MRFS: Mining Rating Fraud Subgraph in Bipartite Graph for Users and Products

Wei Yu, Wenkai Wang, Guangquan Xu, *Member, IEEE*, Huaming Wu, *Senior Member, IEEE*, Hongyan Li, Jun Wang, Xiaoming Li, and Juan Liu

***Abstract*— Fraud in e-commerce fields (e.g., Amazon, Taobao, and so on) and social networks (e.g., Twitter and Weibo) has recently brought a very bad user experience. Rating fraud detection is an urgent issue for improving user experiences. However, existing methods have lots of limitations in some respects, because it is always very hard to acquire sufficient labeled data for fraud detection and detect new fraud patterns. Fortunately, the relationship for users rating (e.g., purchasing and following) products can be represented as a bipartite graph. So the problem of rating fraud detection can be transformed into the problem of abnormal subgraph detection in the bipartite graph. The major challenge of fraud detection is to distinguish fake rates from real user rates. In this article, we focus on mining rating fraud-connected subgraphs in a bipartite graph. The motivation for this work is fraud detection tasks, which can usually be formulated as mining a bipartite graph formed by source nodes (followers and users) and target nodes (followees and products) for malicious patterns. Now, smart fraudsters evade existing detection methods by buying a large pool of users and hijacking honest users, making them look "normal"-this behavior is called "*camouflage*." Accordingly, we propose a fraud detection approach for mining rating fraud subgraph (MRFS), which addresses the problem from the intrinsic metric (e.g., fraudulence, badness and unreliability). The proposed MRFS mines the intrinsic characteristics of nodes and edges from node behavior information, which is an effective and scalable (linear on the input size) algorithm. A large number of comparative experimental results on real-world rating networks show that our proposed MRFS is efficient and universal.**

***Index Terms*— Bipartite graph, e-commerce, fraud detection, graph mining, time series.**

NOMENCLATURE

$\mathcal{U} = \{u_i\}$	Users.
$\mathcal{V} = \{v_j\}$	Items.
$\mathcal{E}$	Edges.
$\mathcal{N}$	Nodes of bipartite network: $\mathcal{U} \cup \mathcal{V}$.
$\mathcal{A}$	Subset of users.
$\mathcal{B}$	Subset of items.
$\mathcal{S}$	Subset of nodes: $\mathcal{S} = \mathcal{A} \cup \mathcal{B}$.
φ_v	Involvement ratio.
$\hat{u}, \hat{v}$	User and item discrete probability distribution.
C_u, C_v	Global discrete probability distribution.
$\mathrm{KL}(\cdot\|\cdot)$	KL-divergence.
$g(\mathcal{S})$	Density metric.
$f(\mathcal{S})$	Total suspiciousness metric.
$\mathrm{score}(u, v)$	Rating score.
$\mathrm{Out}(u)$	Set of edges given by user u.
$\mathrm{In}(v)$	Set of edges received by items v.
$\lvert\mathrm{Out}(u)\rvert$, $\lvert\mathrm{In}(v)\rvert$	Out-degree of users, in-degree of items.
$F(u)$	User's intrinsic metric.
$G(v)$	Item's intrinsic metric.
$R(u, v)$	Edge's intrinsic metric.

Manuscript received 5 January 2022; revised 22 June 2022 and 25 December 2022; accepted 30 December 2022. Date of publication 9 January 2023; date of current version 31 May 2024. This work was supported by the Research Project of Tianjin Education Commission under Grant 2018KJ062. *(Corresponding authors: Wenkai Wang; Huaming Wu.)*

Wei Yu, Hongyan Li, and Xiaoming Li are with the School of International Business, Zhejiang Yuexiu University, Shaoxing 312069, China (e-mail: weiyu@tju.edu.cn; hyli@zyufl.edu.cn; lxm696@tju.edu.cn).

Wenkai Wang and Jun Wang are with the College of Intelligence and Computing, Tianjin University, Tianjin 300350, China (e-mail: wkwang@tju.edu.cn; jun.wang@tju.edu.cn).

Guangquan Xu is with the Tianjin Key Laboratory of Advanced Networking (TANK), College of Intelligence and Computing, Tianjin University, Tianjin 300350, China (e-mail: losin@tju.edu.cn).

Huaming Wu is with the Center for Applied Mathematics, Tianjin University, Tianjin 300072, China (e-mail: whming@tju.edu.cn).

Juan Liu is with the Stomatological Hospital, Tianjin Medical University, Tianjin 300070, China (e-mail: ljljmouse611@163.com).

Digital Object Identifier 10.1109/TCSS.2022.3233821

I. INTRODUCTION

IT IS well known that fraud causes great damage to the business of the web online applications, such as social networks. Most online services depend on hybrid recommendation models to recommend relevant information to users, which are similar to some recommendation mechanisms of web application program interfaces (APIs) [1], [2]. So it is crucial for their performance that user feedback of true interests is legitimate and indicative. Research shows that more than 1/3 of consumers regularly check ratings and comments before choosing to shop online. The growing importance of such virtual approval for driving sales seems to have led to a rather unethical business practice. Instead of waiting for users to approve of your products and services, why not just buy a lot of "likes" and demonstrate your popularity right away? To do this, businesses employ the services of a "click farm" to boost their popularity. For instance, on a social network or media sharing website, users always want to increase the popularity of their accounts by illegally buying many more "likes" [3], [4]; on e-commerce websites, a merchant can make their products more popular through Amazon's fake reviews. Unfortunately, many online sites provide services that charge typically just a small amount of money per 1000 fake links. For example, *taobaojing.cn*, *buy1000followers.co*, *boostlikes.com*, and *buyamazonreviews.com* provide the services of Taobao

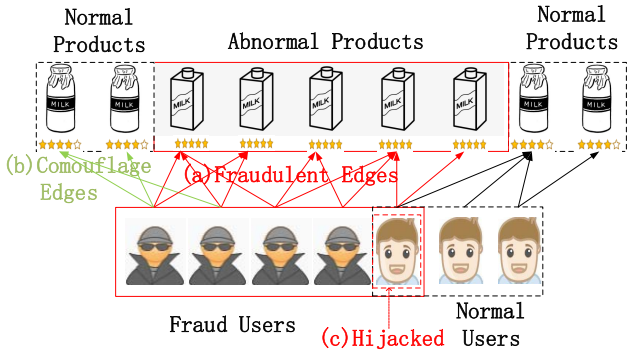


Fig. 1. Schematic of fraud detection problem in E-commerce field. (a) Synchronized behavior: fraudsters generate multiple links to fraudulent products in a short time. (b) Camouflage: fraud users create edges to normal products. (c) Hijacked: some fraudsters use hijacked accounts from normal users.

brushers, fake Twitter followers, Facebook page-likes, and Amazon product reviews, respectively.

In general, fraudsters always act in a very short time to increase the total impact of target items [5], [6]. Since online e-commerce purchasing behavior (online social behavior) actually reflects the relationships between users and products (followers and followees), it can be constructed into a bipartite graph, which is usually used in Web APIs Recommendation [7], [8]. As shown in Fig. 1, the top row represents the product nodes, the bottom row represents the user nodes, and the arrows represent relationships between users and products. Therefore, the problem of fraud detection can be regarded as detecting the suspicious communities in the attributed bipartite graph, of which the source nodes represent users (followers, customers), the target nodes represent items (followees, products), and the directed edges correspond to the interaction from a user to the item including rating and following. The attributes on each edge include a timestamp, rating score, review, and so on, of which the definitions are illustrated in Fig. 1. In detail, the target is to detect the area enclosed by the red solid line. The fraudster is more inclined to give the fraudulent products more five-star praise. Synchronized behavior means that fraudsters generate multiple links to fraudulent products in a short time, camouflage means that fraud users create edges to normal products, and hijacked means that some fraudsters use hijacked accounts from normal users. In a word, fraudsters accurately find the target population in different scenarios, and accurately grasp the victim's psychology, so as to carry out fraud. It makes fraud detection more difficult, and at the same time, the technicalization of fraud has brought fraud cost reduction and fraud threat expansion.

Fraud detection is a challenging problem in e-commerce transaction networks [14], [15]. Many works try to detect fraudulent users and fraudulent items. Previous methods are mainly based on two aspects: 1) some methods detect communities of a bipartite graph and 2) other methods look for an unusual connection structure. These methods directly mine bipartite graphs to obtain dense subgraphs or rare structural models. Such as belief propagation (BP) [16], [17], hyperlink-induced topic search (HITS) [5], [18], singular value decomposition (SVD) [3], [12], Fraudar [9] and changes to community structure [19], etc. However, these approaches have lots of limitations in some respects. It is always very hard to acquire sufficient labeled data for fraud detection because of the scale of the research problem and the cost of the investigation. In addition, they require a lot of manpower and material resources to carry out complex feature engineering. Thirdly, new fraud patterns always can't be detected.

Aiming at the above problems, an unsupervised method named mining rating fraud subgraph (MRFS) is proposed to detect abnormal users and items in this article. In addition, the corresponding optimization algorithm of MRFS is developed to compute the intrinsic metrics for all users, edges, and items by fusing network and behavior properties including time and ratings. Extensive experimental results on eight rating datasets show that MRFS is superior to the baseline methods in detecting fraudulent users and items.

To sum up, the main contributions of this article can be summarized as follows.

1) *Algorithm:* We use complex networks to explore the internal mechanism of fraud and combine network topology attributes and node attributes to solve the relevant problems faced by fraud detection. At the same time, we propose three intrinsic metrics, which emphasize the interaction between the source node (users) and target node (items) in the network in an unsupervised way. And fully considers the fraudulent user's scoring and concentrated attack (temporal bursts) behavior, making our algorithm more resistant to camouflage.
2) *Effectiveness:* A large number of comparative experimental results on semi-real and real datasets show that Our proposed MRFS is superior to the baselines. Our algorithm can maintain high precision (90%), before reaching 0.024 in fraudulent density. However, most of the existing methods have relatively low accuracy as far as we know, especially when the fraudulent density decreases.
3) *Scalability:* MRFS is scalable, with near-linear time complexity in the number of edges. It is easy to scale up applications to large-scale networks.

II. Related Work

Most fraudulent schemes are designed in order to obtain economic benefits. To maximize their financial gains, fraudsters must share various information (e.g., user IDs, IP addresses, and phone numbers). As a result, fraudsters naturally exhibit synchronized behavior on some features, maybe user IDs, or IP addresses. For example, fraudsters often use many fraudulent users for the same fraud [20], [21]. In fact, online fraudsters usually organize fraudulent activities in a short time period [10]. The existing works fall into two main categories: high-density subgraph-based methods and anomaly subgraph structures-based methods.

A. High-Density Subgraph-Based Methods

Due to the limited labels, most previous research studies fraud detection in an unsupervised way to discover

TABLE I
COMPARISON BETWEEN MRFS AND EXISTING METHODS

	Detect dense bloks	Scalability	Camous-resistant	ranks anomalies	uses network
Frauda[9]	✓	✓	✓	×	×
CopyCatch[10]	✓	×	✓	×	×
Fbox[3]	✓	×	×	×	×
HoloScope[11]	✓	✓	✓	×	×
SpokEn[12]	✓	×	✓	×	✓
Rev2[13]	×	×	✓	✓	✓
CatchSync[5]	×	×	✓	✓	×
MRFS	✓	✓	✓	✓	✓

the high-density communities formed by fraud groups [22]. Dense communities mining is effective for detecting fraudulent groups of users and items connected by a large number of links. Fraudar [9] is proposed to measure the suspiciousness of edges to discount popular items. HoloScope [11], as a network topology-based method, dynamically reweights items with the suspicious beliefs of users. Some researchers have also captured abnormal dense user blocks with SVD [3], [23]. CoreScope [24] is proposed to detect abnormally dense communities in which all nodes have a degree of at least k with shingling and K-core algorithms. However, fraudsters can easily evade detection by cutting down the synchronicity of their behaviors.

B. Anomaly Subgraph Structures-Based Methods

Accordingly, this type of method for fraud detection is often based on anomaly subgraph structures of fraud groups [25]. A large number of studies have shown that subgraph structures have a great influence on social networks [26], [27]. Ren et al. [28] propose an Ensemble-based Fraud DETection (ENSEMFDET) method to scale up promotional campaigns fraud detection in bipartite graphs [29] by decomposing the original problem into subproblems on small-sized subgraphs. EdgeCentric [30] studied a method based on the distribution of rating scores to detect the anomaly. BP [16], [17] and HITS [5], [18] intend to catch some specific link attributes, such as sentiments, to find something anomaly. SynchroTrap [20] works on the user similarity graph. However, it cannot detect fraudsters trying to hide. The fraudster can relatively easily manipulate the edge of the fraudulent user to hide this structural pattern.

C. Other Fraud Detection Methods

In addition to the above two methods, there are some other methods. Such as, many previous approaches focus on detecting fraud by checking contents [31], [32]. However, these methods are usually not robust. Even if fraudsters don't understand the detection program, they may try to pretend to be regular users as much as possible. Deep learning is usually used for anomaly detection [33], [34]. But the type is black-box methods and there is almost no explanation for the detected output. When there are enough labeled data, the classifiers for fraud detection can be modeled based on multikernel learning [35], support vector machines [36], and k-nearest neighbor [37] approaches.

Table I shows the comparative analysis of various approaches for fraud detection. Our MRFS approach: 1) has no extra labels; 2) mines the essence of the network; and 3) ranks anomalies.

III. METHOD

This section gives the overall introduction including the notations and definitions used throughout the article, an intuitive description of fraud, and describe our models.

A. Notations and Problem Definition

1) Notations: We consider a set of users $\mathcal{U} = \{u_i\}, i \in 1, 2, \ldots, m$, and items $\mathcal{V} = \{v_j\}, j \in 1, 2, \ldots, n$, connected according to a bipartite networks. At the same time, there should be some attributes on the edges, such as rating score, and timestamp. In Nomenclature, the notations and definitions of the symbols are given. The tasks of the proposed MRFS are can be summarized as follows.

1) *Detecting:* Suspicious subgraph (users and items), a ranking list with suspicious metric scores.
2) *Computing:* Intrinsic metrics for nodes and edges under the prior knowledge of suspiciousness from rating time and score.

Is a user honest? Is the item shoddy? Is the edge reliable? The purpose is to mine the intrinsic features of nodes and edges from the node behavior information. Here, we assume each user has intrinsic fraudulence $F(u)$, each item has an intrinsic badness $G(v)$, and each edge (u, v) has an intrinsic unreliability $R(u, v)$. Obviously, $F(u), G(v), R(u, v)$ are all interrelated.

2) Definition of Related Intrinsic Metric:

1) *The Fraudulence of Users:* Users vary in terms of their fraudulence which indicate whether their ratings are fair.

Clearly, honest user ratings on items are fair. As Frauda [9] suggested, suspicious items attract less attention from non-fraudulent users due to their low quality. They give the good products with high scores and the bad products with low scores. However, fraudulent users usually deviate from the rules above. For example, they may give high ratings to low-quality products. The distribution of ratings that lead to fraudsters is very different from the rating distribution of typical users, as observed by [30]. At the same time, fraudsters groups their attacks, and fraudsters generate multiple links to the item in a short period of time [38]. So, We use these behavioral characteristics to mine the user's essential metric scores $F(u)$, $\forall u \in \mathcal{U}$. The range of the metric is $[0, 1]$, where 0 represents a 100% honest user, while 1 represents a 100% dishonest user.

2) *The Badness of Items:* Items have an intrinsic badness $G(v)$, $\forall v \in \mathcal{V}$. Intuitively, a good product should get much more high positive ratings (star 5) from honest users, and a bad product should get much more high negative ratings (star 1). The range of the metric is $[0, 1]$, 0 denotes 100% good items, while 1 denotes 100% where bad items.

3) *The Unreliability of Edges:* Edges vary in terms of unreliability. Are the edges generated by normal users reliable? But personal opinions are different from most people and can also create unreliable edges. Similarly, the edge generated by fraudulent users must not be reliable. This may create links pointing to normal items to disguise fraud users as normal ones. Therefore, we use unreliability to indicate the edge suspiciousness that fraudsters create for target items. The reliability $R(u, v)$ of connection (u, v) ranges from 0 (a normal edge) to 1 (an abnormal edge) $\forall (u, v) \in \mathcal{E}$.

4) *The Suspiciousness of Networks:* Metric g is used to measure the suspiciousness of a density network. Existing dense block detection methods [39], [40] maximize the arithmetic or geometric average degree. Here we use arithmetic average degree (the difference between the two methods is explained in Holoscope [11]).

B. Description of MRFS

1) Formalization of Intrinsic Metric: Given the definition of these metrics, we now offer MRFS and our analysis of MRFS. Here, we propose a class of **Metrics** g that is used as suspiciousness metrics. We set $\mathcal{A} \subseteq \mathcal{U}$, $\mathcal{B} \subseteq \mathcal{V}$, $\mathcal{S} = \mathcal{A} \cup \mathcal{B}$, and $\mathcal{N} = \mathcal{U} \cup \mathcal{V}$. Our goal is to find a suspicious subgraph S, and to approximately **maximize** $g(\mathcal{S})$. We define a density metric $g(\mathcal{S})$ as [9], [11]

$$g(\mathcal{S}) = \frac{f(\mathcal{S})}{|\mathcal{S}|} = \frac{f(\mathcal{S})}{|\mathcal{A}| + |\mathcal{B}|} \tag{1}$$

where $f(\mathcal{S})$ denotes the total suspiciousness and can be structured as

$$f(\mathcal{S}) = \sum_{(u_i, v_j) \in \mathcal{E}} R(u_i, v_j) + \sum_{u_i \in \mathcal{A}} F(u_i) + \sum_{v_j \in \mathcal{B}} G(v_j). \tag{2}$$

Intuitively, the formula contains the suspiciousness of nodes and edges. The node suspiciousness is the sum of intrinsic metrics corresponding to the users and items in $\mathcal{S}$. The edges suspiciousness is a sum of intrinsic metrics corresponding to the edges in between the $\mathcal{S}$. $g(\mathcal{S})$ satisfies the three axioms (proof in Fraudar [9]), which is the observations of $\mathcal{S}$, and obeys a number of basic axioms as follows.

1) Keeping $|\mathcal{S}|$ fixed, we have that $f(\mathcal{S}) \uparrow \Rightarrow g(\mathcal{S}) \uparrow$.
2) Keeping $f(\mathcal{S})$ fixed, we have that $|\mathcal{S}| \uparrow \Rightarrow g(\mathcal{S}) \downarrow$.
3) Keeping $\rho_{\text{edge}}(\mathcal{S})$ fixed, we have that $\mathcal{S} \uparrow \Rightarrow g(\mathcal{S}) \uparrow$.

Here, the **edge density** $\rho_{\text{edge}}(\mathcal{S})$ is

$$\rho_{\text{edge}}(\mathcal{S}) = \frac{|\mathcal{E}|}{|\mathcal{S}|(|\mathcal{S}| - 1)}. \tag{3}$$

Thus, the problem of subgraph detection can be defined as **1) input**: the bipartite rating network and **2) find**: subgraph of $\mathcal{S}$ that maximizes $g(\mathcal{S})$. Next, we introduce $F(u)$, $G(v)$, $R(u, v)$, respectively. $|\text{Out}(u)|$ gives the cardinality of the edge-attribute value produced from u's neighboring (outgoing) edges. Similarly, $|\text{In}(v)|$ gives the cardinality of the edge-attribute value produced from v's neighboring (incoming) edges.

1) *The Fraudulence of Users:* Intuitively, the user's suspiciousness is mainly determined by his connection to the item. Therefore, we simply define the user's fraudulence score as the average reliability score of their rating

$$F(u) = \frac{\sum_{(u,v) \in \text{Out}(u)} R(u, v)}{|\text{Out}(u)|}. \tag{4}$$

However, this only considers the impact of the rating score and does not include the user's behavioral attributes such as the distribution of rating scores and the time attribute. We will add the distribution of rating scores and the time attributed to them later.

2) *The Badness of Items:* When an item receives rating scores with different reliability edges, more importance should be given to rating scores that have higher reliability edges obviously. Therefore, in order to estimate the badness of the product, we weigh the rating score according to the reliability edge, giving a higher weight to the reliable rating score, which is not important for the rating of the low-reliability edge

$$G(v) = \frac{\sum_{(u,v) \in \text{In}(v)} R(u, v) \cdot \text{score}(u, v)}{|\text{In}(v)|} \tag{5}$$

where 1–5 star corresponds to 1, 0.75, 0.5, 0.25 and 0, respectively. Note that this is a rescaled version of the traditional five-star rating score scale. The bigger the score, the more abnormal.

3) *The Unreliability of Edges:* The edge should be considered reliable if it is given by a generally honest user u, and its rating score is close to the badness value of item v. This deviation is measured as the normalized absolute difference, $|\text{score}(u, v) - G(v)|$

$$R(u, v) = \frac{F(u) + |\text{score}(u, v) - G(v)|}{2}. \tag{6}$$

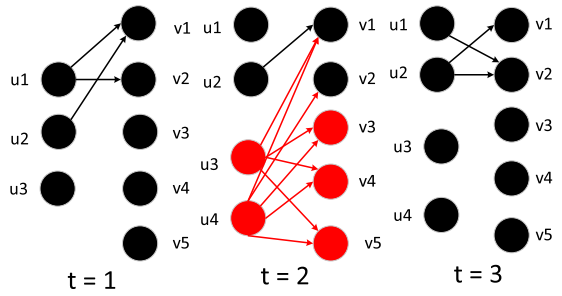


Fig. 2. Sudden appearance of a dense subgraph at $T = 2$.

2) Temporal Bursts and Score Distribution: Here, we aim at detecting the anomalies including the sudden appearance or disappearance of large dense directed subgraphs and the user's rating score distribution.

a) Temporal bursts: Usually, timestamps for edge creation are available in most real settings. We can analyze the time series, and the appearance or disappearance of a large dense subgraph is anomalous only in a very small timestamp. Similarly, the sudden appearance or disappearance of a large number of edges, which just form a dense subgraph, is anomalous. As shown in Fig. 2, user review item, an abnormal dense directed subgraph appears at $t = 2$. And users 3 and 4 launched an attack at time $t = 2$ (see the red subgraph in the figure). In contrast, the appearance of subgraph $\{u3, u4\} \rightarrow \{v3, v4, v5\}$ at $t = 3$ is not anomalous, since it has already been observed at $t = 1$. As shown in works [41] and [11], fraudulent users are bursty. They always give several ratings in a very short timespan. For example, fraudulent products receive multiple ratings and reviews in a short period of time when click farm receives a request. Thus we include the temporal attribute into $G(v)$. Let, φ_v represent the normality' score of items in multiple bursts. Or called the involvement ratio [11], it is a measure of a sudden attack on users in the subgraph. In order to model time series, we incorporate a state-of-the-art algorithm in our framework. A suspicious score $0 \leq \varphi_v \leq 1$ to each item v, higher (lower, resp.) score indicates more anomalous (normal, resp.).

b) Rating score distribution: We now consider edges with rating scores distribution. For each node u_i or v_i, we use the Kullback-Leibler (KL)-divergence to measure the loss between the distributions from the suspicious node and other nodes. The rating deviation K_u or K_v is scaled into [0, 1] by KL-divergence to compute suspiciousness. The abnormality scoring function K_u for user node $u \in \mathcal{U}$ is defined as

$$K_u = |\text{Out}(u)| \cdot \text{KL}(\hat{u} \| C_u), K_v = |\text{In}(v)| \cdot \text{KL}(\hat{v} \| C_v) \quad (7)$$

where $\hat{u}$ gives the discrete probability distribution associated with the user node over the chosen rating score and C_u gives the global discrete probability distribution of the rating score over all edges. Similarly, item K_v is defined as such.

So, the resulting equations are

$$F(u) = \frac{\sum_{(u,v)\in \text{Out}(u)} R(u,v) + \alpha_1 \cdot K_u}{|\text{Out}(u)| + \alpha_1} \quad (8)$$

$$R(u,v) = \frac{\gamma_1 \cdot F(u) + \gamma_2 \cdot |\text{score}(u,v) - G(v)|}{\gamma_1 + \gamma_2} \quad (9)$$

Algorithm 1 Mutually Recursive Procedure

1: **Input:** Rating Network $(\mathcal{N}, \mathcal{E})$, $\alpha_1, \beta_1, \beta_2, \gamma_1, \gamma_2$
2: **Output:** fraudulence, unreliability and badness scores
3: Calculate $K_u, K_v, \varphi_v, \forall u \in \mathcal{U}, \forall v \in \mathcal{V}$
4: Initialize $F^0(u)$, $R^0(u,v)$, $G^0(v)$, with a Gaussian distribution, $(u,v) \in \mathcal{E}$, $t = 0$
5: **while** $error \geq \epsilon$ **do**
6: Update badness of items according to Eq.(10): $\forall v \in \mathcal{V}$,
7: Update unreliability of edges according to Eq.(9): $\forall (u,v) \in \mathcal{E}$,
8: Update fraudulence of users according to Eq. (8): $\forall u \in \mathcal{U}$,
9: $error = \max(\sum_{u\in\mathcal{U}} |F^t(u) - F^{t-1}(u)|, \sum_{v\in\mathcal{V}} |G^t(v) - G^{t-1}(v)|, \sum_{(u,v)\in\mathcal{E}} |R^t(u,v) - R^{t-1}(u,v)|)$
10: **end while**
11: **return** $F^t(u)$, $G^t(v)$, $R^t(u,v)$

$$G(v) = \frac{\sum_{(u,v)\in \text{In}(v)} R(u,v) \cdot \text{score}(u,v) + \beta_1 \cdot K_v + \beta_2 \cdot \varphi_v}{|\text{In}(v)| + \beta_1 + \beta_2}. \quad (10)$$

Here, $\alpha_1, \beta_1, \beta_2, \gamma_1, \gamma_2$ are non-negative integers, to avoid the situation where the node's access is 0. The value of $\alpha_1, \beta_1, \beta_2, \gamma_1, \gamma_2$ are set using parameter sweep, ensure that the denominator of (8)–(10) cannot be 0 (please see the detail description in Section IV-C).

We propose a fusion temporal bursts and score distribution approach to incorporate behavior properties into the formulation. Three equations are the set of mutually recursive definitions of **fraudulence, unreliability, and badness** of the proposed MRFS algorithm, by combing rating network and behavior properties together.

3) Algorithm Description: Algorithm 1 describes the proposed method to calculate the metrics for all users, items, and edges. The algorithm is an iterative algorithm, $F^t(u)$, $G(v)$, $R^t(u,v)$ denote the fraudulence, badness, and unreliability metric score at the end of iteration t. So we get the intrinsic metrics of the nodes and edges in the network. We initialize these values for the first time with a Gaussian distribution of 0 to 1. Then we iteratively update the scores until convergence (see lines 5–10). In detail, convergence occurs when all scores change minimally (see line 5). ϵ is the acceptable error bound, which is set to a very small value. In each iteration, the update of $F^t(u)$, $G^t(v)$, $R^t(u,v)$ takes constant time. The complexity of each iteration is $\mathcal{O}(|\mathcal{E}| + |\mathcal{N}|)$.

Based on the proposed fraudulence, badness, and unreliability score, the detection of the most anomalous subgraph from $\mathcal{N}$ and $\mathcal{E}$ can be formalized as the following optimization problem: $g(\mathcal{S})$. For the above optimization problem, in the worst case, the time cost is exponentially increasing with the node $|\mathcal{N}|$ in the network. Therefore, it is a better choice to develop approximate solutions. We give Algorithm 2, a greedy approach inspired by that of Fraud [9]. Here, we will solve the optimization problem with an efficient unconstrained optimization approach.

Algorithm 2 Greedy Procedure to Maximize a Metric g

Require: Bipartite network $(\mathcal{N}, \mathcal{E})$, $F^t(u)$, $G^t(v)$, $R^t(u, v)$, density metric g of the form (1).
Ensure: Return the largest $g(\mathcal{S})$.

```
MT = Construct priority tree of N
Bestg = calculate suspiciousness g(S) of node set
repeat
    n = Use the MT to pop the node with the lowest score for users or items
    N = N \ n, delete n from N, and the edges associated with n
    Bestg = calculate suspiciousness scores g(S) of current node set
    if Δ_n ≥ 0 then
        Update Bestg = Curg
    end if
    Update MT with respect to new N.
until N is empty or The value of g is the largest
Find suspicious subgraph S.
```

However, how do we optimize the density metric $g(\mathcal{S})$ to maximize the suspiciousness of a subgraph in near-linear time? We start with the entire network, and then repeatedly remove the node that results in the highest value of $g(\mathcal{S})$ evaluated on the remaining set of nodes. According to the calculation formula of $g(\mathcal{S})$, we should delete the node with the smallest suspicious value. Then recalculate Δ_n according to $\Delta_n = g(\mathcal{N} \setminus n) - g(\mathcal{N})$, which represents the change in g when we remove n from the current set. We will choose n to maximize Δ_n at each step. We then repeat this process: we recompute the values of Δ_n, then choose the next node to be deleted, and so on. When n is removed, we only need to update the nodes connected to it. Hence, the updates are fast: during the life cycle of the algorithm, we will perform at most one such update on this edge, for a total of $\mathcal{O}(|\mathcal{N}|)$, updates using appropriate data structures, as we next describe, each update can be performed in $\mathcal{O}(\log |\mathcal{N}|)$ time, total $\mathcal{O}(|\mathcal{N}| \log |\mathcal{N}|)$ time.

We construct a priority tree to help us efficiently find the user or item with a minimum metric score. The data structure is a binary tree with all $\mathcal{N}$ elements as leaves, all at the bottom level of the tree. Each non-leaf node keeps track of the maximum priority of its two child nodes. The priority tree updates the score of the user or item and maintains a new minimum when the priority changes. It also supports fast update priority: because all leaves are stored in a fixed location, we can easily retrieve any leaf nodes and update their priority. Then, after updating the priority of the node, we move the minimum values of the two child nodes up and update them to the parent node. Every operation on the MT requires $\mathcal{O}(\log |\mathcal{N}|)$ time.

To sum up, MRFS is scalable, which is near-linear time complexity in the number of edges. The time complexity of MRFS is shown in Fig. 3. The curve (blue) shows the running time of MRFS, compared with a linear function (black).

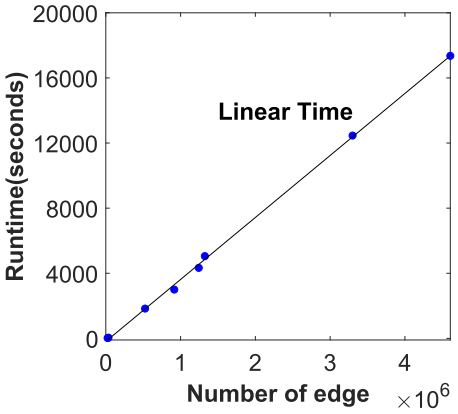


Fig. 3. Time complexity of MRFS.

TABLE II
DATASETS USED IN EXPERIMENTS

Network	# Users	# Items	# Edges
AmaMovies[42]	2,088,620	200,941	4,607,047
AmaBaby[42]	531,896	64,426	915,446
AmaVideo [42]	826,767	50,210	1,324,753
AmaOffice[42]	909,314	130,006	1,243,186
Yelp[1]	686,000	85,300	2,478,000
BeerAdvocate[2]	26,500	50,800	980,400
OTC [43]	4,814	5,858	35,592
Alpha [43]	3,286	3,754	24,186

IV. EXPERIMENTAL ANALYSIS

A. Dataset: Rating Networks

The statistical results of eight datasets, which are publicly available for academic research, are shown in Table II. Due to the lack of ground truth in the data, we use the injected synthetic data, using the same method as in [9]. We mimic the fraudsters' behaviors and randomly choose a certain number of objects as target items in our experiments. Usually, the suspiciousness of popular items is very small, unpopular items are inclined to hire fake reviews, and the indegree of the item we select is less than 100. Since fraudulent accounts can come from hijacked user accounts, we can also uniformly choose a certain number of users from the entire user set as fraudulent users. For camouflage fraudsters, we also imitate the normal user's scoring behavior, and let the fraudsters rate the popular products. To test on different fraudulent densities. The data we synthesize has a fraud density from 0.01 to 1 for testing. At the same time, we also mimicked the time and rating score behavior of fraudster attacks.

In detail, AmaBaby, AmaMovies, and AmaVideo are four collections of ratings about office products, baby-related products, Movies, and video products, respectively, on Amazon [42]. In addition, Yelp is a comment dataset from the largest comment site in the U.S., and BeerAdvocate is a comment dataset about beer. They can be modeled using the network (user, item, timestamp, rating score). Bitcoin over-the-counter (OTC) (Alpha) is a user-to-user trust network of Bitcoin users trading using the OTC (Alpha) platform [43]. The ground truth is defined as benign users are the platform's founder and users he rated highly positively. Fraudulent users are the ones that these trusted users uniformly rate negatively. The proportion

TABLE III
EXPERIMENTAL RESULTS ON REAL DATA WITH INJECTED DENSITY BLOCK

Datasets Name	F_1 score of user nodes				
	Fraudar	HoloScope	CatchSync	Rev2	**MRFS**
Amazon Video $\#C$	0.36	0.90	0.87	0.83	**0.92**
Amazon Video $\#H$	0.66	0.91	0.93	0.89	**0.97**
Amazon Movies $\#C$	0.74	0.82	0.83	0.80	**0.89**
Amazon Movies $\#H$	0.67	0.73	0.82	0.76	**0.95**
Amazon Office $\#C$	0.88	0.66	0.82	0.86	**0.97**
Amazon Office $\#H$	0.56	0.84	0.77	0.79	**0.90**
Datasets Name	F_1 score of item nodes				
	Fraudar	HoloScope	CatchSync	Rev2	**MRFS**
Amazon Video $\#C$	0.45	**0.97**	0.77	0.67	0.95
Amazon Video $\#H$	0.78	0.97	0.79	0.83	**0.98**
Amazon Movies $\#C$	0.77	0.89	0.73	0.82	**0.98**
Amazon Movies $\#H$	0.54	0.67	0.69	0.86	**0.87**
Amazon Office $\#C$	0.85	0.89	0.60	0.72	**0.96**
Amazon Office $\#H$	0.50	0.79	0.78	0.77	**0.91**

of fraudulent users in OTC and Alpha data sets is 3.7% and 3.1%.

B. Baseline Algorithms

We selected four dense-block detection methods as comparative methods.

1) *HoloScope [11]*: This is a graph topology-based weighting scheme that dynamically reweights objects according to our beliefs about which users are suspicious.
2) *Fraudar [9]:* This is an edge weighting scheme based on the inverse logarithm of objects' degrees, which was inspired by IDF.
3) *CatchSync [5]:* This is effective at both the classic problem of labeling suspicious behavior, as well as surfacing new patterns of unusual group behavior.
4) *Rev2 [13]:* This is a method that combines the network and behaviors, and analyze on the network without considering behavior attribute information.

C. Parameter Settings

The parameters are mainly distributed in Algorithm 1, but how do we set the values of $\alpha_1, \beta_1, \beta_2, \gamma_1$ and γ_2? In the unsupervised case, the best combination of these parameters cannot be confirmed. Therefore, the algorithm runs with for several combinations of $\alpha_1, \beta_1, \beta_2, \gamma_1$ and γ_2 as inputs, then find the average of $F(u), G(v)$ and $R(u, v)$. In our experiments, we varied all these parameters from 0 to 2, i.e., $0 \leq \alpha_1, \beta_1, \beta_2, \gamma_1$ and $\gamma_2 \leq 2$, giving $3^5 = 243$ combinations. In order to make the algorithm converge faster, ϵ is set to 0.01.

D. Detecting Suspiciousness Subgraph

We first run the algorithms on four collections of ratings for different types of commodities on Amazon (see Table II). It is very difficult to find low-density fraudsters than high-density fraudsters, so it is a better way to detect low-density fraudsters with high precision. We designed two injection schemes: Camouflage and Hijacked attacks with different injection densities as Fraudar [9]. We propose to use F_1-Score in order to give a comparison of datasets. We apply the following equation to compute F_1-Score:

$$F_1 = \frac{2 \times (\text{precision} \times \text{recall})}{\text{precision} + \text{recall}}. \tag{11}$$

1) *Injection Scheme C:* To simulate the camouflage attack models of fraudsters, we use three types of camouflage attacks as Fraudar: injection of fraud with no camouflage, random camouflage, and biased camouflage. We generate datasets by injecting a fraud group with varying configurations into AmazonVideo, AmazonMovires, and AmazonOffice. In each case, we inject 200 to 2000 fraudulent users and 100 to 1000 fraudulent items with various edge densities into real data.
2) *Injection Scheme H:* In the same way, we also generated hijacking fraud with different edge densities. For the "Hijacked" case, we use a random subset of existing users to form the fraudulent block, the fraudster's account has a camouflage pattern that is substantially similar to that of an honest user.

However, some honest users may add links to fraudulent target items in the real world. It may be that the fraudulent items have achieved the fraud effect and induced normal users. Taking this into consideration, we conducted another experiment using the attack model and added edges between honest users and fraud target items. But the density is sparser than fraud blocks. We added random edges in this subgraph. All other experimental settings have not changed. The fraud detection results of our MRFS and the baselines on the datasets are shown in Table III. Obviously, our method results are better than the baselines. MRFS has a similar and high F_1-score both in detecting fraudulent users and fraudulent items. As we find, MRFS achieves the best F_1-Score among the competitors in most tasks. And also, we find that HoloScope and Rev2 also have relatively good results, although they are not as good as the proposed MRFS. The main reason may be that: 1) HoloScope only considers some topological attributes but does not mine the deep feature of the network and 2) Rev2 combines the network topology and behaviors but not includes the attributes information of behaviors. However,

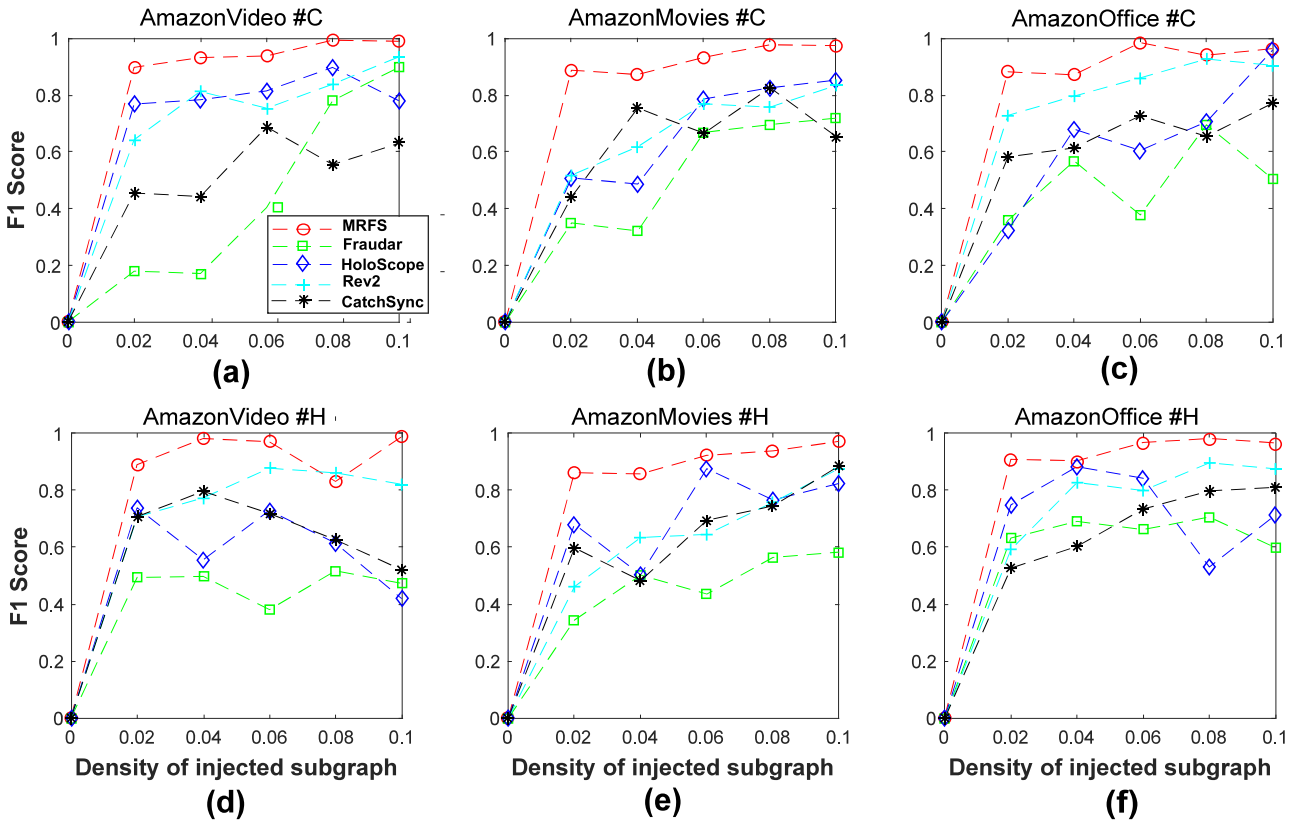


Fig. 4. MRFS outperforms competitors in multiple density of injected subgraph settings. F_1-Score of fraud detection on Amazon data in the experiment with Camouflage and Hijacking.

TABLE IV
$F_1 \geq 90\%$ MINIMUM DETECTION DENSITY

Data Name	Metrics	Nodes			
		Frauda	HoloScope	CatchSync	**MRFS**
Amazon Movies	AUC	0.8953	0.7535	0.8573	**0.9865**
	$F_1 \geq 90\%$	0.1355	0.2781	0.2531	**0.0263**
Amazon Baby	AUC	0.6953	0.7756	0.6751	**0.9723**
	$F_1 \geq 90\%$	0.5300	0.0376	0.4771	**0.0351**
Amazon Video	AUC	0.7351	0.8720	0.6223	**0.8934**
	$F_1 \geq 90\%$	0.4663	0.3526	0.5001	**0.0251**
BeerAdvocate	AUC	0.9142	0.7536	0.6345	**0.9366**
	$F_1 \geq 90\%$	0.4823	0.0621	0.6571	**0.0127**
Yelp	AUC	0.8532	0.7930	0.7548	**0.8733**
	$F_1 \geq 90\%$	0.1352	0.0453	0.7321	**0.0381**

the proposed MRFS considers the topological attributes of the network and the behaviors attributes of nodes and edges simultaneously. We better explore the essentials of nodes and edges in the rating network.

Fig. 4 shows the results of MRFS and four comparative methods on the Amazon networks. When the fraudulent density becomes low, the proposed method can still maintain high. Amazon #*C* network is injected into the density block in three camouflage ways, and then the average of the three F_1-Score is obtained. Compared to those comparative methods, our algorithm can keep as high F_1-Score as more than 90% before reaching 0.024 in density, which is far better than the baseline methods. HoloScope achieves a higher F_1-Score on the sparse network. But, HoloScope achieves excellent performance only when initialization is a priori fraudulent density block. The main reasons are that HoloScope just considers the topological attributes of the network, and the fraudulent density will greatly affect the performance of HoloScope. Therefore, Fig. 4 demonstrates that the proposed MRFS has good robustness.

To better illustrate the advantages of the proposed MRFS for detection on different data sets of injected density blocks, two measures are considered here: area under curve (AUC) and the lowest detection density when F value is greater than 90%. As shown in Table IV, MRFS is superior to other methods in fraud detection and can be able to detect fraud well even at low fraud density. MRFS has a minimum detection density 0.0127, which means that it can detect 20 000 fraudulent users to establish connections for 200 fraudulent items and obtain accurate results even under such low-density fraud. And also, Fig. 5 quantitatively demonstrates the ability of the proposed MRFS in networks. It can be seen from the bar chart that the algorithm has a good performance on real networks.

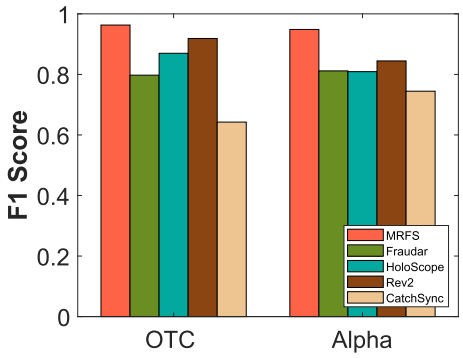


Fig. 5. Fraudsters detection on OTC and alpha networks.

V. Conclusion

In this work, we propose a new MRFS method of detecting fraud in large, edge-attributed real-world bipartite graphs, which have timestamps and rating scores (using the network). This graph include is commonplace in modern e-commerce platforms and other web services. We explored the nature of nodes and edges on the network, and more closely related to the suspiciousness of users and items. We considered the real malicious patterns of various frauds in the experiment, and the experimental results are very impressive. Our future work hopes to further detect fraudulent behaviors that combine structural patterns with behavioral attributes.

References

[1] L. Qi et al., "Finding all you need: Web APIs recommendation in web of things through keywords search," *IEEE Trans. Computat. Social Syst.*, vol. 6, no. 5, pp. 1063–1072, Oct. 2019.

[2] L. Qi, Q. He, F. Chen, X. Zhang, W. Dou, and Q. Ni, "Data-driven web APIs recommendation for building web applications," *IEEE Trans. Big Data*, vol. 8, no. 3, pp. 685–698, Jun. 2022.

[3] N. Shah, A. Beutel, B. Gallagher, and C. Faloutsos, "Spotting suspicious link behavior with fBox: An adversarial perspective," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2014, pp. 959–964.

[4] D. Cheng, S. Xiang, C. Shang, Y. Zhang, F. Yang, and L. Zhang, "Spatio-temporal attention-based neural network for credit card fraud detection," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 1, pp. 362–369.

[5] M. Jiang, P. Cui, A. Beutel, C. Faloutsos, and S. Yang, "CatchSync: Catching synchronized behavior in large directed graphs," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 941–950.

[6] Z. Li, G. Liu, and C. Jiang, "Deep representation learning with full center loss for credit card fraud detection," *IEEE Trans. Computat. Social Syst.*, vol. 7, no. 2, pp. 569–579, Apr. 2020.

[7] L. Qi, H. Song, X. Zhang, G. Srivastava, X. Xu, and S. Yu, "Compatibility-aware web API recommendation for mashup creation via textual description mining," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 17, no. 1, pp. 1–19, Jan. 2021.

[8] W. Gong, W. Zhang, M. Bilal, Y. Chen, X. Xu, and W. Wang, "Efficient web APIs recommendation with privacy-preservation for mobile app development in industry 4.0," *IEEE Trans. Ind. Informat.*, vol. 18, no. 9, pp. 6379–6387, Sep. 2022.

[9] B. Hooi, H. A. Song, A. Beutel, N. Shah, K. Shin, and C. Faloutsos, "FRAUDAR: Bounding graph fraud in the face of camouflage," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 895–904.

[10] A. Beutel, W. Xu, V. Guruswami, C. Palow, and C. Faloutsos, "CopyCatch: Stopping group attacks by spotting lockstep behavior in social networks," in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 119–130.

[11] S. Liu, B. Hooi, and C. Faloutsos, "HoloScope: Topology-and-spike aware fraud detection," in *Proc. ACM Conf. Inf. Knowl. Manag.*, Nov. 2017, pp. 1539–1548.

[12] B. A. Prakash, A. Sridharan, M. Seshadri, S. Machiraju, and C. Faloutsos, "Eigenspokes: Surprising patterns and scalable community chipping in large graphs," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*. Berlin, Germany: Springer, 2010, pp. 435–448.

[13] S. Kumar, B. Hooi, D. Makhija, M. Kumar, C. Faloutsos, and V. S. Subrahmanian, "REV2: Fraudulent user prediction in rating platforms," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, Feb. 2018, pp. 333–341.

[14] C. Wang and H. Zhu, "Representing fine-grained co-occurrences for behavior-based fraud detection in online payment services," *IEEE Trans. Depend. Sec. Comput.*, vol. 19, no. 1, pp. 301–315, Jan. 2022.

[15] G. Xu et al., "Delay-CJ: A novel cryptojacking covert attack method based on delayed strategy and its detection," *Digit. Commun. Netw.*, May 2022, pp. 1–14.

[16] L. Akoglu, R. Chandy, and C. Faloutsos, "Opinion fraud detection in online reviews by network effects," in *Proc. 7th Int. AAAI Conf. Weblogs Social Media*, 2013, pp. 1–11.

[17] M. Dadfarnia, F. Adibnia, M. Abadi, and A. Dorri, "Incremental collusive fraud detection in large-scale online auction networks," *J. Supercomput.*, vol. 76, no. 9, pp. 7416–7437, Sep. 2020.

[18] J. Nanduri, Y.-W. Liu, K. Yang, and Y. Jia, "Ecommerce fraud detection through fraud islands and multi-layer machine learning model," in *Proc. Future Inf. Commun. Conf.* Cham, Switzerland: Springer, 2020, pp. 556–570.

[19] D. Sarma, W. Alam, I. Saha, M. N. Alam, M. J. Alam, and S. Hossain, "Bank fraud detection using community detection algorithm," in *Proc. 2nd Int. Conf. Inventive Res. Comput. Appl. (ICIRCA)*, Jul. 2020, pp. 642–646.

[20] Q. Cao, X. Yang, J. Yu, and C. Palow, "Uncovering large groups of active malicious accounts in online social networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2014, pp. 477–488.

[21] G. Xu et al., "SG-PBFT: A secure and highly efficient distributed blockchain PBFT consensus algorithm for intelligent Internet of Vehicles," *J. Parallel Distrib. Comput.*, vol. 164, pp. 1–11, Jun. 2022.

[22] K. Shin, T. Eliassi-Rad, and C. Faloutsos, "Patterns and anomalies in k-cores of real-world graphs with applications," *Knowl. Inf. Syst.*, vol. 54, no. 3, pp. 677–710, Mar. 2018.

[23] M. Jiang, P. Cui, A. Beutel, C. Faloutsos, and S. Yang, "Inferring lockstep behavior from connectivity pattern in large graphs," *Knowl. Inf. Syst.*, vol. 48, no. 2, pp. 399–428, 2016.

[24] K. Shin, T. Eliassi-Rad, and C. Faloutsos, "CoreScope: Graph mining using k-core analysis—Patterns, anomalies and algorithms," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 469–478.

[25] T. Pourhabibi, K.-L. Ong, B. H. Kam, and Y. L. Boo, "Fraud detection: A systematic literature review of graph-based anomaly detection approaches," *Decis. Support Syst.*, vol. 133, Jun. 2020, Art. no. 113303.

[26] J. Li, T. Cai, K. Deng, X. Wang, T. Sellis, and F. Xia, "Community-diversified influence maximization in social networks," *Inf. Syst.*, vol. 92, Sep. 2020, Art. no. 101522.

[27] T. Cai, J. Li, A. Mian, R.-H. Li, T. Sellis, and J. X. Yu, "Target-aware holistic influence maximization in spatial social networks," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 4, pp. 1993–2007, Apr. 2022.

[28] Y. Ren, H. Zhu, J. Zhang, P. Dai, and L. Bo, "EnsemFDet: An ensemble approach to fraud detection based on bipartite graph," in *Proc. IEEE 37th Int. Conf. Data Eng. (ICDE)*, Apr. 2021, pp. 2039–2044.

[29] X. Song, J. Li, Q. Lei, W. Zhao, Y. Chen, and A. Mian, "Bi-CLKT: Bi-graph contrastive learning based knowledge tracing," *Knowl.-Based Syst.*, vol. 241, Apr. 2022, Art. no. 108274.

[30] N. Shah et al., "EdgeCentric: Anomaly detection in edge-attributed networks," in *Proc. IEEE 16th Int. Conf. Data Mining Workshops (ICDMW)*, Dec. 2016, pp. 327–334.

[31] Y. Wang and W. Xu, "Leveraging deep learning with LDA-based text analytics to detect automobile insurance fraud," *Decis. Support Syst.*, vol. 105, pp. 87–95, Jan. 2018.

[32] A. K. S. Yadav and M. Sora, "Fraud detection in financial statements using text mining methods: A review," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 1020, no. 1, Jan. 2021, Art. no. 012012.

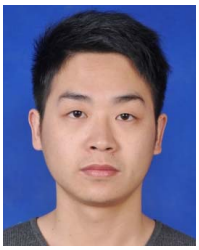
[33] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *Proceedings*. Louvain-la-Neuve, Belgium: Presses Universitaires de Louvain, 2015, p. 89.

[34] M. Zheng, C. Zhou, J. Wu, S. Pan, J. Shi, and L. Guo, "FraudNE: A joint embedding approach for fraud detection," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–8.

[35] S. Hou, Y. Ye, Y. Song, and M. Abdulhayoglu, "HinDroid: An intelligent Android malware detection system based on structured heterogeneous information network," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 1507–1515.

[36] N. K. Gyamfi and J.-D. Abdulai, "Bank fraud detection using support vector machine," in *Proc. IEEE 9th Annu. Inf. Technol., Electron. Mobile Commun. Conf. (IEMCON)*, Nov. 2018, pp. 37–41.

[37] D. Sisodia and D. S. Sisodia, "Quad division prototype selection-based k-nearest neighbor classifier for click fraud detection from highly skewed user click dataset," *Eng. Sci. Technol., Int. J.*, vol. 28, Apr. 2022, Art. no. 101011.
[38] S. Xie, G. Wang, S. Lin, and P. S. Yu, "Review spam detection via temporal pattern discovery," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 823–831.
[39] K. Shin, B. Hooi, and C. Faloutsos, "M-ZOOM: Fast dense-block detection in tensors with quality guarantees," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Cham, Switzerland: Springer, 2016, pp. 264–280.
[40] K. Shin, B. Hooi, J. Kim, and C. Faloutsos, "D-cube: Dense-block detection in terabyte-scale tensors," in *Proc. 10th ACM Int. Conf. Web Search Data Mining*, Feb. 2017, pp. 681–689.
[41] B. Hooi et al., "BIRDNEST: Bayesian inference for ratings-fraud detection," in *Proc. SIAM Int. Conf. Data Mining*. Philadelphia, PA, USA: SIAM, Jun. 2016, pp. 495–503.
[42] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: Understanding rating dimensions with review text," in *Proc. 7th ACM Conf. Recommender Syst.*, Oct. 2013, pp. 165–172.
[43] S. Kumar, F. Spezzano, V. S. Subrahmanian, and C. Faloutsos, "Edge weight prediction in weighted signed networks," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 221–230.

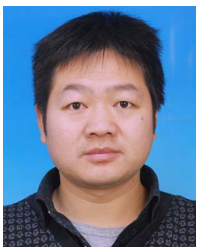


Wei Yu received the B.S. degree from Jinggangshan University, Ji'an, China, in 2012, the M.S. degree from Henan Normal University, Xinxiang, China, in 2015, and the Ph.D. degree from the School of College of Intelligence and Computing, Tianjin University, Tianjin, China, in 2020.

He is currently a Lecturer with Tianjin University, Tianjin, China. His research interests include dynamic complex network analysis, anomaly detection, large-scale data mining, and machine learning.

Wenkai Wang received the B.S. degree from Shanxi University, Taiyuan, China, in 2016, and the M.S. degree from the School of College of Intelligence and Computing, Tianjin University, Tianjin, China, in 2020.

His research interests include dynamic complex network analysis, anomaly detection, fraud detection, and machine learning.



Guangquan Xu (Member, IEEE) received the Ph.D. degree from Tianjin University, Tianjin, China, in March 2008, where he is currently pursuing the Ph.D. degree with the Tianjin Key Laboratory of Advanced Networking (TANK), College of Intelligence and Computing.

He is also a Full Professor with TANK. He is also a Joint-Professor with the Big Data School, Huanghai University, Qingdao, China. He is the Director of the Network Security Joint Laboratory, Tianjin, China and the Network Attack and Defense Joint Laboratory, Tianjin, China. He has published more than 100 papers in reputable international journals and conferences, including IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY (TIFS), IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING (TDSC), IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS (TII), IEEE INTERNET OF THINGS JOURNAL (IoT J), Future Generation Computer Systems (FGCS), *IEEE Communications Magazine*, *Information Sciences*, IEEE WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON CYBERNETICS, ACM TRANSACTIONS ON INTELLIGENT SYSTEMS AND TECHNOLOGY, and *IEEE Network*. His research interests include cybersecurity and trust management.

Dr. Xu is an IET (The Institution of Engineering and Technology) Fellow and a member of China Computer Federation (CCF). He served as a TPC Member for International Conference on Frontiers in Cyber Security (FCS) 2020, IEEE International Conference on Ubiquitous Intelligence and Computing (UIC) 2018, International Conference on Security and Privacy in New Computing Environments (SPNCE) 2019, IEEE UIC 2015, and IEEE International Conference on Engineering of Complex Computer Systems (ICECCS) 2014; and a Reviewer for IEEE ACCESS, ACM TIST, *Journal of Parallel and Distributed Computing (JPDC)*, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS (TITS), *Soft Computing*, FGCS, and *Computational Intelligence*.

Huaming Wu (Senior Member, IEEE) received the B.E. and M.S. degrees in electrical engineering from the Harbin Institute of Technology, Harbin, China, in 2009 and 2011, respectively, and the Ph.D. degree (Hons.) in computer science from Freie Universität Berlin, Berlin, Germany, in 2015.

He is currently an Associate Professor with the Center for Applied Mathematics, Tianjin University, Tianjin, China. His research interests include model-based evaluation, wireless and mobile network systems, mobile cloud computing, and deep learning.

Hongyan Li received the M.S. degree in computer science and technology from the Changchun University of Technology, Changchun, China, in 2019.

She is currently a Lecturer with Zhejiang Yuexiu University, Shaoxing, China. Her research interests include data mining and machine learning.

Jun Wang received the M.S. degree in fundamental mathematics from the Beijing Institute of Technology, Beijing, China, in 2007.

He is currently a Lecturer with Tianjin University, Tianjin, China. His research interests include dynamic complex network analysis, anomaly detection, large-scale data mining, and machine learning.

Xiaoming Li received the master's degree from Xinjiang University, Ürümqi, China, in 2008, and the Ph.D. degree from Tianjin University, Tianjin, China, in 2020.

He is currently a Professor with the School of International Business, Zhejiang Yuexiu University of Foreign Languages, Shaoxing, China. His research interests include human behavior dynamics and multilayer network local community detection.

Juan Liu received the Ph.D. degrees from Tianjin Medical University, Tianjin, China, in September 2016 and July 2017, and the Ph.D. degree from the Dental College, Tohoku University, Sendai, Japan. She is currently pursuing the Ph.D. degree with the Dental College, Tianjin Medical University, in 2017.

She is also an Attending Doctor with the Stomatological Hospital, Tianjin Medical University. She has published more than ten articles in international journals and Chinese core magazines. Her research interests include surface modification of oral implant materials and occlusion-masticaltory muscle-temporomandibular joint disease.

Dr. Liu is a member of the Chinese Stomatological Association (CSA) and the Prosthodontics Committee of CSA.