-based techniques derive a dense matrix $\hat{A}$ from the original matrix A using specific propagation operators. These operators perform multiple stages of information diffusion, predicting link signs between nodes u_i and u_j as $sign(\hat{A_{ij}})$, with likelihood represented by $|\hat{A_{ij}}|$.

Guha et al. [33] conceptualize trust propagation as repetitive matrix operations with four atomic trust propagation types: direct propagation, trust coupling, co-citation, and transpose trust. Both Guha et al. and Lee et al. [34] use a matrix representation approach.

Other methods use alternative representations like subjective logic [35], intuitionistic fuzzy relations [36], and bi-lattice [37] to propagate both trust and distrust through defined operators.

In this paper, we refer to atomic trust and distrust propagation as *tie structures*, which we use to construct the perturbation space for adversarial attacks.

2.4 Adversarial attacks on graph

Adversarial attacks on graph structures aim to modify the adjacency matrix to cause significant classification errors [38, 39]. In a gray-box scenario, attackers use gradient-based

strategies on the adjacency matrix to identify critical modifications. Various methods exist for selecting edge alterations based on gradients [40–43].

These adversarial techniques challenge the robustness of graph-based anomaly detection systems by exploiting vulnerabilities in graph structures and classification models. Ongoing research aims to develop better defense mechanisms to mitigate these attacks.

In this paper, we focus on the perturbation spaces defined by *tie structures*, providing a new perspective on enhancing adversarial attacks.

3 Target model

The goal of the target model is to derive trust scores for each node based on graph-level information. To achieve this, the fairness and goodness algorithm (FGA) and review to reviewer (REV2) are proposed, using iterative definitions of fairness $f(u)$ and goodness $g(v)$. Fairness reflects how objectively node u rates others, while goodness indicates the level of trust placed in node v when rated by others. In REV2, reliability $r(u, v)$ is introduced as an edge-level metric during iterative learning, where reliable ratings are provided by fairer users, closer to goodness scores.

Consider a directed, weighted signed graph $G = (U, R, W)$, where user $u \in U$ generates a rating $(u, v) \in R$ for user $v \in U$. Here, U, R, and W represent the set of all users, ratings, and the scores of ratings, respectively. The rating score $W(u, v) \in [-1, 1]$ signifies the trust of node u in rating node v. Define:

- In(v) as the set of ratings received by node v
- Out(u) as the set of ratings given by node u
- |In(v)| as the count of ratings received by node v
- |Out(u)| as the count of ratings given by node u

This framework helps in systematically calculating trust scores and understanding the dynamics of trust within the network.

3.1 FGA as target model

The recursive formula of FGA is described as follows:

$$\begin{cases} g^{t+1}(u) = \dfrac{1}{|In(u)|} \displaystyle\sum_{u \in In(v)} f^t(v) \times W(u, v); \\ f^{t+1}(v) = 1 - \dfrac{1}{|Out(v)|} \displaystyle\sum_{u \in Out(v)} \dfrac{|W(u, v) - g^{t+1}(u)|}{2} \end{cases} \tag{1}$$

Here, $f(u)$ ranges in [0,1] while $g(u)$ falls in $[-1, 1]$. Both $f(u)$ and $g(u)$ are initialized as 1 and are recursively updated over all nodes until they converge to a small value ε. The convergence condition is $\left|g^{t+1}(u) - g^t(u)\right| < \varepsilon$ or $\left|f^{t+1}(u) - f^t(u)\right| < \varepsilon$. Lizurej et al. consider the goodness metric as a manipulative trust score and have proved that an indirect attack targeting goodness on FGA is NP-hard [17].

3.2 REV2 as target model

The recursive formula of REV2 is expressed as follows:

$$\begin{cases} g^{t+1}(v) = \dfrac{\sum\limits_{(u,v)\in In(v)} r^{t+1}(u,v)\cdot W(u,v) + \beta_1 \cdot \mu_g}{|In(v)| + \beta_1}; \\ r^{t+1}(u,v) = \dfrac{\gamma_1 \cdot f^{t+1}(u) + \gamma_2 \left(1 - \frac{|W(u,v) - g^t(v)|}{2}\right)}{\gamma_1 + \gamma_2}; \\ f^{t+1}(u) = \dfrac{\sum\limits_{(u,v)\in Out(u)} r^t(u,v) + \alpha_1 \cdot \mu_f}{|Out(u)| + \alpha_1} \end{cases} \tag{2}$$

In this context, $f(u)$, $g(v)$, and $r(u, v)$ values fall within [0, 1]. μ_f and μ_g are set as the mean scores of all users' fairness and goodness scores, respectively. $f(u)$, $g(v)$, and $r(u, v)$ are initialized as 1 and are recursively updated until convergence is achieved. The convergence condition is $\max\{|g^{t+1}(u) - g^t(u)|, |f^{t+1}(u) - f^t(u)|, |r^{t+1}(u, v) - r^t(u, v)|\} < \varepsilon$, where ε is a small value. According to [13], the rank of the fairness score can be used to detect fraudulent users. For this process, α_1, γ_1, γ_2, and β_1 are set to 1.

4 Problem statements

4.1 Threat model

This study presents a detailed threat model for adversarial attacks, specifically targeting trust prediction within signed social networks and trust systems like FGA and REV2. Our threat model includes the following key aspects:

Attacker's Objective: Malicious entities, referred to as attackers, aim to disrupt the trust prediction process. They seek to manipulate trust scores for various purposes, such as discrediting specific users, hiding their true intentions, or boosting their trust scores within the network. Attackers achieve these goals by carefully crafting adversarial perturbations.

Knowledge and Capabilities: Attackers know the trust prediction algorithm and the network structure. They can manipulate the network by adding or modifying edges and influencing trust scores. Attackers use techniques like edge addition, deletion, or weight modification, strategically targeting influential pathways in trust propagation. Our focus is on edge injection, as removing edges is technically and practically challenging. For an injected edge (i, j) from i to j, we refer to it as a **positive attack** from i or a **passive attack** toward j.

Attacker's Stealthiness: In indirect attacks, stealthiness is crucial. Unlike direct attacks, indirect attacks blend with genuine trust propagation to avoid detection. Attackers use tactics to ensure their actions do not raise suspicion within the network. Indirect attacks involve the participation of target nodes, their neighbors, and attacker nodes, making stealth a priority.

In summary, our threat model examines how attackers exploit *preferential paths* and *strong ties* within signed graphs, using their knowledge of algorithms, network structures, and perturbation techniques. It emphasizes the importance of stealth in indirect attacks to achieve their goals.

4.2 Problem formulation

In this section, we formulate the indirect attack against trust prediction as a bi-level discrete optimization problem.

We represent the original graph as $G_0 = \{U_0, R_0, W_0, Y_0\}$, where Y_0 is the trust score or trust rank of the nodes. In adversarial attacks, attackers inject malicious edges, resulting in a manipulated graph $G_\varepsilon = \{U_0, R, W, Y_\varepsilon\}$, which is observed by trust system administrators. The trust prediction function is denoted as $p(\cdot)$.

The graph structure is modified by introducing additional edges with weight scores to manipulate trust prediction results. The attacker's primary goal is to minimize or maximize the trust prediction results of all target nodes in T by introducing at most k additional edges. The influence of attacker nodes S should be mediated through intermediary nodes in set $I(T)$. The perturbation space is constructed by connecting attacker nodes with intermediary nodes, denoted as $\Phi_{S,T}(B) = \{S \times I(T)\}_{w\in\{-1,1\}}$. The binary variable B represents the set of newly introduced edges.

The top-k largest entries in B^* are selected for modification through the operation $\sigma^k_\Phi(\cdot)$, denoted as $\bigcup_1^k (i^*, j^*) = \sigma^k_\Phi(B^*) = \arg\max_{k\,(i,j,w)\in\Phi_{S,T}(B^*)} B^*$. The indirect attack against the trust system can then be formulated as a bi-level optimization problem:

$$\bigcup_1^k (i^*, j^*) = \sigma^k_\Phi(B^*) = \underset{\substack{\arg\min_{(G_\varepsilon - G)\in\Phi_{S,T}(B^*);\\ (i,j,w)\in\Phi_{S,T}(B^*)}}}{\arg\max_k} \left(\sum_i^{|T|} (p^{t_i}(G_\varepsilon) - p^{t_i}(G_0)) + \sum_{i=0}^{|\Phi_{S,T}(B^*)|} b_i^* \cdot c_i \right)$$
$$\text{s.t.} \sum_{i=0}^{|\Phi_{S,T}(B)|} b_i = k,\ \Phi_{S,T}(B) := \bigcup_i \Phi_{S,T}(B_i),\ B \in \{0,1\}^{|\Phi_{S,T}(B)|} \tag{3}$$

Here, c_i represents the cost associated with the injected edges. The perturbation space includes various categories of edges for injection. Specifically, in a directed graph, there are two types of 1-hop neighbors for target node T: $Pred(T)$ and $Succ(T)$. Similarly, there are four types of 2-hop neighbors: $Succ(Pred(T))$, $Pred(Succ(T))$, $Succ(Succ(T))$, and $Pred(Pred(T))$. $Pred(\cdot)$ represents the predecessors of nodes, and $Succ(\cdot)$ represents the successors of nodes. $\Phi_{S,T}(B_i)$ denotes the different types of perturbation spaces $\{S \times I_{type_i}(T)\}_{w\in\{-1,1\}}$. Each entry $b_i^* \in B^*$ is a binary variable, taking values from the set $\{0, 1\}$ to indicate the inclusion or exclusion of a specific edge.

5 Analysis of preferential path

To thoroughly understand trust propagation in the trust system, we focus on analyzing a specific pathway. This pathway includes the flow from the attacker node to the target node, covering the target node's neighbor and the injected edge. This structural configuration is referred to as the *tie structure*.

In this section, we distinguish *strong ties* and *weak ties* on FGA from the intricate interplay of *tie structures*. Then, we extend these concepts to *preferential paths* within REV2. The *strong ties* on FGA can be considered a specific example of *preferential paths* because attackers prefer to attack via *strong ties* rather than *weak ties* (to be detailed later).

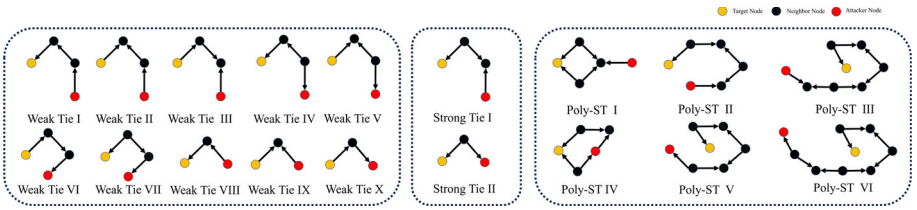


Fig. 2 Concept of weak tie, strong tie, and Poly-ST on FGA. The examples of weak tie and strong tie are classical. The examples on Poly-ST are non-classical because there are more examples that can be enumerated

5.1 Strong ties and weak ties on FGA

Strong ties and *weak ties* describe whether a *tie structure* allows an attacker to indirectly generate a perturbation on *goodness*. The *goodness* metric reflects how much trust is received by other nodes. We also use conductivity and resistance to describe the capability of generating perturbations on target nodes concerning FGA *goodness*. Below, we enumerate all *tie structures* involving either 1-hop or 2-hop neighbors of the target node. Then, we distinguish *strong ties* and *weak ties* among them.

Definition 1 (TIE STRUCTURE) For the graph structure consisting of a target node t, an attacker node, and intermediary node(s), we call them Tie Structures.

Definition 2 (WEAK TIE I to X) For the tie structure consisting of a target node t, an attacker node, and 1) $Succ(t)$ (referred to as Weak Tie IX and Weak Tie X in Fig. 2); 2) $Pred(t)$ and $Pred(Pred(t))$ (referred to as Weak Tie I and Weak Tie IV in Fig. 2); 3) $Succ(t)$ and $Succ(Succ(t))$ (referred to as Weak Tie III and Weak Tie VI in Fig. 2); 4) $Succ(t)$ and $Pred(Succ(t))$ (referred to as Weak Tie II and Weak Tie V in Fig. 2); 5) attacker positive attack toward $Pred(t)$ (referred to as Weak Tie VIII in Fig. 2); and 6) attacker passive attack toward $Succ(Pred(t))$ (referred to as Weak Tie VII in Fig. 2), we call them Weak Ties.

Definition 3 (STRONG TIE I and II) For the tie structure consisting of a target node t, an attacker node, and 1) attacker positive attack toward $Succ(Pred(t))$ (referred to as Strong Tie I in Fig. 2) and 2) attacker passive attack toward $Pred(t)$ (referred to as Strong Tie II in Fig. 2), we call them Strong Ties.

Definition 4 (POLYMORPHIC STRONG TIE) For the tie structure consisting of a target node t, an attacker node, and 1) if there simultaneously exist multiple Strong Tie I (referred to as Poly-ST I in Fig. 2); 2) if there simultaneously exist multiple Strong Tie II; and 3) if there simultaneously exist Strong Tie I and Strong Tie II (referred to as Poly-ST IV in Fig. 2), we call them Polymorphic Strong Ties.

The concept of polymorphic strong tie (Poly-ST) can be extended to more than 2-hop neighbors (referred to as Poly-ST II, Poly-ST III, Poly-ST V, and Poly-ST VI in Fig. 2). Table 1 summarizes all indirect attack cases within at most 2-hop neighbors, filtering out two cases as strong tie candidates according to rigging properties (to be detailed later). We will also quantitatively analyze strong ties and weak ties. More details will be presented and discussed in the experimental evaluation.

Table 1 Indirect attack tie structures on FGA

Categories	Positive attack	Passive attack
Pred(T)	✗	✓
Succ(T)	✗	✗
Succ(Pred(T))	✓	✗
Pred(Succ(T))	✗	✗
Succ(Succ(T))	✗	✗
Pred(Pred(T))	✗	✗

✓ is the case capable to generate perturbation on target node's goodness, which is more likely to be *strong tie*. ✗ are the cases cannot influence target node's goodness. The positive and passive attack is in the view of attackers

5.1.1 Ties rigging axioms

The mutual recursive learning process on FGA leads to a noticeable propagation bias. This bias, seen as trust propagation on *tie structures*, can be understood through properties of conductivity and resistance. Trust propagation tends to follow a *preferential path*. For more detailed proofs, refer to [12, 17].

Axiom 1 (DIRECTION-ORIENTED RIGGING). *A perturbation on $f(Pred(t))$ will change $g(t)$, and similarly, a perturbation on $g(Succ(t))$ will change $f(t)$.*

The incoming edge of t starts from $Pred(t)$, and the outgoing edge of t ends with $Succ(t)$. From Eq. (1), a perturbation on $f(Pred(t))$ changes $g(t)$, and a perturbation on $g(t)$ changes $f(Pred(t))$. Similarly, a perturbation on $g(Succ(t))$ changes $f(t)$, and a perturbation on $f(t)$ changes $g(Succ(t))$. Axiom 1 demonstrates the interdependence of metrics in mutual recursive learning. In [12], the goodness axiom and fairness axiom also highlight this dependence.

Axiom 2 (SMOOTH GOODNESS). *An increase in $f(Pred(t))$ results in a proportional increase in $g(t)$.*

This is a simplified explanation of the citation axiom in [17] (Axiom 1).

Axiom 3 (OBVIOUS FAIRNESS METRIC). *If node $Pred(t)$ rates all its successor nodes with a rating bias $\gamma = |g(u) - W(Pred(t), u)| = 0$, then $f(Pred(t)) = 1$. If node $Pred(t)$ rates all its successor nodes with a rating bias $\gamma = |g(u) - W(Pred(t), u)| = 2$, then $f(Pred(t)) = 0$.*

This is a simplified explanation of the citation axiom in [17] (Axiom 9). This axiom indicates that the rating bias determines the fairness score of nodes.

5.1.2 Tie strength analysis

Trust propagation shows conductivity on *strong ties* and resistance on *weak ties*. Here, we explore the reasons behind this propagation bias using ideal topological models to analyze trust propagation on FGA quantitatively. We show that the effectiveness of *strong ties* depends on both the sign and weight of the injected edge.

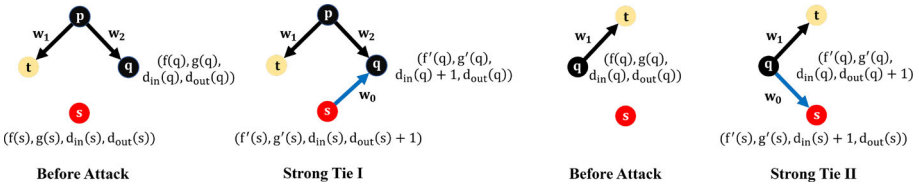


Fig. 3 Tie strength analysis on strong ties

To simplify the discussion, we assume the edge weights w_1 and w_2 in Fig. 3 are positive, as negative edges are relatively rare in signed graphs. For Strong Tie I, this involves reducing $g(q)$, which then decreases $f(p)$ and, consequently, $g(t)$. For Strong Tie II, the rigging mechanism focuses on reducing $f(q)$, which then lowers $g(t)$.

Theorem 1 (RESISTANCE OF WEAK TIES) *From Weak Tie I to Weak Tie X, $g(t)$ remains unchanged.*

Proof If there is a perturbation on $f(Pred(t))$, then $g(t)$ will change. This rigging property can be expressed as $\Delta g(t) \Leftrightarrow \Delta f(Pred(t))$. Similarly, if there is a perturbation on $g(Succ(t))$, $f(t)$ will change, expressed as $\Delta f(t) \Leftrightarrow \Delta g(Succ(t))$. Only two ties can generate a perturbation on $g(t)$: $\Delta g(t) \Leftrightarrow \Delta f(Pred(t)) \Leftrightarrow \Delta g(Succ(Pred(t)))$ and $\Delta g(t) \Leftrightarrow \Delta f(Pred(t))$. These correspond to Strong Tie I and Strong Tie II. Given the topological model of *strong ties* and *weak ties*, there is only one path from the attacker node to the target node. Therefore, *weak ties* do not influence $g(t)$. □

Theorem 2 (CONDUCTIVITY OF STRONG TIE I) *For Strong Tie I, if (s, q) is injected with weight w_0 and $f(s)w_0 < g(q) < w_2$, then $\Delta g(t) < 0$; or if $g(q) > \max(w_2, f(s)w_0)$ and $w_0 < \frac{2w_2-1}{f(s)}$, then $\Delta g(t) < 0$.*

Proof As shown in Fig. 3, if there is an injected edge attack, then $g'(q) = \frac{g(q)d_{in}(q)+f(s)w_0}{d_{in}(q)+1}$. Thus, $\Delta g(q) = g'(q) - g(q) = \frac{f(s)w_0-g(q)}{d_{in}(q)+1}$. If $g(q) > f(s)w_0$, $g(q)$ decreases; otherwise, $g(q)$ increases. The perturbation of $\Delta f(p)$ is $\Delta f(p) = \frac{1}{2d_{out}(p)}(|w_2 - g(q)| - |w_2 - g'(q)|)$. There are four cases:

$$w_2 > g(q), w_2 > g'(q), \Delta f(p) = \frac{\Delta g(q)}{2d_{out}(p)}; \tag{4a}$$

$$w_2 < g(q), w_2 > g'(q), \Delta f(p) = \frac{2g(q) - 2w_2 + \Delta g(q)}{2d_{out}(p)}; \tag{4b}$$

$$w_2 < g(q), w_2 < g'(q), \Delta f(p) = -\frac{\Delta g(q)}{2d_{out}(p)}; \tag{4c}$$

$$w_2 > g(q), w_2 < g'(q), \Delta f(p) = \frac{2w_2 - 2g(q) - \Delta g(q)}{2d_{out}(p)} \tag{4d}$$

We expect $\Delta g(q) < 0$ and $\Delta f(p) < 0$, so that $\Delta g(t) = \frac{\Delta f(p)w_1}{d_{in}(t)} < 0$ when $w_1 > 0$. In (4c), if $\Delta g(q) < 0$, then $\Delta f(p) > 0$; in (4d), if $g(q) < w_2 < g'(q)$, then $\Delta g(q) > 0$. Thus, only two cases need to be discussed.

CASE I: If $f(s)w_0 < g(q) < w_2$, then $\Delta g(q) < 0$, $\Delta f(p) < 0$, and $\Delta g(t) < 0$.

Table 2 Status theory in trust propagation on REV2

Categories	Positive edge	Negative edge
(low fairness, high fairness)	✓	✗
(high fairness, low fairness)	✗	✓

✓ is the case satisfies status theory while ✗ is the case disobey status theory. In original datasets, for negative edges defy status theory, there are 3.725% in *bitcoin-alpha* and 5.727% in *bitcoin-otc*. For positive edges defy status theory, there are 47.168% in *bitcoin-alpha* and 46.084% in *bitcoin-otc*

CASE II: We have $\Delta g(q) = \frac{f(s)w_0 - g(q)}{d_{in}(q)+1} < 2w_2 - 2\,g(q)$, $-1 < g(q) < \frac{2(d_{in}(q)+1)w_2 - f(s)w_0}{2d_{in}(q)+1} < 1$, $\frac{2w_2 - f(s)w_0 - 1}{2(1-w_2)} < d_{in}(q) < \frac{2w_2 - f(s)w_0 - g(q)}{2(g(q)-w_2)}$. So, if $g(q) > \max(w_2, f(s)w_0)$ and $w_0 < \frac{2w_2-1}{f(s)}$, then $\Delta g(q) < 0$, $\Delta f(p) < 0$, and $\Delta g(t) < 0$. □

Theorem 3 (CONDUCTIVITY OF STRONG TIE II) *For Strong Tie II, if* (q, s) *is injected with weight* w_0 *and* $f(p)w_0 < g(s) < w_0$, $\Delta g(t) < 0$*; or if* $g(q) > \max(w_0, f(s)w_0)$ *and* $w_0 > \frac{1}{2-f(s)}$, *then* $\Delta g(t) < 0$.

Proof As shown in Fig. 3, if there is an injected edge attack, then $\Delta g(s) = g'(s) - g(s) = \frac{f(q)w_0 - g(s)}{d_{in}(s)+1}$ and $\Delta f(q) = \frac{1}{2d_{out}(q)}(|w_0 - g(s)| - |w_0 - g'(s)|)$. We expect $\Delta f(q) < 0$, so that $\Delta g(t) = \frac{\Delta f(q)w_1}{d_{in}(t)} < 0$ when $w_1 > 0$.

CASE I: If $f(p)w_0 < g(s) < w_0$, then $\Delta g(s) < 0$, $\Delta f(q) < 0$, and $\Delta g(t) < 0$.

CASE II: We have $\Delta g(s) = \frac{f(q)w_0 - g(s)}{d_{in}(s)+1} < 2w_0 - 2\,g(q)$, $-1 < g(s) < \frac{2(d_{in}(s)+1)w_0 - f(s)w_0)}{2d_{in}(s)+1} < 1$, $\frac{2w_0 - f(s)w_0 - 1}{2(1-w_0)} < d_{in}(s) < \frac{2w_0 - f(s)w_0 - g(q)}{2(g(q)-w_0)}$. So, if $g(q) > \max(w_0, f(s)w_0)$ and $w_0 > \frac{1}{2-f(s)}$, then $\Delta g(s) < 0$, $\Delta f(q) < 0$, and $\Delta g(t) < 0$. □

5.2 Preferential path on REV2

We aim to identify which nodes are crucial in forming *preferential paths* on REV2. To assist in this analysis, we introduce *status theory* as outlined in Table 2. In a signed directed graph, a positive edge from A to B means A believes B's status is higher than A's, while a negative edge from C to D means C believes D's status is lower than C's. We use fairness score to represent status. *Status theory* helps narrow the scope of perturbation spaces, allowing us to focus on situations where drastic changes might occur in the trust system. Our main focus is on injected edges that violate the principles of status theory, as these are more likely to disrupt normal trust prediction. Violations include a positive edge from a high-fairness node to a low-fairness node and a negative edge from a low-fairness node to a high-fairness node.

Table 3 provides a comprehensive list of all possible preferential paths identified in our experiments. Unlike the theoretical definitions of *strong ties* and *weak ties* on FGA, the concept of *preferential path* in REV2 is defined empirically (see experiment results on REV2). The main difference between *strong ties* and *preferential paths* is that the latter are the preferred pathways among all conductive paths. To clarify, if a *tie structure* is a *preferential path*, it must also meet the criteria of a *strong tie*. However, a *strong tie* may not always be a *preferential path*.

Table 3 Indirect attack cases tie structures on REV2

Categories	Passive attack (+1)	Positive attack (−1)
Pred(T)	♥	♡
Succ(T)	♥	♡
Succ(Pred(T))	♡	♥
Pred(Succ(T))	♥	♡
Succ(Succ(T))	♡	♥
Pred(Pred(T))	♥	♡

♥ means the case is more likely to be preferential path. ♡ means the case cannot be a preferential path because of empirical test. The positive and passive attack is in the view of target nodes

6 Proposed method

The overall architecture of the proposed method is shown in Fig. 4. The processes of *sensitivity analysis* and *universe analysis* help distinguish *preferential paths* from other *tie structures* in FGA and REV2. It is important to note that the construction of the perturbation space allows for the reutilization of results from adversarial training, which may lead to adversarial retraining if necessary.

6.1 Perturbation space construction

The perturbation space is constructed by connecting attacker nodes to the neighbors of target nodes (intermediary nodes), denoted as $\Phi_{S,T}(B) = \{S \times I(T)\}_{w\in\{-1,1\}}$. This pathway, consisting of an attacker node, injected edge, target node's neighbor, and the target node, forms a new *tie structure*. Additionally, the injected edge should be weighted. We refer to these two processes as *tie structure generation* and *edge weight generation*.

The perturbation space $\Phi_{S,T}(B_0)$ includes edges originating from attacker nodes and pointing to $Succ(Pred(T))$. Similarly, the perturbation spaces $\Phi_{S,T}(B_1)$, $\Phi_{S,T}(B_2)$, and $\Phi_{S,T}(B_3)$ include edges connecting attacker nodes to $Pred(Succ(T))$, $Succ(Succ(T))$, and $Pred(Pred(T))$, respectively. Recognizing that there are shared nodes among the neighborhoods of these four types of 2-hop neighbors, we combine the entire perturbation space under the term $\Phi_{S,T}(B)$. This is defined as $\Phi_{S,T}(B) := \bigcup_i \Phi_{S,T}(B_i)$, representing a comprehensive universe of perturbations.

For simplification, we assume that each edge has a uniform cost of 1. The optimization problem can be simplified and expressed as the following formula:

$$\bigcup_{1}^{k}(i^*, j^*) = \sigma_{\Phi}^{k}(B^*) = arg\max_{\substack{k \\ (G_\varepsilon - G)\in\Phi_{S,T}(B^*); \\ (i,j,w)\in\Phi_{S,T}(B^*)}} arg\min\left(\sum_{i}^{|T|}(p^{t_i}(G_\varepsilon))\right)$$
$$\text{s.t. } \Phi_{S,T}(B) := \bigcup_i \Phi_{S,T}(B_i),\ B \in \{0,1\}^{|\Phi_{S,T}(B)|} \tag{5}$$

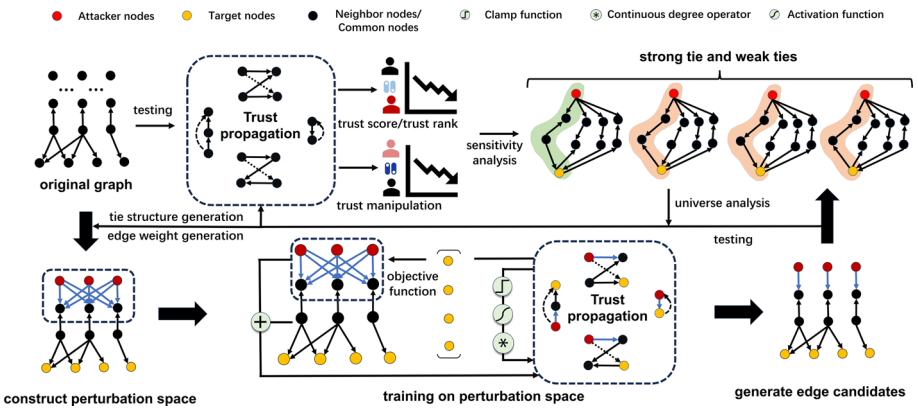


Fig. 4 Architecture of proposed method on FGA and REV2

6.2 Continuous graph and degree operators

The optimization task is challenging due to the inherently discrete nature of the binary variable B. To address this challenge, we present a novel approach involving the redefinition of B as $\widetilde{B}$. Each element $\widetilde{b}_i \in B^*$ denotes the probability of manipulation pertaining to the $edge_i$. This redefinition effectively transforms the underlying graph structure from a discrete graph to a continuous one, creating a probability space derived from the original perturbation space. The corresponding perturbation space is $\Phi_{S,T}(\widetilde{B}) = \{S \times I(T)\}_{w\in\{-1,1\}}^{p=0.5}$, where $p = 0.5$ is the initialized probability. This continuous representation of the graph allows us to conduct optimization operations more efficiently. The continuous graph can be expressed as $\widetilde{\mathcal{G}} = \Phi_{S,T}(\widetilde{B}) \cup G_0$. The optimization problem can be rewritten as follows:

$$\begin{aligned} &\bigcup_{1}^{k}(i^*, j^*) = \sigma_{\Phi}^{k}(\widetilde{B}^*) = \underset{\substack{k \\ (G_\varepsilon - G)\in\Phi_{S,T}(\widetilde{B}^*); \\ (i,j,w)\in\Phi_{S,T}(\widetilde{B}^*)}}{arg\max}\ arg\min\left(\sum_{i}^{|T|}(p^{t_i}(G_\varepsilon))\right) \\ &\text{s.t. } \Phi_{S,T}(\widetilde{B}) := \bigcup_{i}\Phi_{S,T}(\widetilde{B}_i),\ \widetilde{B} \in \{0,1\}^{|\Phi_{S,T}(\widetilde{B})|} \end{aligned} \tag{6}$$

The introduction of the continuous graph necessitates the formulation of new indegree and outdegree operators because the degree operators on the continuous graph should be differentiable with respect to the variable $\widetilde{B}$. We define degree operators using the hyperbolic tangent function, with a smoothness parameter β. The newly defined indegree and outdegree operators can be expressed as follows:

$$\begin{aligned} d_{in}(u_{i_0}) &= \sum_{j=i_0,(i,j)\in G} e_{ij} + \sum_{j=i_0,(i,j,w,p)\in\Phi_{S,T}(\widetilde{B})} \frac{\tanh(\beta p)+1}{2} \\ d_{out}(u_{i_0}) &= \sum_{i=i_0,(i,j)\in G} e_{ij} + \sum_{i=i_0,(i,j,w,p)\in\Phi_{S,T}(\widetilde{B})} \frac{\tanh(\beta p)+1}{2} \end{aligned} \tag{7}$$

6.3 Optimizing non-smooth trust rank

In REV2, trust scores are arranged in ascending order to produce a deterministic trust rank, $\mathcal{R}(p(G_0))$, where the top-ranked nodes are considered unfair or anomalous. Our goal is to avoid sorting and use gradient-based optimization techniques instead. Inspired by *Softrank* by Taylor et al. [44], we propose an approximate algorithm to generate rank distributions without explicit sorting. For a given node, $node_i$, we aim to estimate the probability that $node_i$ ranks higher than another node, $node_j$. This probability, denoted as π_{ij}, is given by:

$$\pi_{ij} \equiv Pr(s_i - s_j > 0) = \frac{1}{\beta} \int_{-\infty}^{s_i - s_j} 1 - \tanh^2(\beta s)\, ds$$

This probability reflects how often $node_i$ will achieve a higher trust rank than $node_j$ in repeated pairwise comparisons. By aggregating these probabilities for a node being ranked higher than every other node, the expected trust rank of $node_i$ is:

$$E[r_i] = \sum_{i \neq j, j=1}^{|U|} \pi_{ij}$$

We introduce an attacker's deterministic function, $\mathcal{F}_{atk}(\cdot)$, to model the impact of manipulation on trust rank. This function determines whether the manipulation aims to minimize or maximize the trust rank. Specifically, $\mathcal{F}_{atk}(\cdot)$ is defined as 1 for minimizing the trust rank and -1 for maximizing it. In REV2, the optimization problem is formulated as follows:

$$\begin{aligned} &\bigcup_{1}^{k} (i^*, j^*) = \sigma_{\Phi}^{k}(\widetilde{B}^*) = arg \max_{k} \; arg \min \mathcal{F}_{atk}\left(\sum_{i}^{|T|} \mathcal{R}^{t_i}(p(G_\varepsilon)) \right) \\ &\qquad\qquad (G_\varepsilon - G) \in \Phi_{S,T}(\widetilde{B}^*); \\ &\qquad\qquad (i, j, w) \in \Phi_{S,T}(\widetilde{B}^*) \\ &\text{s.t. } \Phi_{S,T}(\widetilde{B}) := \bigcup_{i} \Phi_{S,T}(\widetilde{B}_i),\ \widetilde{B} \in \{0, 1\}^{|\Phi_{S,T}(\widetilde{B})|} \end{aligned} \tag{8}$$

6.4 Projection and distribution of gradient

We use gradient descent for the optimization problem, as detailed in previous works [39–43]. We start with $\widetilde{B}$ initialized at 0.5 and update it using the rule: $\widetilde{B} \leftarrow \Pi_{[0,1]}(\widetilde{B} - \alpha \nabla_{\widetilde{B}} \mathcal{F}_{atk}(\mathcal{C}_{\widetilde{B}}))$, where α is the learning rate. Here, $\mathcal{C}_{\widetilde{B}}$ represents the cumulative trust rank of target nodes from $\Phi_{S,T}(\widetilde{B}) \cup G_0$. The gradient projection ensures that $\widetilde{B}$ remains within the range [0, 1]. This projection is also applied during updates to fairness, goodness, and reliability. After several optimization steps, $\widetilde{B}$ converges to values close to either 1 or 0, indicating a higher probability of edge injection. Finally, the gradient distribution (in descending order) is used to select candidate edges.

As a summary, the proposed *vicinage*-attack method is outlined in Algorithm 1. The algorithm includes an outer loop for optimization steps and an inner loop for iterative learning. Parameters n and l denote the number of iterations for the outer and inner loops, respectively. The core idea of our adversarial attack algorithm is to treat the constructed perturbation space as a hyperparameter and compute the gradient of the attacker's objective with respect to it.

Algorithm 1 Gradient-based Indirect Attack

Input: constructed perturbation space $\Phi_{S,T}(\widetilde{B})$, original graph $G_0 = (U_0, R_0, W_0)$, budget k, iteration n, l, target node T, learning rate α

Function: trust system $REV2$; newly defined degree operators $In(\cdot)$ and $Out(\cdot)$, attacker's deterministic function $\mathcal{F}_{atk}(\cdot)$, smooth trust rank function $\mathcal{R}_{\widetilde{B}}(\cdot)$

Output: Candidate edges $G_\varepsilon - G_0$

function VICINAGE- ATTACK($\Phi_{S,T}(\widetilde{B})$, G_0, k, n, l, T)
  $\widetilde{\mathbf{B}} = 0.5$
  for $i = 1$ to n **do**
    for $j = 1$ to l **do**
      for each node u in U_0 **do**
        Update $g(u)$ via $In(\Phi_{S,T}(\widetilde{B}) \cup G_0)$
        $g(u) = \Pi_{[0,1]} g(u)$
      end for
      for each edge (u, v) in $\Phi_{S,T}(\widetilde{B}) \cup G_0$ **do**
        Update $r(u, v)$
        $r(u, v) = \Pi_{[0,1]} r(u, v)$
      end for
      for each node u in U_0 **do**
        Update $f(u)$ via $Out(\Phi_{S,T}(\widetilde{B}) \cup G_0)$
        $f(u) = \Pi_{[0,1]} f(u)$
      end for
    end for
    for each node t in T **do**
      $\mathcal{C} \leftarrow \mathcal{C} + \mathcal{R}^t_{\widetilde{B}}(\Phi_{S,T}(\widetilde{B}) \cup G_0)$
    end for
    for each edge $\widetilde{\mathbf{b}}_\mathbf{i}$ in $\widetilde{\mathbf{B}}$ **do**
      $\widetilde{\mathbf{b}}_\mathbf{i} \leftarrow \Pi_{[0,1]}(\widetilde{\mathbf{b}}_\mathbf{i} - \alpha \nabla_{\widetilde{b}_i} \mathcal{F}_{atk}(\mathcal{C}_{\widetilde{B}}))$
    end for
  end for
  Select Candidate Edges: $\bigcup_1^k (i^*, j^*) = \sigma^k_\Phi(B^*)$
end function

7 Evaluation

7.1 Dataset description

We evaluated our proposed attacks using two real-world signed networks from the Stanford Network Analysis Project (SNAP: http://snap.stanford.edu/index.html). The datasets used are:

- *bitcoin-alpha*: This dataset includes 3,783 nodes and 24,186 edges.
- *bitcoin-otc*: This dataset includes 5,881 nodes and 35,592 edges.

Both datasets represent web-of-trust networks from Bitcoin trading. In these networks, each node represents a user, and the interactions between users are private and voluntary. In particular, each node pair represents a completed transaction, and the edge weights indicate ratings given by customers to suppliers. These ratings are initially in the range of $[-10, 10]$, but for our analysis, they are standardized to the interval $[-1, 1]$.

7.2 Comparison methods

We evaluate our proposed attack method by comparing it with three baseline strategies:

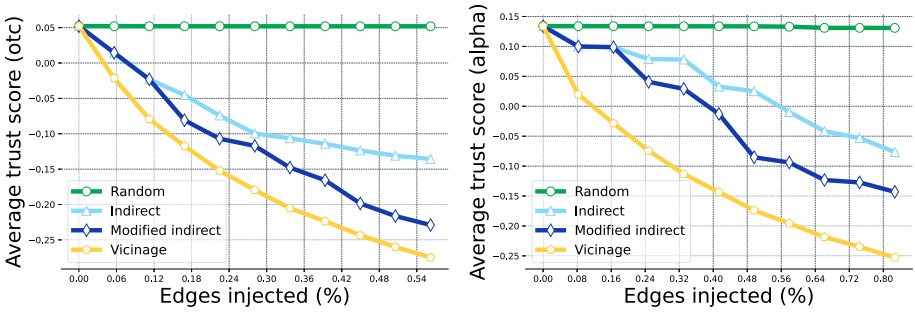


Fig. 5 Compare *vicinage*-attack with baselines. The experimental results are obtained by taking the average value of five independent experiments

Random Attack: This method involves randomly selecting edges from the perturbation space to manipulate trust.

Indirect Attack: This approach uses a brute-force method to examine all edges in the perturbation space. For each edge, we test its impact on the trust score or rank of target nodes. The edges are then ranked by their influence, and the top-k most influential edges are selected. This approach follows [17].

Modified Indirect Attack: This method improves upon the *Indirect Attack* by dividing the budget of k into n epochs. Within each epoch, the influence of all potential edges is ranked, and the top-$\frac{k}{n}$ edges are chosen as candidates. This approach is also detailed in [17].

For the modified indirect attacks, the performance depends on the parameter n. We found that even with $n = 1$, the brute-force method is more time-consuming compared to our proposed *vicinage*-attack method. To optimize computational efficiency, we set a maximum of optimization steps for our proposed method:

- For FGA: We use up to five steps.
- For REV2: We use up to four steps.

For the modified indirect attack, we set n as follows:

- For FGA: n is set to five.
- For REV2: n is set to four.

7.3 Adversarial attack on FGA

7.3.1 Experiment results

Figure 5 shows that the *vicinage*-attack method outperforms the baseline strategies in terms of effectiveness. The *Indirect*-attack and *Modified Indirect*-attack methods do not adequately solve the combination optimization problem. We aim to ensure that $\Phi_{S,T}(B_4)$ generates only *weak ties*. Specifically, $\Phi_{S,T}(B_4)$ is defined as $\Phi_{S,T}(B_1) \cup \Phi_{S,T}(B_2) \cup \Phi_{S,T}(B_3) - \Phi_{S,T}(B_0)$. Figure 6 compares the results from different perturbation spaces: $\Phi_{S,T}(B_0)$, $\Phi_{S,T}(B_1)$, $\Phi_{S,T}(B_2)$, $\Phi_{S,T}(B_3)$, and $\Phi_{S,T}(B_4)$. The results indicate that perturbations from $\Phi_{S,T}(B_4)$ are less effective, highlighting the importance of *strong ties* in making adversarial attacks more effective.

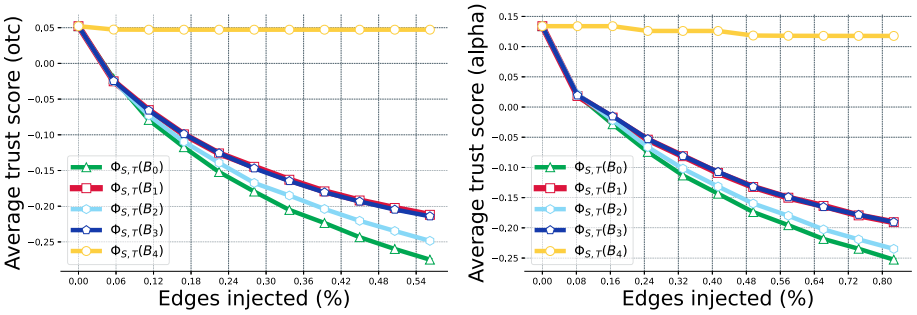


Fig. 6 Compare strong tie with weak ties. The experimental results are obtained by taking the average value of five independent experiments

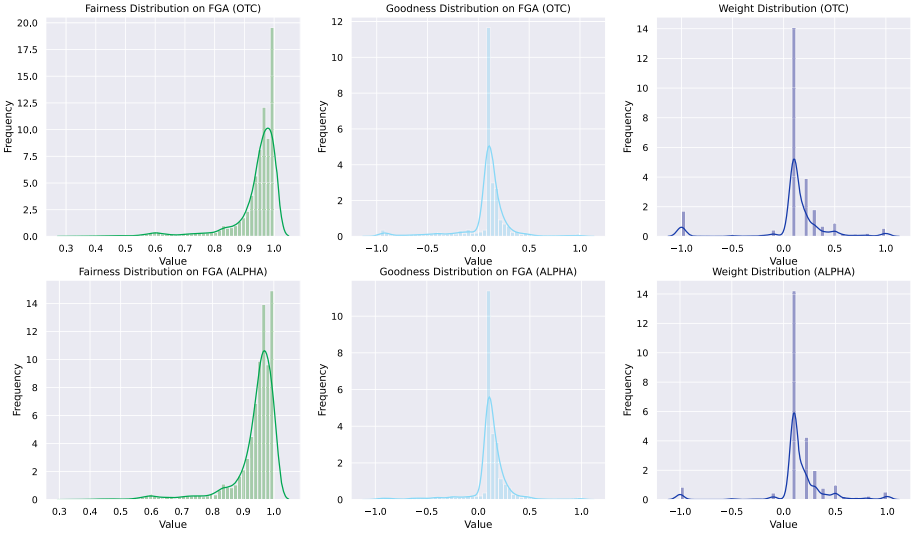


Fig. 7 Distribution of fairness and goodness on FGA, and distribution of edge weights

7.3.2 Prerequisite of strong tie

We observed that Strong Tie II is ineffective when the weight of the injected edge is set to -1. This section examines why this happens, using statistical analysis and the prerequisites for *strong ties*.

For Strong Tie II, Case I is defined as $f(p)w_0 < g(s) < w_0$. This condition cannot be satisfied if $w_0 < 0$, where w_0 is the weight of the injected edge. Case II is defined as $g(q) > \max(w_0, f(s)w_0)$ and $w_0 > \frac{1}{2-f(s)}$. Figure 7 shows the distributions of fairness and goodness in FGA, as well as the distribution of edge weights. Most fairness values are in the range (0.9, 1), while most goodness values are in (0.5, 0.25). Most edge weights are also in (0.5, 0.25). Satisfying Case II is difficult because $w_0 > \frac{1}{2-f(s)}$ implies w_0 should be close to 1, which conflicts with the assumption of $w_0 = -1$.

For Strong Tie I, Case I is defined as $f(s)w_0 < g(q) < w_2$. Case II is defined as $g(q) > \max(w_2, f(s)w_0)$ and $w_0 < \frac{2w_2-1}{f(s)}$. Case II can be approximated as $g(q) > w_2$

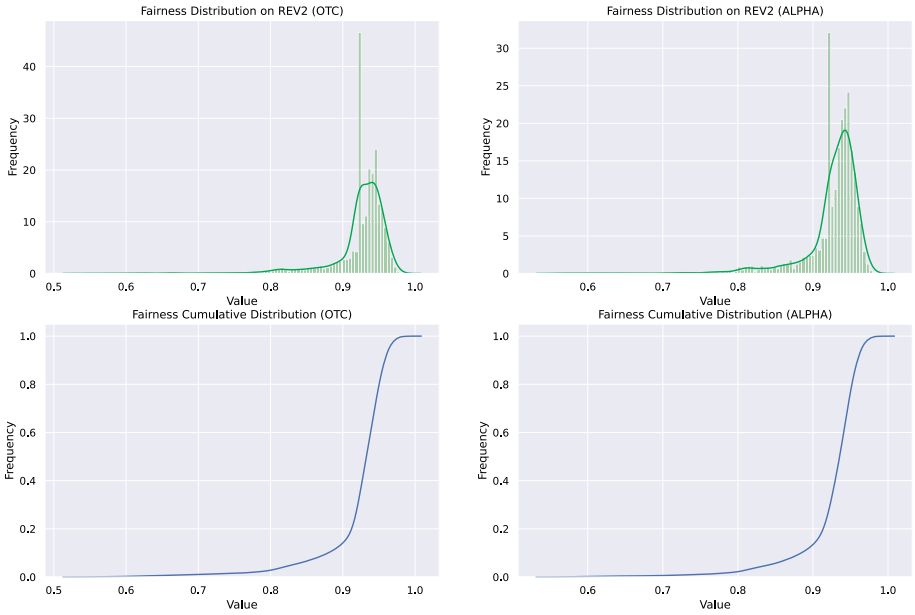


Fig. 8 Fairness distribution on REV2

and $w_0 < \frac{2w_2-1}{f(s)}$. Given the majority distribution, the requirement can be simplified to $w_0 < \frac{2w_2-1}{f(s)}$. This condition is more easily met when $w_0 = -1$.

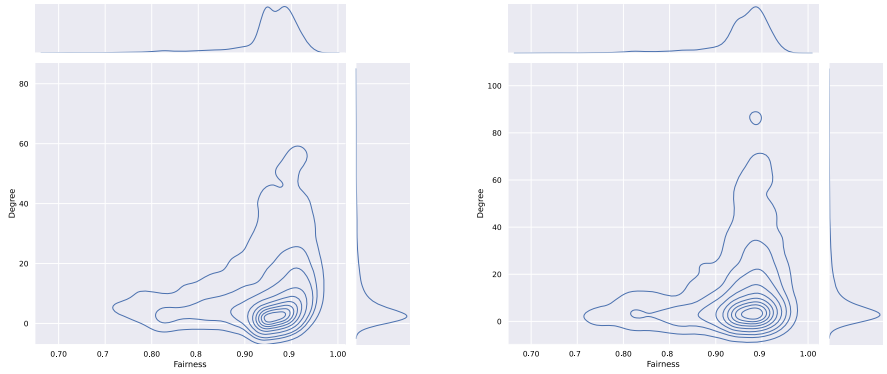
7.4 Adversarial attack on REV2

7.4.1 Experiment setting

In the REV2 experiments, we evaluate the effectiveness of the *vicinage*-attack by observing how trust propagates through the network. We focus on nodes with low fairness as our target nodes, assuming they would be motivated to avoid being labeled as anomalies or unfair. Consequently, attackers may attempt to enhance the trustworthiness of these nodes. Figure 8 shows the cumulative distribution function (CDF) of fairness. There is a gradual increase in the range $(0.8, 0.9)$ and a steep rise in the range $(0.9, 1)$. This suggests an opportunity for attackers to manipulate trust ranks. Target nodes T have fairness $f(t) < 0.88$, while high-fairness nodes S have $f(s) > 0.96$ and $d(s) > d_{\text{threshold}}$. Figure 9 shows that the degree values cluster around $d_{\text{threshold}}$.

7.4.2 Experiment results on REV2

We aim to identify *preferential paths* in REV2. The perturbation space includes both *positive attacks* and *passive attacks*, with interactions involving 1-hop and 2-hop neighbors. Figures 10, 11, and 12 show that the *vicinage*-attack method outperforms the baseline methods in finding the best perturbation edges. The *caveat line* in these figures indicates that 10% of nodes, with the lowest fairness, are considered unfair or anomalies.



(a) Joint distribution of fairness and degree on REV2 in *bitcoin-otc*.

(b) Joint distribution of fairness and degree on REV2 in *bitcoin-alpha*.

Fig. 9 **a** Joint distribution of fairness and degree on REV2 in *bitcoin-otc*. **b** Joint distribution of fairness and degree on REV2 in *bitcoin-alpha*

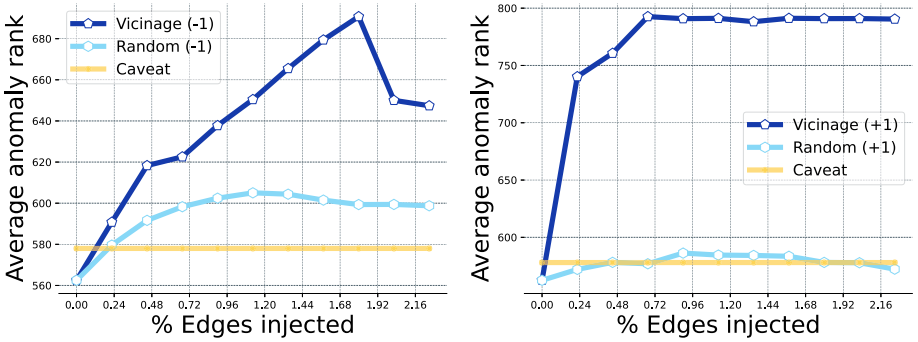


Fig. 10 Compare *vicinage* with baseline on *bitcoin-otc*. The experimental results are obtained by taking the average value of five independent experiments

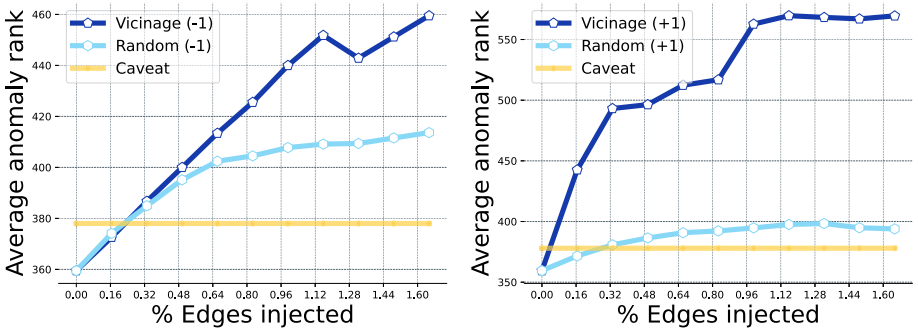


Fig. 11 Compare *vicinage* with baseline on *bitcoin-alpha*. The experimental results are obtained by taking the average value of five independent experiments

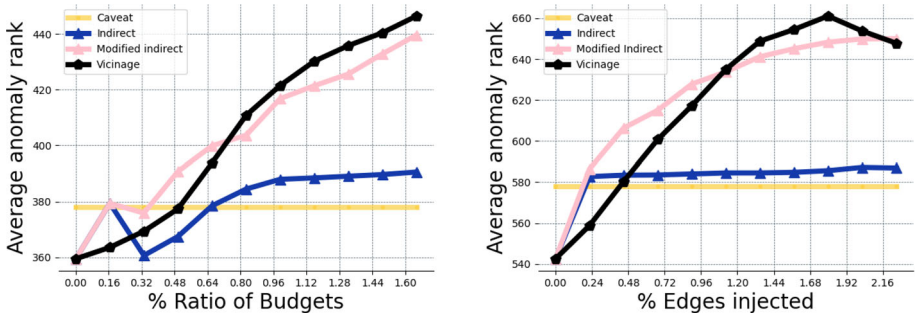


Fig. 12 Compare *vicinage* with baselines on *bitcoin-alpha* and *bitcoin-otc*. The experimental results are obtained by taking the average value of five independent experiments

When $n < 4$: The *vicinage*-attack is more effective than both the indirect attack and modified indirect attack methods, demonstrating its efficiency with fewer iterations.

When $n = 4$: The performance of the modified indirect attack approaches that of the *vicinage*-attack, suggesting that with enough iterations, it can be competitive, though it is still more time-consuming.

When $n > 4$: The modified indirect attack becomes increasingly time-consuming with little improvement in performance, highlighting the inefficiency of exhaustive methods compared to the *vicinage*-attack.

The gradient-based approach of the *vicinage*-attack enables more efficient exploration of the perturbation space, allowing for quicker identification of strong ties and preferential paths, which are crucial for effective adversarial attacks on trust prediction systems.

Figures 13, 14, 15, and 16 provide additional results. We define four perturbation spaces: $\Phi_{S,T}(B_0)$, $\Phi_{S,T}(B_1)$, $\Phi_{S,T}(B_2)$, and $\Phi_{S,T}(B_3)$. These spaces involve edges connecting high-fairness nodes S to $Succ(Pred(T))$, $Pred(Succ(T))$, $Succ(Succ(T))$, and $Pred(Pred(T))$. We also derive subsets from these perturbation spaces, denoted as $\Phi_{S,T}(B_4)$, $\Phi_{S,T}(B_5)$, $\Phi_{S,T}(B_6)$, and $\Phi_{S,T}(B_7)$, as follows:

$$\begin{cases} \Phi_{S,T}(B_4) = \Phi_{S,T}(B_0) - \Phi_{S,T}(B_1) - \Phi_{S,T}(B_2) - \Phi_{S,T}(B_3); \\ \Phi_{S,T}(B_5) = \Phi_{S,T}(B_1) - \Phi_{S,T}(B_0) - \Phi_{S,T}(B_2) - \Phi_{S,T}(B_3); \\ \Phi_{S,T}(B_6) = \Phi_{S,T}(B_2) - \Phi_{S,T}(B_0) - \Phi_{S,T}(B_1) - \Phi_{S,T}(B_3); \\ \Phi_{S,T}(B_7) = \Phi_{S,T}(B_3) - \Phi_{S,T}(B_0) - \Phi_{S,T}(B_1) - \Phi_{S,T}(B_2); \end{cases} \tag{9}$$

We also define two perturbation spaces for 1-hop neighbors: $\Phi_{S,T}(B_8)$ and $\Phi_{S,T}(B_9)$, which include edges connecting high-fairness nodes S to $Pred(T)$ and $Succ(T)$. We then derive two subsets: $\Phi_{S,T}(B_{10})$ and $\Phi_{S,T}(B_{11})$, defined as:

$$\begin{cases} \Phi_{S,T}(B_{10}) = \Phi_{S,T}(B_8) - \Phi_{S,T}(B_9); \\ \Phi_{S,T}(B_{11}) = \Phi_{S,T}(B_9) - \Phi_{S,T}(B_8). \end{cases} \tag{10}$$

Finally, Fig. 17 presents a broader analysis with additional subsets derived from the universe of perturbation spaces. These subsets are:

$$\begin{cases} \Phi_{S,T}(B_{12}) = \Phi_{S,T}(B_1) \cup \Phi_{S,T}(B_2) \cup \Phi_{S,T}(B_3) - \Phi_{S,T}(B_0); \\ \Phi_{S,T}(B_{13}) = \Phi_{S,T}(B_0) \cup \Phi_{S,T}(B_2) \cup \Phi_{S,T}(B_3) - \Phi_{S,T}(B_1); \\ \Phi_{S,T}(B_{14}) = \Phi_{S,T}(B_0) \cup \Phi_{S,T}(B_1) \cup \Phi_{S,T}(B_3) - \Phi_{S,T}(B_2); \\ \Phi_{S,T}(B_{15}) = \Phi_{S,T}(B_0) \cup \Phi_{S,T}(B_1) \cup \Phi_{S,T}(B_2) - \Phi_{S,T}(B_3). \end{cases} \tag{11}$$

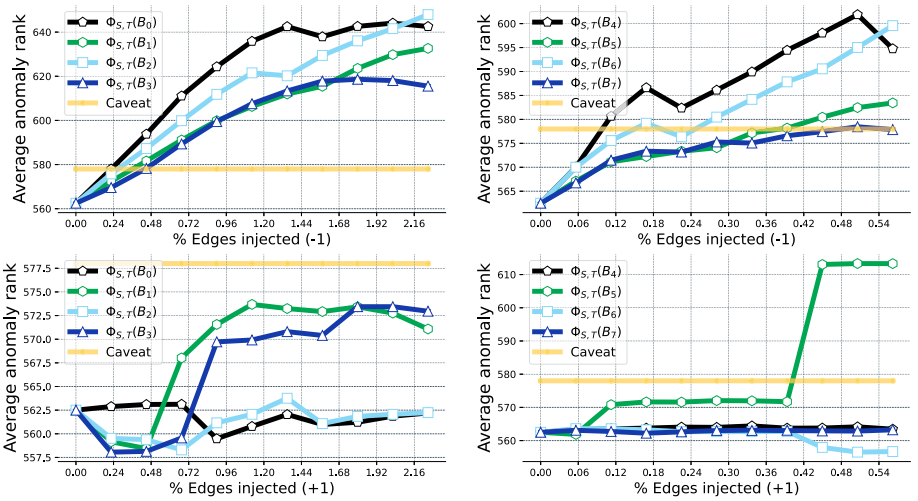


Fig. 13 Preferential path analysis over 2-hop cases on *bitcoin-otc*. The rank below the caveat is regarded as unfair or anomaly nodes. The experimental results are obtained by taking the average value of five independent experiments

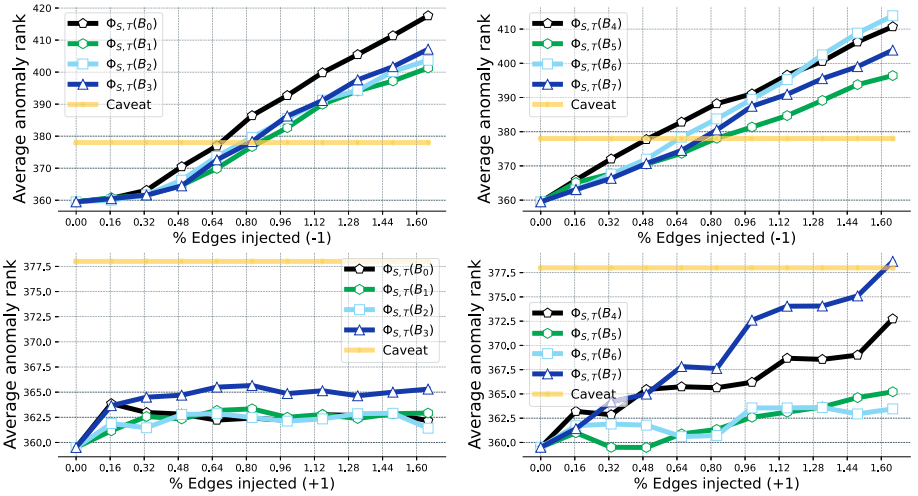


Fig. 14 Preferential path analysis over 2-hop cases on *bitcoin-alpha*. The experimental results are obtained by taking the average value of five independent experiments

$$
\begin{cases}
\Phi_{S,T}(B_{16}) = \Phi_{S,T}(B_2) \cup \Phi_{S,T}(B_3) - \Phi_{S,T}(B_0) - \Phi_{S,T}(B_1); \\
\Phi_{S,T}(B_{17}) = \Phi_{S,T}(B_1) \cup \Phi_{S,T}(B_3) - \Phi_{S,T}(B_0) - \Phi_{S,T}(B_2); \\
\Phi_{S,T}(B_{18}) = \Phi_{S,T}(B_1) \cup \Phi_{S,T}(B_2) - \Phi_{S,T}(B_0) - \Phi_{S,T}(B_3); \\
\Phi_{S,T}(B_{19}) = \Phi_{S,T}(B_0) \cup \Phi_{S,T}(B_3) - \Phi_{S,T}(B_1) - \Phi_{S,T}(B_2); \\
\Phi_{S,T}(B_{20}) = \Phi_{S,T}(B_0) \cup \Phi_{S,T}(B_2) - \Phi_{S,T}(B_1) - \Phi_{S,T}(B_3); \\
\Phi_{S,T}(B_{21}) = \Phi_{S,T}(B_0) \cup \Phi_{S,T}(B_1) - \Phi_{S,T}(B_2) - \Phi_{S,T}(B_3).
\end{cases}
\tag{12}
$$

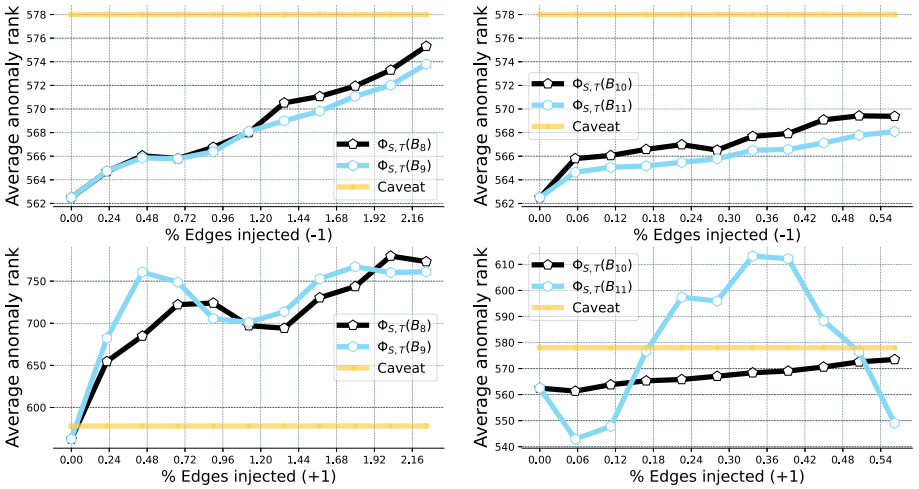


Fig. 15 Preferential path analysis over 1-hop cases on *bitcoin-otc*. The experimental results are obtained by taking the average value of five independent experiments

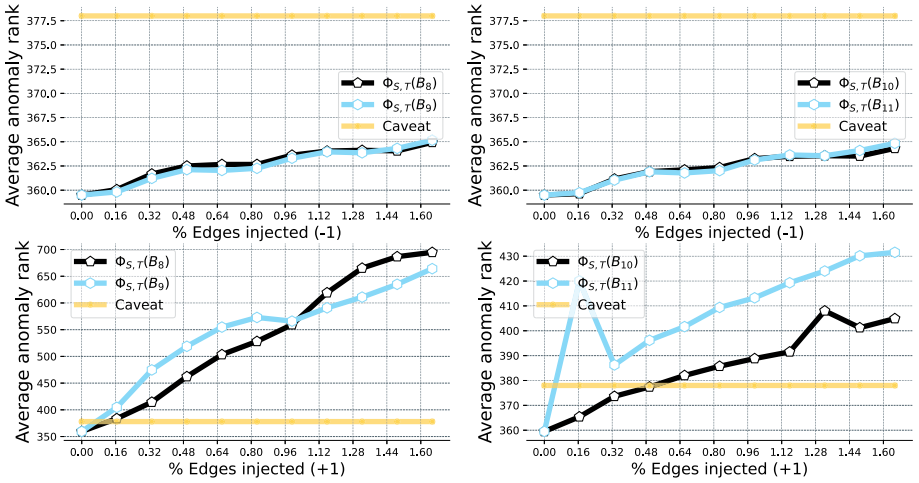


Fig. 16 Preferential path analysis over 1-hop cases on *bitcoin-alpha*. The experimental results are obtained by taking the average value of five independent experiments

8 Related realm and application

Our research highlights the presence of *preferential paths* within trust systems, which introduces a bias in how trust is propagated. This has important implications for improving trust systems, especially in fields such as e-commerce, online reviews, and peer-to-peer networks. Additionally, our findings emphasize the need for *vulnerability detection* to identify and prevent adversarial attacks that exploit these *preferential paths*. Furthermore, *preferential paths* can be leveraged in *trust chain discovery* to uncover implicit trust relationships.

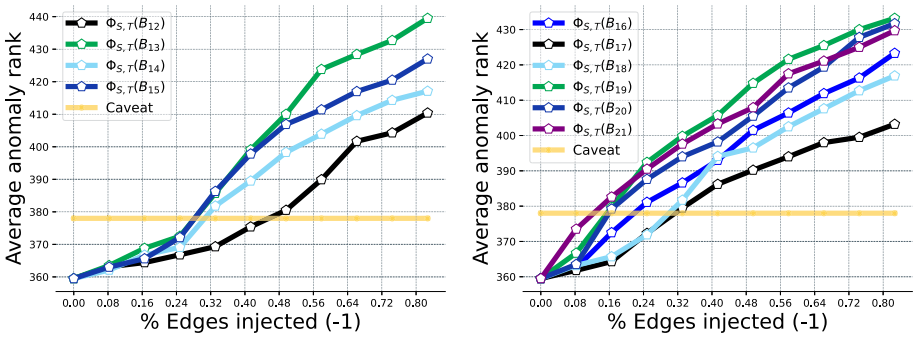


Fig. 17 Supplementary preferential path analysis over 2-hop cases on *bitcoin-alpha* from subsets of universe on constructed perturbation spaces

8.1 Vulnerability testing on trust systems

The *preferential paths* in trust systems can be seen as potential vulnerabilities. Adversarial techniques have been used to explore such vulnerabilities in graph-related tasks, such as backdoor attacks in graph classification [45–48]. In a trust system, target nodes can be carefully chosen and labeled by administrators, allowing adversarial techniques to find the most critical paths that could be exploited. Our proposed *vicinage*-attack serves as a method to test these vulnerabilities. Reference [49] identifies three key conditions for adversarial attacks: stealthiness, consistency, and inconsistency.

- **Stealthiness**, denoted as $\mathcal{S}(S, T)$, means that attacker nodes S can be several hops away from the target nodes T while still being effective in their attack. Defenders cannot easily distinguish between suspicious nodes and target nodes due to the limited number of injected edges, which form only a small part of the original graph.

- **Consistency**, denoted as $\mathcal{C}(G_0, G_\varepsilon)$, means that trust prediction results for most nontarget nodes remain stable or change only slightly after an attack. The attack is designed to target specific nodes, so the behavior of attackers may seem random but is actually purposeful.

- **Inconsistency**, denoted as $\mathcal{I}(G_0, G_\varepsilon)$, indicates that the trust prediction results for target nodes change significantly after an attack.

8.2 Trust chain discovery in trust systems

In this context, the perturbation space $\Phi_T(B) \in G_0$ should represent a modified version of the original graph, focusing on edges that are indirectly connected to the target nodes. This technique, sometimes called an *adversarial example* [49–52], incorporates *preferential paths* into the trust chain discovery process. By leveraging these paths, we can uncover hidden trust relationships within the network [53–55]. This approach can be particularly useful in cases where explicit trust endorsements are lacking or where indirect relationships need to be assessed.

Moreover, our findings suggest that trust chain discovery involving *preferential paths* may extend beyond a static evaluation of trustworthiness. It can involve monitoring dynamic changes in trust relationships over time. Analyzing the influence along *preferential paths* can reveal variations in trust levels, helping to identify anomalies or adversarial manipulations in the network.

9 Limitation and future work

9.1 Limitation on scalability and generalization

Although the *vicinage*-attack strategy represents a significant improvement in adversarial attacks on trust prediction systems, there are some limitations. First, the complexity of finding preferential paths and executing *vicinage*-attacks increases as the network size grows. This can make it challenging to apply the method to large-scale social networks or online platforms with millions of nodes and edges. Second, the *vicinage*-attack has been specifically designed for FGA and REV2, and its effectiveness in other trust prediction algorithms has not been extensively tested. This limits the generalizability of our results to other types of trust prediction systems.

9.2 Future work on explainable defense mechanism

To overcome these limitations and advance the field, future work will involve splitting large graphs into subgraphs based on their density and sparsity. This approach will not only address scalability issues but also help in developing an explainable defense mechanism, as different subgraphs may show unique characteristics and vulnerabilities. Additionally, exploring graph neural networks (GNNs) could be valuable for handling large-scale networks. GNNs may provide new insights into preferential behaviors in the context of adversarial attacks and help improve overall defense strategies.

10 Conclusion

In this paper, we have identified the presence of *preferential paths* within trust systems and their significant impact on how trust is spread. Based on these findings, we have developed a new approach called the *vicinage*-attack. This method specifically targets edges along *preferential paths* to strengthen adversarial attacks. Our research advances the field of adversarial attack techniques and provides a deeper understanding of trust propagation patterns.

By highlighting the propagation bias in both FGA and REV2, we offer valuable insights that can help build more robust trust systems and improve defense mechanisms. This work paves the way for better understanding of adversarial attacks on trust systems, ultimately contributing to the security and reliability of online platforms and social networks.

Acknowledgements This research was partly supported by the National Science Foundation of China (No. 62106210) and the Hong Kong Research Grant Council (No. PolyU25210821).

Author Contributions Yu Bu contributed to the conceptualization and design of the study, conducted data analysis and interpretation, and drafted the manuscript. Additionally, Yulin Zhu conducted a literature review, contributed to part of the source code, and participated in manuscript editing and revision. Longling Geng collected and processed experimental data, performed statistical analyses. Kai Zhou provided critical guidance throughout the study, and contributed to the revision of the manuscript.

Declarations

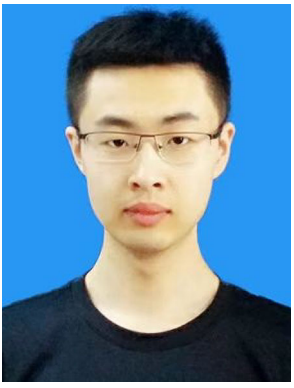
Competing interests The authors declare no competing interests.

References

1. Tang J, Chang Y, Aggarwal C, Liu H (2016) A survey of signed network mining in social media. ACM Comput Surv 49(3):1–37. https://doi.org/10.1145/2956185
2. Leskovec J, Huttenlocher D, Kleinberg J (2010) Signed networks in social media. In: Proceedings of the SIGCHI conference on human factors in computing systems, pp 1361–1370. https://doi.org/10.1145/1753326.1753532
3. Kunegis J (2014) Applications of structural balance in signed social networks. arXiv preprint arXiv:1402.6865
4. West R, Paskov HS, Leskovec J, Potts C (2014) Exploiting social network structure for person-to-person sentiment analysis. Trans Assoc Comput Linguist 2:297–310. https://doi.org/10.1162/tacl_a_00184
5. Jing Y, Wang H, Shao K, Huo X (2021) Relation representation learning via signed graph mutual information maximization for trust prediction. Symmetry 13(1):115. https://doi.org/10.3390/sym13010115
6. Vallarano N, Tessone CJ, Squartini T (2020) Bitcoin transaction networks: an overview of recent results. Front Phys 8:286. https://doi.org/10.3389/fphy.2020.00286
7. Evans DS (2014) Economic aspects of Bitcoin and other decentralized public-ledger currency platforms. University of Chicago Coase-Sandor Institute for Law Economics Research Paper (685). https://doi.org/10.2139/ssrn.2424516
8. Wijaya DA (2016) Extending asset management system functionality in bitcoin platform. In: 2016 international conference on computer, control, informatics and its applications (IC3INA). IEEE, pp 97–101. https://doi.org/10.1109/ic3ina.2016.7863031
9. Jethava G, Rao UP (2022) A novel trust prediction approach for online social networks based on multi-faceted feature similarity. Cluster Comput 25(6):3829–3843. https://doi.org/10.1007/s10586-022-03617-z
10. Jethava G, Rao UP (2022) An interaction-based and graph-based hybrid approach to evaluate Trust in Online Social Networks (OSNs). Arab J Sci Eng 47(8):9615–9628. https://doi.org/10.1007/s13369-021-06332-w
11. Jethava G, Rao UP (2024) Exploring security and trust mechanisms in online social networks: an extensive review. Comput Secur. https://doi.org/10.1016/j.cose.2024.103790
12. Kumar S, Spezzano F, Subrahmanian VS, Faloutsos C (2016) Edge weight prediction in weighted signed networks. In: 2016 IEEE 16th international conference on data mining (ICDM). IEEE, pp 221–230. https://doi.org/10.1109/icdm.2016.0033
13. Kumar S, Hooi B, Makhija D, Kumar M, Faloutsos C, Subrahmanian VS (2018) Rev2: fraudulent user prediction in rating platforms. In: Proceedings of the Eleventh ACM international conference on web search and data mining, pp 333–341. https://doi.org/10.1145/3159652.3159729
14. Lin D, Wu J, Fu Q, Zheng Z, Chen T (2023) RiskProp: account risk rating on ethereum via de-anonymous score and network propagation. arXiv preprint arXiv:2301.00354
15. Fu Q, Lin D, Wu J, Zheng Z (2023) A general framework for account risk rating on Ethereum: toward safer blockchain technology. IEEE Trans Comput Soc Syst. https://doi.org/10.1109/tcss.2023.3263382
16. Sun L, Dou Y, Yang C, Zhang K, Wang J, Philip SY, Li B (2022) Adversarial attack and defense on graph data: a survey. IEEE Trans Knowl Data Eng. https://doi.org/10.1109/tkde.2022.3201243
17. Lizurej T, Michalak T, Dziembowski S (2023) On manipulating weight predictions in signed weighted networks. arXiv preprint arXiv:2302.02687
18. Bu Y, Zhu Y, Geng L, Zhou K (2024) Uncovering strong ties: a study of indirect Sybil attack on signed social network. In: ICASSP 2024-2024 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 4535–4539. https://doi.org/10.1109/icassp48485.2024.10447587

19. Douceur JR (2002) The Sybil attack. In: International workshop on peer-to-peer systems. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 251–260. https://doi.org/10.1007/3-540-45748-8_24
20. Piro C, Shields C, Levine BN (2006) Detecting the Sybil attack in mobile ad hoc networks. In: 2006 Securecomm and workshops. IEEE, pp 1–11. https://doi.org/10.1109/seccomw.2006.359558
21. Yu H, Kaminsky M, Gibbons PB, Flaxman A (2006, August) Sybilguard: defending against Sybil attacks via social networks. In: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications, pp 267–278. https://doi.org/10.1145/1159913.1159945
22. Mohaisen A, Kim J (2013) The sybil attacks and defenses: a survey. arXiv preprint arXiv:1312.6349. https://doi.org/10.6029/smartcr.2013.06.009
23. Kleinberg JM (1998) Authoritative sources in a hyperlinked environment. In: SODA, vol 98, pp 668–677. https://doi.org/10.1515/9781400841356.514
24. Mishra A, Bhattacharya A (2011) Finding the bias and prestige of nodes in networks based on trust scores. In: Proceedings of the 20th international conference on World wide web, pp 567–576. https://doi.org/10.1145/1963405.1963485
25. Li RH, Xu Yu J, Huang X, Cheng H (2012) Robust reputation-based ranking on bipartite rating networks. In: Proceedings of the 2012 SIAM international conference on data mining. Society for Industrial and Applied Mathematics, pp 612–623. https://doi.org/10.1137/1.9781611972825.53
26. Akoglu L, Chandy R, Faloutsos C (2013) Opinion fraud detection in online reviews by network effects. In: Proceedings of the international AAAI conference on web and social media, vol 7, No. 1, pp 2–11. https://doi.org/10.1609/icwsm.v7i1.14380
27. Shahriari M, Jalili M (2014) Ranking nodes in signed social networks. Soc Netw Anal Min 4(1):1–12. https://doi.org/10.1007/s13278-014-0172-x
28. Traag V, Nesterov Y, Van Dooren P (2010) Exponential ranking: taking into account negative links. Soc Inform 2010:192–202. https://doi.org/10.1007/978-3-642-16567-2_14
29. Wu Z, Aggarwal CC, Sun J (2016) The troll-trust model for ranking in signed networks. In: Proceedings of the ninth ACM international conference on web search and data mining, pp 447–456. https://doi.org/10.1145/2835776.2835816
30. Leskovec J, Huttenlocher D, Kleinberg J (2010) Signed networks in social media. In: Proceedings of the SIGCHI conference on human factors in computing systems, pp 1361–1370. https://doi.org/10.1145/1753326.1753532
31. Yang SH, Smola AJ, Long B, Zha H, Chang Y (2012, August) Friend or frenemy? Predicting signed ties in social networks. In: Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval, pp 555–564. https://doi.org/10.1145/2348283.2348359
32. Wang Y, Wang X, Tang J, Zuo W, Cai G (2015) Modeling status theory in trust prediction. In: Proceedings of the AAAI conference on artificial intelligence, vol 29, no. 1. https://doi.org/10.1609/aaai.v29i1.9460
33. Guha R, Kumar R, Raghavan P, Tomkins A (2004) Propagation of trust and distrust. In: Proceedings of the 13th international conference on world wide web, pp 403–412. https://doi.org/10.1145/988672.988727
34. Lee W, Lee YC, Lee D, Kim SW (2021) Look before you leap: confirming edge signs in random walk with restart for personalized node ranking in signed networks. In: Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval, pp 143–152. https://doi.org/10.1145/3404835.3462923
35. Knapskog SJ (1998) A metric for trusted systems. In: Proceedings of the 21st National Security Conference. NSA, pp 16–29
36. De Cock M, Da Silva PP (2006) A many valued representation and propagation of trust and distrust. In: Fuzzy logic and applications: 6th international workshop, WILF 2005, Crema, Italy, September 15-17, 2005, Revised Selected Papers 6. Springer Berlin Heidelberg, pp 114–120. https://doi.org/10.1007/11676935_14
37. Victor P, Cornelis C, De Cock M, Pinheiro da Silva P (2006) Towards a provenance-preserving trust model in agent networks. In: WWW2006 conference proceedings, special interest tracks, posters and workshops
38. Dai H, Li H, Tian T, Huang X, Wang L, Zhu J, Song L (2018) Adversarial attack on graph structured data. In: International conference on machine learning. PMLR, pp 1115–1124
39. Wang B, Gong NZ (2019) Attacking graph-based classification via manipulating the graph structure. In: Proceedings of the 2019 ACM SIGSAC conference on computer and communications security, pp 2023–2040. https://doi.org/10.1145/3319535.3354206
40. Zügner D, Akbarnejad A, Günnemann S (2018) Adversarial attacks on neural networks for graph data. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, pp 2847–2856. https://doi.org/10.1145/3219819.3220078
41. Chen J, Wu Y, Xu X, Chen Y, Zheng H, Xuan Q (2018) Fast gradient attack on network embedding. arXiv preprint arXiv:1809.02797

42. Zügner D, Günnemann S (2019) Adversarial attacks on graph neural networks via meta learning. In: International conference on learning representations (ICLR). arXiv:1902.08412pdf
43. Zhu Y, Lai Y, Zhao K, Luo X, Yuan M, Ren J, Zhou K (2021) Binarizedattack: structural poisoning attacks to graph-based anomaly detection. arXiv preprint arXiv:2106.09989. https://doi.org/10.1109/icde53745.2022.00006
44. Taylor M, Guiver J, Robertson S, Minka T (2008) Softrank: optimizing non-smooth rank metrics. In: Proceedings of the 2008 international conference on web search and data mining, pp 77–86. https://doi.org/10.1145/1341531.1341544
45. Xi Z, Pang R, Ji S, Wang T (2021) Graph backdoor. In: 30th USENIX Security Symposium (USENIX Security 21), pp 1523–1540
46. Zhang Z, Jia J, Wang B, Gong NZ (2021) Backdoor attacks to graph neural networks. In: Proceedings of the 26th ACM symposium on access control models and technologies, pp 15–26. https://doi.org/10.1145/3450569.3463560
47. Xu J, Xue M, Picek S (2021) Explainability-based backdoor attacks against graph neural networks. In: Proceedings of the 3rd ACM workshop on wireless security and machine learning, pp 31–36. https://doi.org/10.1145/3468218.3469046
48. Yang S, Doan BG, Montague P, De Vel O, Abraham T, Camtepe S, Kanhere SS (2022) Transferable graph backdoor attack. In: Proceedings of the 25th international symposium on research in attacks, intrusions and defenses, pp 321–332. https://doi.org/10.1145/3545948.3545976
49. Wu B, Liu L, Zhu Z, Liu Q, He Z, Lyu S (2023) Adversarial machine learning: a systematic survey of backdoor attack, weight attack and adversarial example. arXiv preprint arXiv:2302.09457
50. Wu H, Wang C, Tyshetskiy Y, Docherty A, Lu K, Zhu L (2019) Adversarial examples on graph data: deep insights into attack and defense. arXiv preprint arXiv:1903.01610. https://doi.org/10.24963/ijcai.2019/669
51. Zhang J, Li C (2019) Adversarial examples: opportunities and challenges. IEEE Trans Neural Netw Learn Syst 31(7):2578–2593. https://doi.org/10.1109/tnnls.2019.2933524
52. Xiao C, Li B, Zhu JY, He W, Liu M, Song D (2018) Generating adversarial examples with adversarial networks. arXiv preprint arXiv:1801.02610. https://doi.org/10.24963/ijcai.2018/543
53. Malik S, Dedeoglu V, Kanhere SS, Jurdak R (2019, July) Trustchain: Trust management in blockchain and iot supported supply chains. In: 2019 IEEE international conference on blockchain (blockchain). IEEE, pp 184–193. https://doi.org/10.1109/blockchain.2019.00032
54. Otte P, de Vos M, Pouwelse J (2020) TrustChain: a Sybil-resistant scalable blockchain. Future Gener Comput Syst 107:770–780. https://doi.org/10.1016/j.future.2017.08.048
55. Bapna R, Gupta A, Rice S, Sundararajan A (2017) Trust and the strength of ties in online social networks. MIS Q 41(1):115–130



Yu Bu received his Bachelor's degree from Harbin Institute of Technology and his Master's degree from New York University. He is currently pursuing his PhD at The Hong Kong Polytechnic University. His research focuses on adversarial network analysis and adversarial machine learning, with a particular interest in developing robust algorithms and methodologies to address security challenges in machine learning systems.

Yulin Zhu received a B.S. degree from Wuhan University, Wuhan, China, in 2012 and a Ph.D. degree from The Chinese University of Hong Kong, HKSAR, in 2020. After that, he served as a research fellow and postdoctoral fellow with the Department of Computing, The Hong Kong Polytechnic University, HKSAR. Now, he works as an assistant professor at the Department of Computer Science, Hong Kong Chu Hai College, HKSAR. His research focuses on AI security, trustworthy graph learning, and graph mining. He has published several papers such as IEEE ICDE, ACM CCS, IEEE S&P, IEEE TKDE, IEEE TIFS, and IEEE TNNLS.

Longling Geng obtained her Bachelor's degree in Computing from the Department of Computing at The Hong Kong Polytechnic University (HKSAR), under the supervision of Li Qing and Zhou Kai. Her research focuses on a wide range of topics, including artificial intelligence systems, combinatorial optimization problems, data security, digital communication, dynamic information systems, environmental protection, feedback loops, and home security systems.

Kai Zhou received the PhD degree from the Department of Electrical and Computer Engineering, Michigan State University, in 2018. He is an assistant professor with the Department of Computing at The Hong Kong Polytechnic University. His research interests center around security with emphasis on adversarial network analysis, adversarial machine learning, and data security and privacy. He worked as a postdoc with the Department of Computer Science, Washington University, in St. Louis from 2018 to 2020.