

Article

TPoison: Data-Poisoning Attack against GNN-Based Social Trust Model

Jiahui Zhao ^{1,*}, Nan Jiang ^{1,*}, Kanglu Pei ², Jie Wen ³, Hualin Zhan ¹ and Ziang Tu ¹

¹ College of Information Engineering, East China Jiaotong University, Nanchang 330013, China; 2021068081200002@ecjtu.edu.cn (J.Z.); james392@sohu.com or 2018029081100006@ecjtu.edu.cn (H.Z.); tza362202@163.com or 2021068081200004@ecjtu.edu.cn (Z.T.)

² School of Mathematics and Statistics, the University of Sydney, Camperdown, NSW 2006, Australia; kanglu.pei@sydney.edu.au

³ College of Electrical and Automation Engineering, East China Jiaotong University, Nanchang 330013, China; wenjie412@126.com or 2019029081100008@ecjtu.edu.cn

* Correspondence: jiangnan1018@gmail.com or jiangnan@ecjtu.edu.cn

Abstract: In online social networks, users can vote on different trust levels for each other to indicate how much they trust their friends. Researchers have improved their ability to predict social trust relationships through a variety of methods, one of which is the graph neural network (GNN) method, but they have also brought the vulnerability of the GNN method into the social trust network model. We propose a data-poisoning attack method for GNN-based social trust models based on the characteristics of social trust networks. We used a two-sample test for power-law distributions of discrete data to avoid changes in the dataset being detected and used an enhanced surrogate model to generate poisoned samples. We further tested the effectiveness of our approach on three real-world datasets and compared it with two other methods. The experimental results using three datasets show that our method can effectively avoid detection. We also used three metrics to illustrate the effectiveness of our attack, and the experimental results show that our attack stayed ahead of the other two methods in all three datasets. In terms of one of our metrics, our attack method decreased the accuracies of the attacked models by 12.6%, 22.8%, and 13.8%.

Keywords: social trust model; graph neural network; data-poisoning attack

MSC: 68T07



Citation: Zhao, J.; Jiang, N.; Pei, K.; Wen, J.; Zhan, H.; Tu, Z. TPoison: Data-Poisoning Attack against GNN-Based Social Trust Model. *Mathematics* **2024**, *12*, 1813. <https://doi.org/10.3390/math12121813>

Academic Editor: Andrea Scozzari

Received: 22 May 2024

Revised: 5 June 2024

Accepted: 7 June 2024

Published: 11 June 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction and Preliminaries

1.1. Introduction

In the era of the Internet, online social networks have evolved into an essential component of people's everyday existence. On these online social networking platforms, users can give different social trust ratings to other users. Social trust voting reflects the degree of trust between users and is extremely important for online marketing, recommender systems, and other related applications to predict social trust voting between users. The researchers proposed the social trust model to predict unknown social trust voting.

Social trust models are becoming increasingly vital across various contexts. They can be used in e-commerce to increase the purchase rates of products, help users to identify malicious behavior by others on modern social platforms, etc. When social trust models are attacked and become unreliable, applications that use social trust models are also severely affected.

Attacks on the traditional social trust model have a long history. Malicious attackers disguise multiple identities and establish intentional social relationships with real users to manipulate the entire social trust network and reduce its reliability, which becomes a sybil attack [1]. One attacker may not be able to manipulate the whole social network;

two or more attackers adopt the collusion attack [2] method to launch a coordinated attack to increase the success rate of the attack. With in-depth research, attack methods, such as a conflicting behavior attack [3] and on-off attack [4], were proposed, which diminish the trust voting of genuine users or elevate the voting of malicious users, and ultimately result in a decline in the reliability of the social trust model. Moreover, a control chart [5,6] was first applied to quality management in industrial production for risk control and the monitoring of anomalies in production, but now researchers have used control charts to monitor anomalies in social networks [7–9]. The successful attack of the above methods on traditional social trust models indicates that these traditional methods did not consider robustness issues in their design.

In recent years, researchers demonstrated that graph neural networks can successfully learn graph data in several scenarios, such as node classification and link prediction, with results in a leading position [10–12]. With the success of graph neural networks, researchers have increased their investigations of social trust scenarios [13]. When social trust models use graph neural networks to improve performance, the threats against graph neural networks are also brought into play [14,15]. Multiple white-box attacks against graph neural networks were proposed: FGA [16] can quickly generate attacks using existing nodes by leveraging the model's gradient information. Greedy-GAN [17] injects new attack nodes into the graph through a greedy algorithm. MGA [18] enables the attack to escape the local optima and generate a stronger attack using the momentum gradient method. TGA [19] creates attack nodes by comparing the gradient information of deep neural network embeddings across different time snapshots, and NICKI [20] targets specific types of nodes by injecting new nodes.

In recent years, numerous researchers have directed their efforts toward enhancing the performance of social trust models (e.g., prediction accuracy, F1 scores) without considering the robustness of the models. A specific attack method against neural networks is called data poisoning, where an attacker can inject carefully crafted poisoned data into the training model, causing the neural network to produce wrong predictions, reducing the performance of the neural network model. Data-poisoning attacks have already posed a threat to applications such as speech recognition systems [21], spam-filtering systems [22], recommendation systems [23–25], and facial recognition systems [26]. In this case, data-poisoning attacks will compromise the reliability of the graph-neural-network-based social trust model, which, in turn, may lead to serious financial losses and adverse social impacts. Overall, reducing the performance of the model will benefit the attacker, either in terms of business or ethical considerations.

Although the previous work shown in Table 1 has achieved convincing results in the attack of traditional social trust models, such as a sybil attack and its variants conflict behavior attack, on-off attack, and other attack methods, to compromise the social trust network, making it less trustworthy, they do not attack the relevant features of the social trust model itself, especially the GNN-based social trust model. This study aimed to bridge the gap in this area by evaluating the impact of our data-poisoning method on GNN-based social trust models and further substantiating the problem of the vulnerability and robustness of social trust models. We designed a data-poisoning attack method against graph-neural-network-based social trust models and demonstrated that the attacker is able to effectively degrade the performance of the social trust model while evading detection. Also compared with other attacks against a graph neural network, such as in [27], this attack method does not need to inject a lot of nodes into the graph (creating new users), which aids the attack since creating a large number of new users may be detected and blocked by social network platforms. Our method is more stealthy and only requires the creation of social relationships between existing users.

In summary, the main contributions of this paper are as follows:

1. Based on the characteristics of online social networks, we propose a data-poisoning attack method in a gray-box environment for GNN-based social trust models.

2. If the node degree distribution changes significantly, the attack will be very easy to detect. To overcome this problem, we used a two-sample test for power-law distributions of discrete data to keep the node degree distribution relatively unchanged and preserve important social relationships.
3. To showcase the effectiveness of the method, we performed a series of quantitative and qualitative experiments on three real-world datasets. The results demonstrate that our method exhibited superior attack performance compared with the other attack methods.

Table 1. Recent research on attack and defense of social trust models.

Title	Authors	Year	Innovation Points
The sybil attack [28].	Douceur, J.R.	2002	The first attack method on social networks.
SybilGuard: Defending Against Sybil Attacks via Social Networks [2].	Yu, H.; Kaminsky, M.; Gibbons, P.B.; Flaxman, A.D.	2008	A decentralized defense method against a sybil attack is proposed based on user characteristics.
SybilLimit: A Near-Optimal Social Network Defense against Sybil Attacks [1].	Yu, H.; Gibbons, P.B.; Kaminsky, M.; Xiao, F.	2008	Based on Sybilguard, the use of the near-optimal approach allows for a further increase in the number of attack edges.
The social web and privacy: Practices, reciprocity and conflict detection in social networks [29].	Gürses, S.; Berendt, B.	2010	Detecting conflict behavior attacks and modeling the real-world impacts of such attacks in social networks.
Pollution, bad-mouthing, and local marketing: The underground of location-based social networks [30].	Costa, H.; Merschmann, L. H.; Barth, F.; Benevenuto, F.	2014	Research into attack and defense measures against bad-mouthing attacks within social networks, alongside the initial validation of the impact of such attacks in real environments.
On-off attack management based on trust [31].	Sony, S.M.; Sasi, S.B.	2016	Defense mechanisms against on-off attacks utilizing predictability trust and sliding windows, coupled with a practical assessment to validate the efficacy of on-off attacks within social networks.
SybilFlyover: Heterogeneous graph-based fake account detection model on social networks [32].	Siyu, L.; Jin, Y.; Gang, L.; Tianrui, L.; Kui, Z.	2022	Using the utilization of social data sourced from social networks for attacks and defenses.
Adversarial attacks against dynamic graph neural networks via node injection [27].	Yanan, J.; Hui, X.	2024	Node injection attacks on dynamic graph neural networks based on graph structure vulnerability.

The main contents of the subsequent sections of this paper are as follows: The second and third parts of Section 1 are an introduction to the problem scope and related preliminary knowledge of the data-poisoning attack on social trust models based on graph neural networks. Section 2 is a detailed description of the three modules in our attack. Section 3 analyzes the impact produced by using different attack methods on the performance of three real-world datasets in the victim model and shows that our attack method produced a stronger attack effect compared with the other attack methods. Section 4 summarizes the main work of this paper and suggests future research directions.

1.2. Problem Scope

With the rapid development of artificial intelligence technology, especially the development and wide application of graph neural networks, the social trust model based on graph neural networks has demonstrated a powerful ability to predict social trust evaluation. Compared with traditional social trust models, graph-neural-network-based models are better able to capture complex dependencies in graph structures, thus improving the accuracy of prediction. However, as graph neural networks are widely used in various fields, their robustness problem has gradually attracted the attention of researchers.

In graph neural networks, data-poisoning attacks are a typical malicious attack method. The attacker seeks to compromise the model's performance by poisoning the dataset. Due to the problem of robustness related to graph neural networks, social trust models based on

graph neural networks may also be affected by this. Once the attacker successfully poisons the dataset, the social trust voting predicted by the graph-neural-network-based social trust model is likely to be biased, thus affecting its performance in practical applications and, hence, commercial interests.

Because of the inherent openness and anonymity, nearly anyone can readily access a user's social relationships on these platforms. Users in social networks can be attacked by methods such as social engineering [33]. In order to more closely match the real-world attack environment, a data-poisoning attack method against the graph neural network social trust model in a gray-box environment is proposed. This is a more difficult but more contextualized real-world attack environment than the white-box environment, and this setting makes the research more difficult and challenging.

In the context of a gray-box environment, the attack is restricted to accessing only a constrained subset of information pertaining to the model, only the relationships between the inputs and the predicted outputs, and cannot get a full picture of the specific network structure and learning parameters of the model. The attacker cannot directly obtain detailed information about the model, as in the case of white-box attacks, and can only discover the weaknesses of the attacked model through other methods. Attacking in a gray-box environment is more challenging compared with a white-box attack, but it also provides a more realistic simulation environment for the attack, which helps to expose the security issues of GNN-based social trust models.

1.3. Preliminaries

The study of social trust modeling involves a number of disciplines, such as psychology, sociology, and computer science, and is an interdisciplinary study. Predicting social trust evaluations is an even more complex problem that requires the consideration of various factors. In previous studies, modeling social trust was usually done by traversing the traditional method of trust propagation from the user who gives the vote to the user who receives the vote for the whole social trust network to learn and predict social trust evaluations. These methods require computing a large amount of data in the graph and analyzing each node and edge individually to determine the propagation and aggregation rules of trust. There are still problems, such as computational inefficiency and high resource demand, when applied to larger-scale social networks.

Therefore, researchers further explored how to satisfy the need to improve the computational efficiency and reduce computational resources while maintaining high model performance. In addition, machine learning techniques are also combined with the relevant features of social trust in cross-disciplines to fully utilize the advantages of both to achieve more accurate and efficient predictions.

With the continuous development of GNNs in recent years, GNN-related techniques were shown to be able to learn on graph-structured data and have achieved advanced results in applications such as community detection and link prediction. Social trust modeling using graph neural network techniques has better task performance, a shorter computation time, and fewer computational resource requirements compared with other methods.

To describe our attack method in detail, we first define a social trust network as $G = (F, R, V)$. Let $F = \{f_1, f_2, \dots, f_n\}$ be the set of all users in a social trust network, where f_i represents user i and n denotes the total count of users within the network. $R = \{r_{1 \rightarrow 2}, r_{2 \rightarrow 1}, \dots, r_{(n-1) \rightarrow n}\}$ is the set of social relations between all users. Specifically, if $r_{m \rightarrow n} = 1$, then this means user m is a trustor, user n is a trustee, and there is a one-way trustor–trustee social relationship between user m and user n . Let $V = \{v_{1 \rightarrow 2}, \dots, v_{(n-1) \rightarrow n} | r_{1 \rightarrow 2}, \dots, r_{(n-1) \rightarrow n} \in R\}$ be the social trust vote given by the trustor to the trustee in a known social relationship. The social trust model predicts the missing social trust vote $\tilde{V}$ by learning the known social relationships R and social trust vote V in the network such that $\tilde{V} = \{\tilde{v}_{m \rightarrow n} | r_{m \rightarrow n} \notin R\}$.

As shown in Figure 1, our study was built in a gray-box attack setting. Due to the openness of social networks, we assumed that the social trust network dataset was already

known, but we could not access the neural network structure, learning parameters, and other contents of the model. We define the social trust model as a learning classifier:

$$f = \sigma(\text{GNNs}(F, R, V)) \quad (1)$$

where σ represents the activation function and GNNs represent the unknown graph neural network methods.

Our objective was to optimize the disparity in model performance between the dataset before poisoning $G = (F, R, V)$ and the dataset after poisoning $G' = (F, R', V')$.

$$\max \sum_{i=1}^n \sum_{j=1}^n |f_{i \rightarrow j}(F, R', V') - f_{i \rightarrow j}(F, R, V)| \quad (2)$$

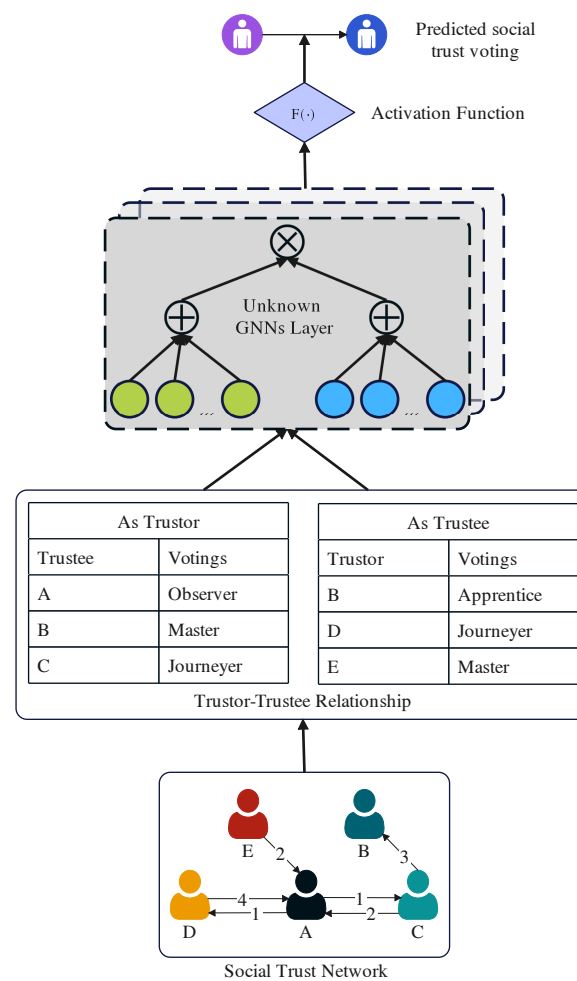


Figure 1. Illustration of GNN-based social trust model.

2. Poisoning Method

We propose a data-poisoning attack method on graph-neural-network-based social trust models in a gray-box environment that can generate poisoning samples that have an impact on the attacked model and can effectively avoid detection. Attacks on gray-box environments have more practical value for the protection of and enhancing the robustness of the models.

2.1. Avoidance Detection Module

Usually, data-poisoning attacks need to put a sufficient proportion of poisoned samples in the original dataset to have an impact on the model performance, but a large number of poisoned samples are easily detected by the victims. Social relations are different from text

or images, and it is difficult to detect the difference before and after poisoning by visual inspection. To avoid poisoning samples being detected, we first limited the maximum number of modified social relations R' and trust votes V' by ε .

$$\sum[|R' - R| + |V' - V|] \leq \varepsilon \quad (3)$$

Furthermore, to ensure that our poisoned samples were similar enough to the original dataset, we wanted to retain important network features in the social trust network and that our perturbations did not overly affect the node degree distribution in the social trust network. We refer to a two-sample test for power-law distributions of discrete data [34], which for a large sample, will obey a chi-square distribution with one degree of freedom.

$$\Lambda = -2 \times l(H_0 | R \cup R') + 2 \times l(H_1 | R) + 2 \times l(H_1 | R') \quad (4)$$

We set up significance tests to evaluate whether R and R' are from an identical power-law distribution. The null hypothesis H_0 is satisfied if R and R' come from the identical power-law distribution and hypothesis H_1 is satisfied if R and R' come from different power-law distributions.

In order to calculate the log-likelihood value l in Equation (4), we used the maximum likelihood estimation (MLE) [35] to give accurate parameter estimates; given a sample $|R| = \{|r_1|, |r_2|, \dots, |r_i| | i \in F, |r_i| \geq r_{min}\}$, the log-likelihood function is

$$l(\alpha, r_{min}) = n \ln \alpha + n \alpha \ln r_{min} + (\alpha + 1) \sum_{i,j \in F, r_i \rightarrow j \in R} \ln |r_i| \quad (5)$$

where $|r_i|$ represents the number of friends of user i and r_{min} denotes the minimum number of friends required for a user to be included in the computation. In the discrete case, the calculation of α in the power-law distribution only applies to the case $r \geq r_{min}$. n is the number of all users in a social network that satisfies $r \geq r_{min}$.

Although we are currently unable to exactly calculate the scaling parameter α in the discrete case, an approximate expression is given by [34]

$$\alpha_R \simeq 1 + n \left[\sum_{i \in F, r_i \in R} \ln \frac{r_i}{r_{min} - \frac{1}{2}} \right]^{-1} \quad (6)$$

Therefore, we can calculate Λ to determine whether R and R' come from the same power-law distribution. We used the classical p-value of 0.05 as the threshold value for accepting or rejecting the null hypothesis.

$$\Lambda \leq 0.05 \quad (7)$$

Equations (3) and (7) form a challenging constraint $C_{\varepsilon, \Lambda}^R$, and the ultimate goal of the non-detectable design is to satisfy that R and R' possess the same trend in the node degree distribution.

Overall, the avoidance detection module initially scrutinizes the node degree distribution of the original dataset to verify that the poisoned dataset maintained adherence to the power-law distribution. Subsequently, by employing the two-sample test method for assessing the power-law distribution of discrete data, samples that minimally alter the power-law distribution are identified as the candidate set of poisoning samples. The objective was to ensure that the poisoning samples preserve the power-law distribution of the original sample set to the greatest extent possible.

This approach effectively preserves the integrity of influential nodes within the original dataset. Given that influential nodes play a significant role in the dataset, any substantial alteration to them could potentially disrupt the connectivity and topology of the dataset, thereby impacting the transmission of social trust.

The flow chart of the avoidance detection module is given in Figure 2, which is used to illustrate the design and algorithm of the avoidance detection module more clearly.

Through the above steps, it is possible to effectively generate poisoning samples that have less impact on the degree of distribution of the nodes.

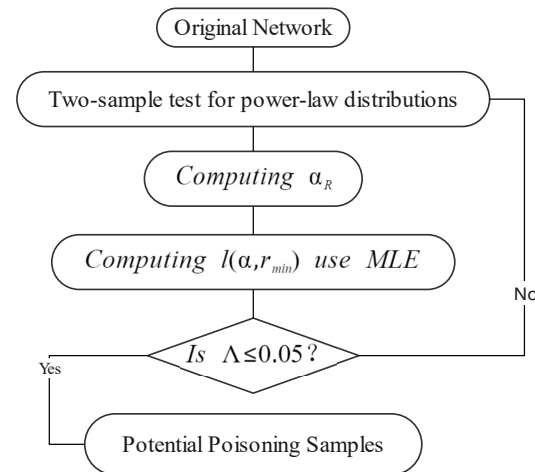


Figure 2. The flow chart of the avoidance detection module.

2.2. Gray-Box Attack Module

In [36], a network was characterized by its asymmetry and propagative nature. To illustrate the characteristics of such a network, a small network was constructed as shown in Figure 3. The asymmetry of voting is evident when considering the perspectives of users A and C in the network. The voting of user A regarding user C ($A \rightarrow C$) is different from the social trust voting of user C regarding user A ($C \rightarrow A$). At the same time, there is a vote between user C and user B ($C \rightarrow B$), and based on the feature of social network propagability, $A \rightarrow C \rightarrow B$ forms a potential social trust evaluation ($A \rightarrow B$).

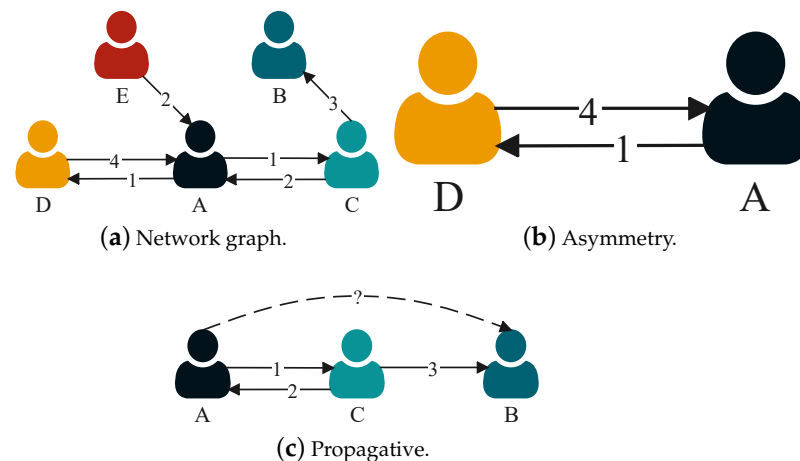


Figure 3. The feature illustrations of a social trust network: an example.

Even though we have access to nearly all social relationships, we could not steal the parameters of the trained model and explore which GNN method was used in the social trust model. We did not focus on the specific GNN method used by the social trust model; we only adapted the poisoning method based on the characteristics of social trust networks so that we could effectively attack the social trust model.

A surrogate model approach for undirected graphs proposed in [37] was shown to be successfully transferred to attack a real model with good results. Therefore, we hoped that this approach could also attack our social trust model. In general, we could attack an

undirected graph model with a linear model that removed the nonlinear activation function while preserving the idea of graph convolution:

$$f_{\mathcal{L}}(\bar{\mathcal{A}}) = \sigma(\bar{\mathcal{A}}\bar{\mathcal{A}}XW_{H_1}W_{H_2}) = \sigma(\bar{\mathcal{A}}^2XW_{\mathcal{C}}) \quad (8)$$

In Equation (8), W_{H_1} denotes the first layer of the surrogate model's learnable parameter matrix, and W_{H_2} denotes the second layer of the learnable parameter matrix. In the training phase of the surrogate model, $W_{H_1} \times W_{H_2}$ can be simplified to a single matrix $W_{\mathcal{C}}$. $\bar{\mathcal{A}} = \bar{\mathcal{D}}^{-\frac{1}{2}}\bar{\mathcal{A}}\bar{\mathcal{D}}^{-\frac{1}{2}}$, where $\bar{\mathcal{A}} = \mathcal{A} + \mathcal{I}_N$, and $\bar{\mathcal{D}}_{ii} = \sum_j \bar{\mathcal{A}}_{ij}$, $\sigma(\cdot)$ is a simple linear activation function. $\mathcal{A}$ represents the adjacency matrix of the social network dataset, while $\mathcal{I}_N$ is a diagonal matrix with all entries set to 1. The $\bar{\mathcal{A}}$ matrix is created by adding together the $\mathcal{A}$ and $\mathcal{I}_N$ matrices, thereby incorporating the structural information of the nodes into the training process. The $\bar{\mathcal{A}}$ matrix is a symmetric normalized Laplacian matrix derived from a fast approximate convolution [11] of the graph and has since become widely used in graph convolutional neural networks. The X matrix is the feature matrix containing the features of the graph.

However, this approach cannot be applied to the network, where users may assume two distinct roles, trustor and trustee, due to the asymmetric nature of social networks. The social trust voting given by a user and the social trust voting received by a user may differ. It is not enough to consider only the existence of reciprocal social relationships between two users when attacking. We propose a two-way attack method that classifies the social relationships and social trust voting in social trust networks according to what we call input interaction and output interaction.

To tackle this asymmetry property of social trust networks, we split a social network graph $G = (F, R, V)$ into an input social trust network graph $G_i = (F, R_i, V_i)$ and an output social trust network graph $G_o = (F, R_o, V_o)$, where

$$R_i = \left\{ r_{ij}, \text{for } i \leq j \right\}, R_o = \left\{ r_{ij}, \text{for } i \geq j \right\} \quad (9)$$

$$V_i = \left\{ v_{ij}, \text{for } i \leq j \right\}, V_o = \left\{ v_{ij}, \text{for } i \geq j \right\} \quad (10)$$

To ensure the symmetry of the adjacency matrix, we added the transpose matrix to the original matrix:

$$R_i = R_i + R_i^T, R_o = R_o + R_o^T \quad (11)$$

$$V_i = V_i + V_i^T, V_o = V_o + V_o^T \quad (12)$$

2.3. Attack Evaluation Module

Each social relationship R corresponds to a social trust vote V . We previously filtered out the social relationships R that were not easily detectable during the attack. From this, we could attack asymmetric social trust networks through a linear approach. Algorithm 1 shows the pseudo-code for generating the poisoning samples. Specifically, we used a local optimal solution approach to generate the best samples without exceeding the limits by ranking all samples according to their scores for the model performance in Equation (13), where $\oplus$ denotes the concatenation operation between two vectors.

$$\mathcal{C}_{Attack} = \max_{c_{\epsilon, \Lambda}^R} [\mathcal{S}(f_{\mathcal{L}}(V_i)) \oplus \mathcal{S}(f_{\mathcal{L}}(V_o))] \quad (13)$$

The goal of our task was to maximize the difference between the pre-poisoning and post-poisoning model accuracies; therefore, we could define the scoring function for the poisoning method given the original sample G and the parameter matrix $W_{\mathcal{C}}$ as

$$\mathcal{S}[G, W_{\mathcal{C}}] = \left(f_{\mathcal{L}}(\bar{V}^2 X W_{\mathcal{C}})_{\alpha \rightarrow \beta, v} - f_{\mathcal{L}}(\bar{V}^2 X W_{\mathcal{C}})_{\alpha \rightarrow \beta, v_{\alpha \rightarrow \beta}} \right) \quad (14)$$

The above is done to select and generate the poisoning samples. Given a social trust network graph G , a candidate set is formed after confirming the evaluation of social relations that meet the requirements $\mathcal{C}_{\varepsilon, \Lambda}^R$. The social relationship in the candidate set can be changed. The social trust voting that has the greatest impact on the performance of the model is selected from the candidate set by the scoring function, and it is selected as the final poisoning sample C_{Attack} .

Algorithm 1 An algorithm to find the best poisoning sample under constrained conditions.

Output: Best poisoning samples C_{Attack}

Input: Social trust network $G = (F, R, V)$, poisoning budget ε

- 1: Use linear function on G to get $W_{Combine}$
 - 2: **while** $\sum[|R' - R| + |V' - V|] \leq \varepsilon$ **do**
 - 3: $C \leftarrow$ samples under constraint
 - 4: $\mathcal{S}_{Combine} \leftarrow \mathcal{S}(f_{\mathcal{L}}(V_i)) \oplus \mathcal{S}(f_{\mathcal{L}}(V_o))$
 - 5: $C_{Attack} \leftarrow \max_{\mathcal{C}_{\varepsilon, \Lambda}^R} \mathcal{S}_{Combine}$
 - 6: **end while**
-

3. Experimental Results

3.1. Dataset Description

Because of the destructive nature of the attack, it was not possible to validate the effectiveness of the attack by attacking an ongoing social platform. Therefore, we chose three widely used real-world datasets to validate the effectiveness of our data-poisoning attacks.

The initial dataset was Advogato [38], which is a network community site for free software development workers, where all users can express their trust level with others, which offers different trust levels: {Observer, Apprentice, Journeyer, Master}.

Another network dataset is Pretty-Good-Privacy (PGP) [39], which is a cryptographic application that applies ideas related to social trust networks, which provides secure encryption services and authentication services for data transmission, and similar to the initial one, contains different voting levels of trust in the PGP dataset.

The third dataset is BitcoinOTC [40]. This network represents the trust relationships among individuals who engage in Bitcoin trading on the BitcoinOTC platform. Since the Bitcoin platform is an anonymous platform where the identities of all users are hidden, in order to prevent fraudulent transactions and detect malicious users to prevent them from trading with normal users, someone created the BitcoinOTC platform, where users can rate each other's trust on a scale from -10 (distrust) to 10 (trust). To facilitate comparison with other datasets, we defined users with ratings between -10 and -5 as Observer, users between -5 and 0 as Apprentice, users between 0 and 5 as Journeyer, and users between 5 and 10 as Master. Statistics for these three datasets are listed in the Table 2.

Table 2. Statistical description of three datasets.

	Advogato	PGP	BitcoinOTC
# of nodes	5.2 K	10.7 K	5.8 K
# of edges	47.1 K	24.3 K	35.5 K
Density	0.003548	0.000426	0.002490
Maximum degree	941	205	1298
Minimum degree	1	1	1
Average degree	18	4	12

3.2. Experimental Settings

All of our experiments were run on a machine with an Intel Core i7-11700 CPU, 64 GB RAM, 500 GB SSD, and GeForce RTX 3060 GPU.

To showcase the effects of the attack, we employed Guardian [13] as our attacked model with its default configuration in Table 3. The model used node2vec as the graph-

embedding method, providing an initial embedding of 128 dimensions for each user, and three graph convolutional layers with [32, 64, 32] layers each. For the hyperparameters, we configured the learning rate to 0.01, the dropout to 0.0, and the normalization factor to 10^{-5} . We assessed the effects of the attack after each model's training session, which consisted of 200 epochs. We randomly divided each dataset into two parts, where 80% of the votes formed the training dataset, and the remaining 20% served as the test set of the model, and the test set was removed during the training period. One-hot encoding was used in the training phase. In the dataset, four different types of trust voting were included, and each trust vote was transformed into the following representation:

$$\{[0, 0, 0, 1]^T, [0, 0, 1, 0]^T, [0, 1, 0, 0]^T, [1, 0, 0, 0]^T\} \quad (15)$$

Table 3. The configuration of the model.

Parameters	Settings
Embedding_dim	128
Learning rate	0.01
Dropout rate	0
Normalization factor	10^{-5}
Layer numbers	[32, 64, 32]

3.3. Experimental Results and Analysis

3.3.1. Effectiveness of the Avoidance Detection Module

Since related studies found that the propagation speed of information tends to follow a power-law distribution in the networks, this means that a few pieces of information will quickly spread to a large number of users. By studying the power-law distribution, it can be found that a few individual users with high influence play a key role in the social trust network, and their social trust voting may have a profound impact on the whole network.

Considering the substantial volume of data within social trust networks, changes in the network structure are difficult to detect directly. We demonstrated the importance of restricting $c_{\varepsilon, \Delta}^R$ by comparing the frequency of node degree distributions. The frequency of the node degree distribution can be examined to visually detect whether the social trust network has been tampered with. Figure 4 shows the changes in the node degree distribution frequency of the Advogato dataset under different attacks. The closer the poisoned sample is to the original network, the more it can avoid detection. If not restricted, social trust networks will become increasingly different from the original network after being poisoned. This illustrates that the restrictions we set on changing social relationships make the changes more imperceptible.

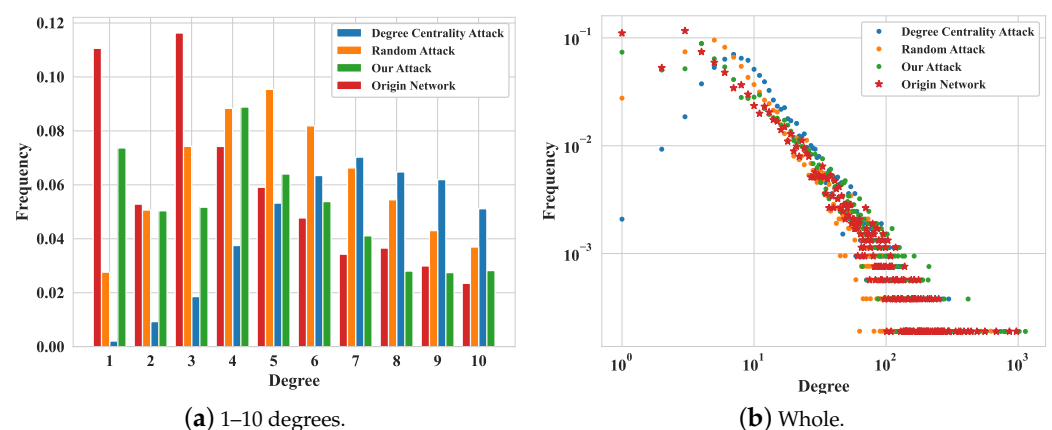


Figure 4. The node degree distribution of the Advogato dataset.

3.3.2. Impact of Attacks on Model Performance

To comprehensively assess the efficacy of our poisoning approach, we evaluated the model's accuracy in three real-world datasets, and to avoid unbalanced samples, we also evaluated the F1-weighted score.

Accuracy, which is widely used to evaluate the performance of neural network models, is an intuitively simple representation of the number of samples in a model whose predictions match the real labels as a proportion of the total number of samples.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (16)$$

TP (true positive) and TN (true negative) denote the count of samples accurately predicted by the model, while FP (false positive) and FN (false negative) represent the count of samples incorrectly predicted by the model.

The F1-weight score comprehensively evaluates the accuracy, recall, and generalization ability of the model. This evaluation metric indicates the combined ability of the model to predict different samples in the model prediction process.

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (17)$$

We first implemented the attacked models in the three original datasets Advogato, PGP, and BitcoinOTC, with accuracies of 73%, 87.3%, and 89.8%, and F1-weighted scores of 72.9%, 87.3%, and 88.9%, respectively.

Given that there is no baseline for such attacks against GNN-based social trust models, to validate the efficacy of our attacks, we compared our proposed method with two widely used attack methods for GNN attacks, which are random-poisoning attacks and degree-centrality-poisoning attacks.

A random-poisoning attack selects a random set of users in a social dataset and gives random users random social trust ratings. It continuously picks users until a set number of poisonings is reached. This method is very simple, but in the subsequent experiments, we showed that the social trust model was unable to detect this kind of attack, and it also had an impact on the performance of the social trust model.

The degree of a node in a graph refers to the count of the connected edges of that node. In a social trust network, the higher the degree in the graph's node, the greater its degree centrality, signifying its increased significance within the network, which means that this user is more important to the whole network. Degree-centrality attacks are performed on nodes in the social trust network that have a larger degree of nodes. The attack will create a social relationship between two nodes of greater degree and give random social trust voting until the poisoning ratio is reached.

$$Degree = \frac{N_{degree}}{n - 1} \quad (18)$$

where n denotes the number of nodes, and N_{degree} denotes the degree of the node.

As depicted in Figure 5, we established a compact social network in Section 2. In our analysis, the random attack randomly targets any of the social relationships within the network. Conversely, the degree-centrality attack prioritizes the targeting of edges associated with nodes possessing higher degrees.

The baseline and the performance after using different attack methods and different poisoning ratios of {5%, 10%, 15%, 20%} are shown in Figures 6 and 7. Figure 6 shows the accuracy of these datasets for different attack methods and different poisoning ratios, and Figure 7 shows the F1-weighted score.

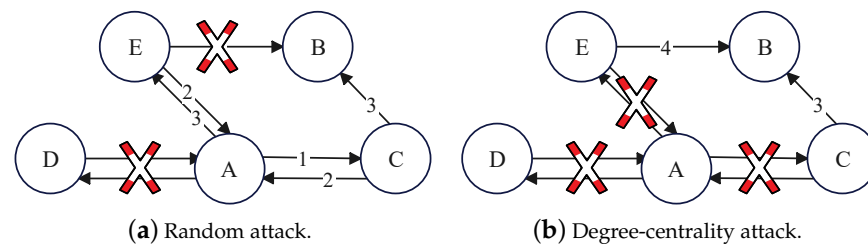


Figure 5. The illustrations of random and degree-centrality attacks.

As shown, our attack method was valid for any percentage of poisoned samples. The model's accuracy and F1-weighted score dropped as the percentage of poisoning increased, but our proposed method consistently led among all these poisoning methods.

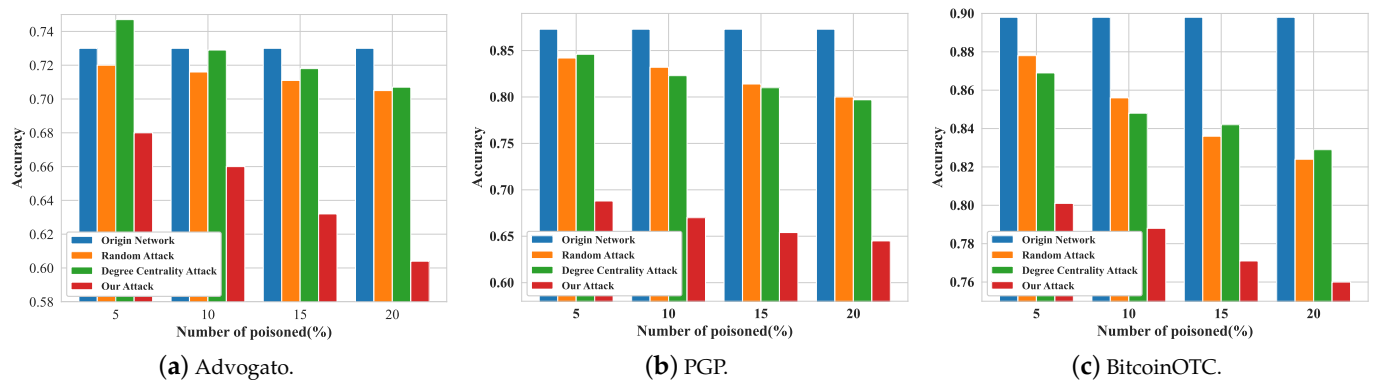


Figure 6. Accuracy under different attacks and poisoning ratios.

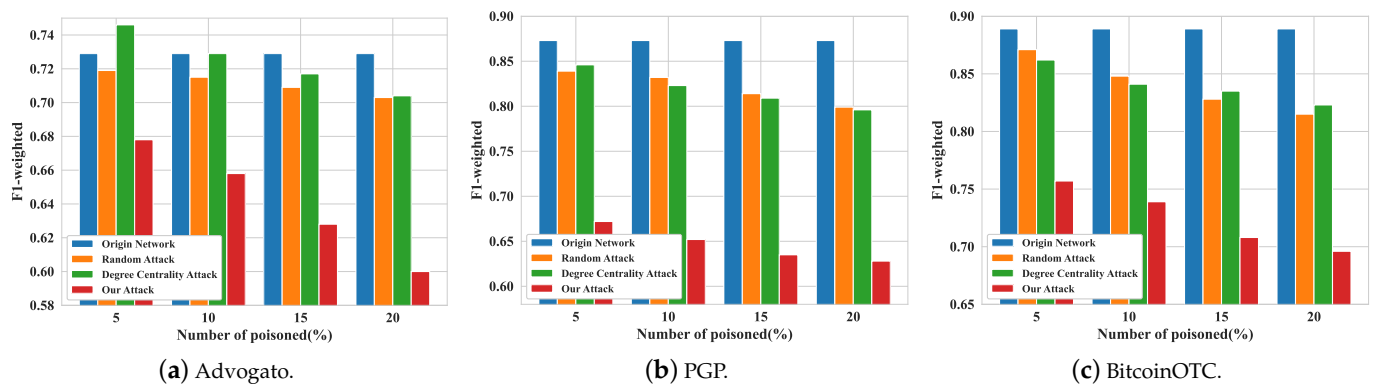


Figure 7. F1-weighted score under different attacks and poisoning ratios.

As with the other attack methods, the impact of our method on the accuracy and F1-weighted score gradually increased with the ratios of poisoning attacks. Our method's effect on the model's accuracy and F1-weighted score surpassed those of the other attack methods, namely, the random attack and degree-centrality attack, across all ratios of poisoning attack samples.

To illustrate the effect of our attack in more detail, we elaborate on the effect of our attack on the attacked model in terms of the accuracy and F1-weighted score in Tables 4 and 5. Our attack method decreased the model accuracy by 12.6%, 22.8%, and 13.8%, and decreased the F1-weighted score by 12.9%, 24.5%, and 19.3% at the 20% poisoning ratio, respectively.

Table 4. Accuracy of three datasets under different poisoning methods and ratios.

Origin Network		Advogato 0.73			
Poisoning ratio	5%	10%	15%	20%	
Random attack	0.720	0.716	0.711	0.705	
Degree-centrality attack	0.747	0.729	0.718	0.705	
Our attack	0.680	0.660	0.632	0.604	
Origin Network		PGP 0.873			
Poisoning ratio	5%	10%	15%	20%	
Random attack	0.842	0.832	0.814	0.8	
Degree-centrality attack	0.846	0.823	0.81	0.797	
Our attack	0.688	0.67	0.654	0.645	
Origin Network		BitcoinOTC 0.898			
Poisoning ratio	5%	10%	15%	20%	
Random attack	0.878	0.856	0.836	0.824	
Degree-centrality attack	0.869	0.848	0.842	0.829	
Our attack	0.801	0.788	0.771	0.76	

Table 5. F1-weighted score of three datasets under different poisoning methods and ratios.

Origin Network		Advogato 0.729			
Poisoning ratio	5%	10%	15%	20%	
Random attack	0.719	0.715	0.709	0.703	
Degree-centrality attack	0.746	0.729	0.717	0.704	
Our attack	0.678	0.658	0.628	0.6	
Origin Network		PGP 0.873			
Poisoning ratio	5%	10%	15%	20%	
Random attack	0.839	0.832	0.814	0.799	
Degree-centrality attack	0.846	0.823	0.809	0.796	
Our attack	0.672	0.652	0.635	0.628	
Origin Network		BitcoinOTC 0.889			
Poisoning ratio	5%	10%	15%	20%	
Random attack	0.871	0.848	0.828	0.815	
Degree-centrality attack	0.862	0.841	0.835	0.823	
Our attack	0.757	0.739	0.708	0.696	

3.3.3. Impact of Attacks on Model Training

The loss function plays a vital role in the training of models. The loss function serves to quantify the disparity between the model's predictions and the actual labels. By optimizing the loss function, the model can better align with the data and enhance its performance. By contrasting the training loss of the model before and after the data-poisoning attack, we could discern the attack's influence on the model training.

We also compare the loss functions in Figures 8 and 9. The model was tuned to the training parameters by the loss function during the training process. The inability to decrease the loss function will lead to an increase in the training costs and the consumption of computational resources.

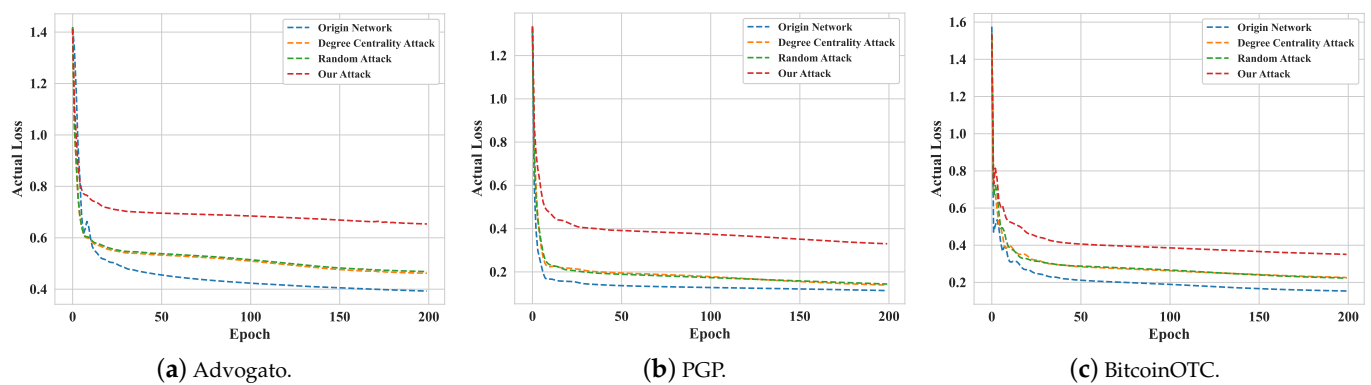


Figure 8. The actual loss under different attacks at a 20% poisoning ratio.

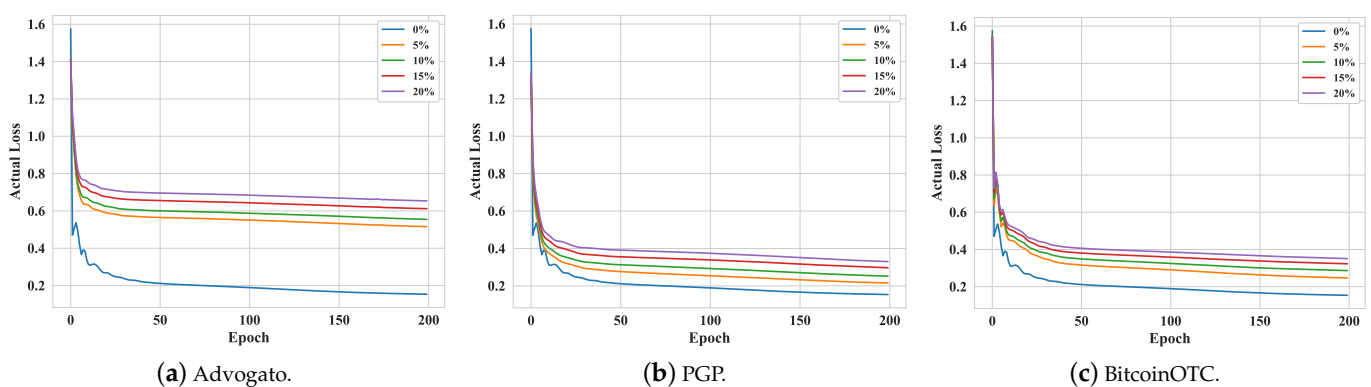


Figure 9. The actual loss under our attack at different poisoning ratios.

4. Conclusions

In this paper, we introduce a data-poisoning attack method in a gray-box environment tailored for GNN-based social trust models, leveraging insights from the characteristics of social trust networks. Our approach prioritizes preserving the integrity of influential social relationships by maintaining the node degree distribution through a two-sample test for power-law distributions. Through extensive experimentation on three datasets, we showcased the efficacy of our attack method in compromising the performance of social trust models.

Examining the robustness of social trust models stands as a critical endeavor, and our study contributes essential insights for crafting more resilient models.

In future endeavors, we aim to enhance the efficiency of attacks against social trust models while concurrently designing robust models and effective attack detection mechanisms to bolster the security and reliability of social trust networks. We aspire to tackle more challenging attack scenarios, such as black-box environments, which necessitate greater computational resources and query accesses to the attacked model, thereby intensifying the difficulty of the attack and making it more susceptible to detection by the victim.

It is crucial to emphasize that attacks are not the ultimate goal of security. Rather, the objective is to bolster the robustness of social trust models through attacks and cultivate more secure social trust networks.

Author Contributions: Conceptualization, N.J.; methodology, J.Z. and N.J.; validation, J.Z.; data curation, K.P.; software, H.Z.; visualization, J.W.; writing—original draft, J.Z.; writing—review and editing, J.Z. and Z.T.; supervision, N.J.; project administration, J.W.; funding acquisition, N.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the National Natural Science Foundation of China under grant 62172160 and grant 62062034, in part by the Jiangxi Provincial Natural Science Foundation under grant 20212ACB212002, and in part by the Excellent Scientific and Technological Innovation Teams of Jiangxi Province under grant 20181BCB24009.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author due to privacy reasons.

Acknowledgments: We are very grateful to all the editors and reviewers for their helpful suggestions, which greatly improved the presentation of this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yu, H.; Gibbons, P.B.; Kaminsky, M.; Xiao, F. SybilLimit: A Near-Optimal Social Network Defense against Sybil Attacks. In Proceedings of the 29th IEEE Symposium on Security and Privacy, Oakland, CA, USA, 18–21 May 2008; pp. 3–17.
2. Yu, H.; Kaminsky, M.; Gibbons, P.B.; Flaxman, A.D. SybilGuard: Defending Against Sybil Attacks via Social Networks. *IEEE/ACM Trans. Netw.* **2008**, *16*, 576–589. [\[CrossRef\]](#)
3. Samreen, S.; Jabbar, M. Countermeasures for Conflicting Behavior Attack in a Trust Management Framework for a Mobile Ad hoc Network. In Proceedings of the IEEE International Conference on Computational Intelligence and Computing Research, Coimbatore, India, 14–16 December 2017; pp. 1–4.
4. Sun, Y.L.; Han, Z.; Yu, W.; Ray Liu, K.J. Attacks on Trust Evaluation in Distributed Networks. In Proceedings of the 40th IEEE Annual Conference on Information Sciences and Systems, Princeton, NJ, USA, 22–24 March 2006; pp. 1461–1466.
5. Xue, L.; Wang, Q.; An, L.; He, Z.; Feng, S.; Zhu, J. A nonparametric adaptive EWMA control chart for monitoring mixed continuous and categorical data using self-starting strategy. *Comput. Ind. Eng.* **2024**, *188*, 109930. [\[CrossRef\]](#)
6. Yeganeh, A.; Shadman, A.; Shongwe, S.C.; Abbasi, S.A. Employing evolutionary artificial neural network in risk-adjusted monitoring of surgical performance. *Neural Comput. Appl.* **2023**, *35*, 10677–10693. [\[CrossRef\]](#)
7. Salmasnia, A.; Namdar, M.; Abolfathi, M.; Ajaly, P. Statistical design of a VSI-EWMA control chart for monitoring the communications among individuals in a weighted social network. *Int. J. Syst. Assur. Eng. Manag.* **2021**, *12*, 495–508. [\[CrossRef\]](#)
8. Flossdorf, J.; Fried, R.; Jentsch, C. Online monitoring of dynamic networks using flexible multivariate control charts. *Soc. Netw. Anal. Min.* **2023**, *13*, 87. [\[CrossRef\]](#)
9. Noorossana, R.; Hosseini, S.S.; Heydarzade, A. An overview of dynamic anomaly detection in social networks via control charts. *Qual. Reliab. Eng. Int.* **2018**, *34*, 641–648. [\[CrossRef\]](#)
10. Hamilton, W.L.; Ying, R.; Leskovec, J. Inductive Representation Learning on Large Graphs. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 1025–1035.
11. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In Proceedings of the 5th International Conference on Learning Representations, Toulon, France, 24–26 April 2017; pp. 1–14.
12. Zhang, M.; Chen, Y. Link Prediction Based on Graph Neural Networks. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; pp. 5171–5181.
13. Lin, W.; Gao, Z.; Li, B. Guardian: Evaluating Trust in Online Social Networks with Graph Convolutional Networks. In Proceedings of the 39th IEEE Conference on Computer Communications, Toronto, ON, Canada, 6–9 July 2020; pp. 914–923.
14. Zügner, D.; Borchert, O.; Akbarnejad, A.; Günnemann, S. Adversarial attacks on graph neural networks: Perturbations and their patterns. *ACM Trans. Knowl. Discov. Data* **2020**, *14*, 1–31. [\[CrossRef\]](#)
15. Jin, W.; Li, Y.; Xu, H.; Wang, Y.; Ji, S.; Aggarwal, C.; Tang, J. Adversarial Attacks and Defenses on Graphs. *SIGKDD Explor. Newsl.* **2021**, *22*, 19–34. [\[CrossRef\]](#)
16. Chen, J.; Wu, Y.; Xu, X.; Chen, Y.; Zheng, H.; Xuan, Q. Fast gradient attack on network embedding. *arXiv* **2018**, arXiv:1809.02797.
17. Wang, X.; Cheng, M.; Eaton, J.; Hsieh, C.J.; Wu, F. Attack graph convolutional networks by adding fake nodes. *arXiv* **2018**, arXiv:1810.10751.
18. Chen, J.; Chen, Y.; Zheng, H.; Shen, S.; Yu, S.; Zhang, D.; Xuan, Q. MGA: Momentum gradient attack on network. *IEEE Trans. Comput. Soc. Syst.* **2020**, *8*, 99–109. [\[CrossRef\]](#)
19. Chen, J.; Zhang, J.; Chen, Z.; Du, M.; Xuan, Q. Time-aware gradient attack on dynamic network link prediction. *IEEE Trans. Knowl. Data Eng.* **2021**, *35*, 2091–2102. [\[CrossRef\]](#)
20. Sharma, A.K.; Kukreja, R.; Kharbanda, M.; Chakraborty, T. Node injection for class-specific network poisoning. *Neural Netw.* **2023**, *166*, 236–247. [\[CrossRef\]](#)
21. Zong, W.; Chow, Y.; Susilo, W.; Do, K.; Venkatesh, S. TrojanModel: A Practical Trojan Attack against Automatic Speech Recognition Systems. In Proceedings of the 44th IEEE Symposium on Security and Privacy, San Francisco, CA, USA, 21–25 May 2023; pp. 1667–1683.
22. Nelson, B.; Barreno, M.; Chi, F.J.; Joseph, A.D.; Rubinstein, B.I.P.; Saini, U.; Sutton, C.; Tygar, J.D.; Xia, K. Exploiting Machine Learning to Subvert Your Spam Filter. In Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats, San Francisco, CA, USA, 15 April 2008; pp. 1–9.

23. Li, B.; Wang, Y.; Singh, A.; Vorobeychik, Y. Data Poisoning Attacks on Factorization-Based Collaborative Filtering. In Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 1893–1901.
24. Fang, M.; Yang, G.; Gong, N.Z.; Liu, J. Poisoning Attacks to Graph-Based Recommender Systems. In Proceedings of the 34th ACM Annual Computer Security Applications Conference, San Juan, PR, USA, 3–7 December 2018; pp. 381–392.
25. Fang, M.; Gong, N.Z.; Liu, J. Influence Function Based Data Poisoning Attacks to Top-N Recommender Systems. In Proceedings of the Web Conference, New York, NY, USA, 20–24 April 2020; pp. 3019–3025.
26. Shan, S.; Wenger, E.; Zhang, J.; Li, H.; Zheng, H.; Zhao, B.Y. Fawkes: Protecting Privacy against Unauthorized Deep Learning Models. In Proceedings of the 29th USENIX Security Symposium, Boston, MA, USA, 12–14 August 2020; pp. 1589–1604.
27. Jiang, Y.; Xia, H. Adversarial attacks against dynamic graph neural networks via node injection. *High-Confid. Comput.* **2024**, *4*, 100185. [[CrossRef](#)]
28. Douceur, J.R. The sybil attack. In Proceedings of the International Workshop on Peer-to-Peer Systems, Cambridge, MA, USA, 7–8 March 2002; Springer: Berlin/Heidelberg, Germany, 2002; pp. 251–260.
29. Gürses, S.; Berendt, B. The social web and privacy: Practices, reciprocity and conflict detection in social networks. In *Privacy-Aware Knowledge Discovery, Novel Applications and New Techniques*; CRC Press: Boca Raton, FL, USA, 2010; pp. 395–429.
30. Costa, H.; Merschmann, L.H.; Barth, F.; Benevenuto, F. Pollution, bad-mouthing, and local marketing: The underground of location-based social networks. *Inf. Sci.* **2014**, *279*, 123–137. [[CrossRef](#)]
31. Sony, S.M.; Sasi, S.B. On-Off attack management based on trust. In Proceedings of the 2016 Online International Conference on Green Engineering and Technologies (IC-GET), Coimbatore, India, 19 November 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1–4.
32. Li, S.; Yang, J.; Liang, G.; Li, T.; Zhao, K. SybilFlyover: Heterogeneous graph-based fake account detection model on social networks. *Knowl.-Based Syst.* **2022**, *258*, 110038. [[CrossRef](#)]
33. Adekunle, T.S.; Alabi, O.O.; Lawrence, M.O.; Ebong, G.N.; Ajiboye, G.O.; Bamisaye, T.A. The Use of AI to Analyze Social Media Attacks for Predictive Analytics. *J. Comput. Theor. Appl.* **2024**, *2*, 169–178.
34. Bessi, A. Two samples test for discrete power-law distributions. *arXiv* **2015**, arXiv:1503.00643.
35. Clauset, A.; Shalizi, C.R.; Newman, M.E.J. Power-Law Distributions in Empirical Data. *SIAM Rev.* **2009**, *51*, 661–703. [[CrossRef](#)]
36. Sherchan, W.; Nepal, S.; Paris, C. A Survey of Trust in Social Networks. *ACM Comput. Surv.* **2013**, *45*, 47:1–47:33. [[CrossRef](#)]
37. Zügner, D.; Akbarnejad, A.; Günnemann, S. Adversarial Attacks on Neural Networks for Graph Data. In Proceedings of the 24th ACM International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 2847–2856.
38. Massa, P.; Salvetti, M.; Tomasoni, D. Bowling Alone and Trust Decline in Social Network Sites. In Proceedings of the 8th IEEE International Conference on Dependable, Autonomic and Secure Computing, Chengdu, China, 12–14 December 2009; pp. 658–663.
39. Boguñá, M.; Pastor-Satorras, R.; Díaz-Guilera, A.; Arenas, A. Models of social networks based on social distance attachment. *Phys. Rev. E* **2004**, *70*, 056122. [[CrossRef](#)] [[PubMed](#)]
40. Kumar, S.; Hooi, B.; Makhija, D.; Kumar, M.; Faloutsos, C.; Subrahmanian, V. Rev2: Fraudulent user prediction in rating platforms. In Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, Marina Del Rey, CA, USA, 5–9 February 2018; pp. 333–341.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.