A Contrast Metric for Fraud Detection in Rich Graphs

Shenghua Liu, Bryan Hooi, and Christos Faloutsos

Abstract—How can we detect fraud in a big graph with rich properties, as online fraudsters invest more resources, including purchasing large pools of fake user accounts and dedicated IPs, to hide their fraudulent attacks? To achieve robustness, existing approaches detected dense sub-graphs as suspicious patterns in an unsupervised way, such as average degree maximization. However, such approaches suffer from the bias of including more nodes than necessary, resulting in lower accuracy and increased need for manual verification. Therefore, we propose HoloScope, which introduces a novel metric "contrast suspiciousness" integrating information from graph topology and spikes to more accurately detect fraudulent users and objects. Contrast suspiciousness dynamically emphasizes the contrasting patterns between fraudsters and normal users, making HoloScope capable of distinguishing the synchronized and strange behaviors of fraudsters by means of topology, bursts and drops, and rating scores. In addition, we provide theoretical bounds for how much this method increases the time cost needed for fraudsters to conduct adversarial attacks. Moreover, HoloScope has a concise framework and sub-quadratic time complexity, making the algorithm reproducible and scalable. In extensive experiments, HoloScope achieved significant accuracy improvements on real data with injected labels and true labels, when compared with state-of-the-art fraud detection methods.

Index Terms—Graph mining, time series, fraud detection, contrast suspiciousness

1 Introduction

How can we detect online fraudsters as they manipulate geo-locations, internet providers, and IP addresses via large IP pools (with as many as 852,992 dedicated IPs) to hide fraudulent attacks? Online fraud has become an increasingly serious problem because of the high potential reward it offers to fraudsters, which can be as much as $5 million from 300 million fake "views" per day, according to a report [1] in December 2016. Suppose that a fraudster has a accounts or IPs, and serves a customer who wants to buy 200 ratings or clicks for each of b products. We suppose that each account can only rate one product once as most of systems do. Since the fraudster has to add 200 ratings to each product out of a possible a ratings, the density of the fraudulent block created is: $(200 \cdot b)/(a \cdot b) = 200/a$. Thus, with a large number of enough user accounts or IPs, the fraudster can serve as many products as needed while keeping the density low. This presents a difficult challenge for most existing fraud detection methods.

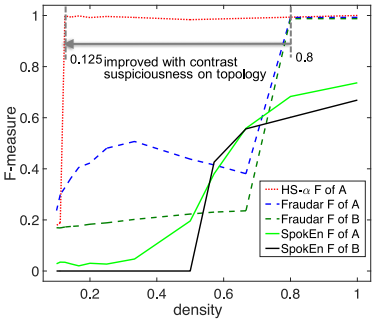
Supervised classification methods lose their advantages in this situation, since fraudsters are capable of adversarial manipulation of some key model features without much effort [2]. Graphs are built on all the behaviors between users or between users and objects, which cannot be avoided. An unsupervised method based on graphs then attracts more researchers, i.e., dense block detection, since of its robustness of fraud detection. Current dense block detection methods [3], [4], [5] maximize the arithmetic or geometric average degree. We use "fraudulent density" to indicate the edge density that fraudsters create for target objects. However, those methods have a bias of including more nodes than necessary, especially as the fraudulent density decreases, as we verified empirically. This bias results in low precision, which then requires intensive manual work to verify each user. Fraudar [6], [7] proposed an edge weighting scheme based on the inverse logarithm of objects' degrees to reduce this bias, a scheme inspired by IDF [8], [9]. However, this weighting scheme is fixed globally and affects both suspicious and normal edges, lowering Fraudar's precision, as can be seen from results using semi-real (with injected labels) and real data (see Fig. 1).

Researchers gradually realized that purely measuring suspiciousness with topological density limited the performance of fraud detection. Accurately detecting fraudulent blocks of lower density requires aggregating more sources of information [4], [10], [11]. Consider the attribute of the creation time of edges: fraudulent attacks tend to be concentrated in time; e.g., fraudsters may surge to retweet a message, creating one or more sudden bursts of activity [12], [13], followed by sudden drops after the attack is complete. Sudden bursts and drops have not been directly considered together in previous work.

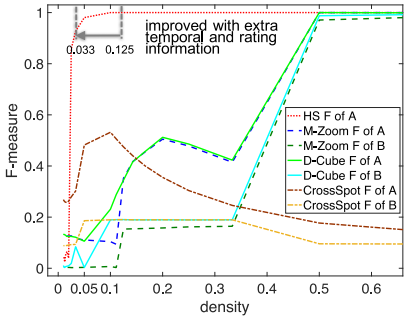
Tensor-based methods provide a natural way to integrate different attributes of graph edges. [4], [5], [10], [14]

- *S. Liu is with the CAS Key Laboratory of Network Data Science & Technology, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100864, China, and Carnegie Mellon University, Pittsburgh, PA 15213. E-mail: liu.shengh@gmail.com.*
- *B. Hooi and C. Faloutsos are with Carnegie Mellon University, Pittsburgh, PA 15213. E-mail: bhooi@andrew.cmu.edu, christos@cs.cmu.edu.*

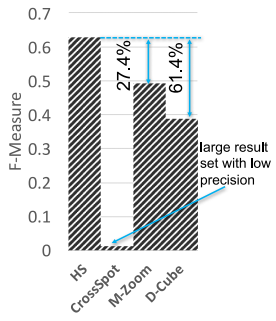
Manuscript received 4 Mar. 2018; revised 16 Aug. 2018; accepted 8 Oct. 2018. Date of publication 17 Oct. 2018; date of current version 4 Nov. 2019. (Corresponding author: Shenghua Liu.)
Recommended for acceptance by T. Palpanas.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2018.2876531



(a) HS-α using topology information outperforms baselines



(b) HS using holistic attributes provides clear improvement, and performs the best.



(c) HS achieves the best F-measure, on real data from Sina Weibo

Fig. 1. (a) and (b) show experimental results using a BeerAdvocate dataset. The better methods are able to detect fraud with high accuracy, even when fraudulent density (plotted on the horizontal axis) is low. Both HS-α and HS are our methods, where the former only uses topology information. We increase the number of injected fraudsters from 200 to 2,000 for HS-α, and to 20,000 for HS, while the decreasing density of fraudulent edges is shown on the horizontal axis from right to left. Compared with HS-α, HS, which makes holistic use of several signals achieves further improvement. (c) shows accuracy (F measure of precision and recall) results on Sina Weibo, with ground truth labels.

incorporate rich attributes into a multi-mode tensor formulation, e.g., IPs, rating scores and time. However, those methods rely on time-binning to incorporate temporal information, and they then treat time bins independently, which loses information about bursts and drops.

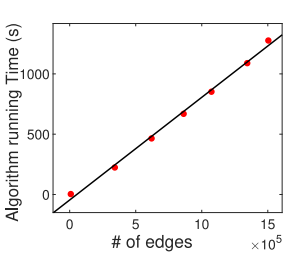
Therefore, we propose HoloScope, an unsupervised approach, which combines suspicious signals from graph topology, temporal bursts and drops, and rating deviation. Our contrast-based weighting scheme dynamically re-weights objects according to our beliefs about which users are suspicious. On topology, HoloScope captures the suspiciousness that arises when an object is connected mostly by suspicious users while other users seldom connect to it. Temporally, HoloScope detects suspicious spikes of bursts and drops, which increases the time cost needed for fraudsters to conduct an attack. In terms of rating, our approach takes into account the deviation between an object's ratings as given by suspicious users and non-suspicious users. For popularizing fraudsters, the ratings would be heavily biased towards high scores, while honest users give low scores to low-quality objects [15]. KL-divergence is naturally used as the metric for the deviation of distributions of ratings given by the two groups of users.

In summary, our contributions are:

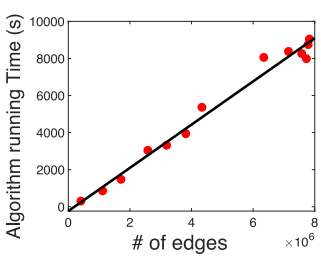
- *Novel suspiciousness metric:* We propose a dynamic *contrast suspiciousness* metric, which emphasizes the contrast behaviors between fraudsters and honest users in an unsupervised way. At the same time, the contrast suspiciousness provides a unified suspiciousness framework, which can make holistic use of several signals including, but not limited to, connectivity (i.e., topology), temporal bursts and drops, and rating deviation in a systematic way.
- *Robustness and theoretical analysis of fraudsters' obstruction:* We show that if the fraudsters use less than a theoretical bound of time for an attack, they will cause a suspicious drop or burst. In other words, HoloScope obstructs fraudsters by increasing the time needed to perform an attack. This theorem guarantees temporal robustness: no matter how the fraudsters manipulate the creation time of fraudulent links, they will be caught if the attack takes less than a fixed amount of time.
- *Effectiveness:* We achieved higher accuracy than the baselines on semi-real and real datasets. In fact, HoloScope using only topology information (HS-α) outperformed the graph-based baselines (see Fig. 1a), while HoloScope (HS) using all signals achieved further improvement outperforming the tensor-based baselines (see Figs. 1b and 1c). The dynamic weighting of object nodes with contrast suspiciousness enables both HS-α and HS achieve better detection accuracy.
- *Scalability:* The running time of HoloScope increases in an almost linear relationship with the number of edges (see Fig. 2). The algorithm is theoretically proved as fast as sub-quadratic time of the number of nodes, with a reasonable assumption.

In addition, using microblog Sina Weibo[1] data, HoloScope achieved a higher F-measure than the baselines in detecting the ground truth labels, with high precision and recall. The code of HoloScope is open-sourced for reproducibility.[2]

The rest of the paper is organized as follows. Section 2 summarizes the related works. In Section 3, we describe our contrast suspiciousness metric, and propose an algorithm for detecting fraud. Experiments are designed and conducted in Section 4. Section 5 presents conclusions.



(a) BeerAdvocate dataset



(b) Amazon Electronics dataset

Fig. 2. HoloScope (HS) runs in near-linear time.

1. The largest microblog service in China, http://www.weibo.com
2. https://github.com/shenghua-liu/HoloScope

2 Related Works

The labels for fraudsters are very limited, probably causing supervised machine learning methods to overfit some specific features from labeled fraudsters, e.g., rating to the same products at the same time, and using the same words. Because of this overfitting, such methods lack robustness. Thus most existing works study fraud detection in an unsupervised way due to the limited amount of labels, which are based on the density of blocks within adjacency matrices [16], [20], or multi-way tensors [4], [5]. OddBall [21] found new rules and patterns in the distribution of eigenvalues for anomaly detection. [20] detected suspicious patterns in the plots of singular vector decomposition (SVD). Instead of detecting density block by average degree [3], [22] and CoreScope [23] proposed to use Shingling and K-core algorithms respectively to detect anomalous dense blocks in huge graphs. Taking into account the suspiciousness of each edge or node in a real-world graph potentially allows for more accurate detection. Fraudar [6], [7] proposed to weight edges' suspiciousness by the inverse logarithm of objects' in-degree, to discount popular objects. [24] found that the degrees in a large community follow a power law distribution, forming hyperbolic structures. This suggests penalizing high degree objects to avoid unnecessarily detecting the dense core of hyperbolic community [25]. The spikes in degree distributions were studied, and synchronized behaviors were detected in [26], [27]. Deep neural network methods are used for anomaly detection [28], [29], but these are black-box approaches that provide little interpretability regarding the detected output.

In addition to topological density, EdgeCentric [15] studied the distribution of rating scores to find the anomalies. In terms of temporal attributes, the identification of burst periods has been studied in [30]. A recent work, Sleeping Beauty (SB) [31], more intuitively defined the awakening time for a paper's citation at the beginning of burst period. [32] detected the outliers of time series as the changing point. [33] clustered the temporal patterns of text phrases and hash tags in Twitter, and [34], [35] studied the temporal dynamics of networks separately on ego-network and network motifs. Meanwhile, [11], [36] modeled the time stamped rating scores with Bayesian model and autoregression models respectively for anomalous behavior detection. Even though [12], [37], [38] have used burst patterns to detect review spam, a sudden drop in temporal spikes has not been considered yet. [12] detected spams in singleton reviews, where each spammer writes only no more than one review in the system. The algorithms found the common period in which multiple time series have bursts, including the time series of ratios of singleton reviewers. We solve a different fraud detection problem in which spammers have to reuse the limited accounts to create as many fake reviews as possible. Hence our method finds a suspicious signal from topological connections.

Aggregating suspiciousness signals from different attributes is challenging for unsupervised learning. [39] proposed Reciprocal Rank Fusion (RRF) scores for combining different rank lists in information retrieval. However, RRF applies to ranks, which throws away the suspiciousness scores in fraud detection.

Without explicitly applying aggregation, researchers used the tensor-based methods to consider different attributes.

TABLE 1
Comparison between HoloScope and Other Fraud Detection Algorithms

	Fraudar [6], [7]	SpokEn [16]	CopyCatch [17]	CrossSpot [10]	BP-based methods [18], [19]	M-Zoom/D-Cube [4], [5]	HoloScope
scalability	✓	✓	✓	✓	✓	✓	✓
camouflage	✓	✓		✓		✓	✓
hy-community			?	?			✓
spike-aware			?				✓
rating deviation				✓			✓

CrossSpot [10], [14], a tensor-based algorithm, estimated the suspiciousness of a block using a Poisson model. However, it did not take into account the difference between popular and unpopular objects. Moreover, although CrossSpot, M-Zoom [4] and D-Cube [5] can consider edge attributes like rating time and scores via a multi-mode tensor, they require a time-binning approach. When time is split into bins, attacks that create bursts and drops may not stand out clearly after time-binning, since each time bin is treated as an independent dimension in the temporal mode of tensor. The problem of choosing bin widths for histograms was studied by Sturges [40], assuming an approximately normal distribution, and by Freedman-Diaconis [41] based on statistical dispersion. However, the binning approaches were proposed for the time series of a single object, which is not appropriate for different kinds of objects in a real-world graph; that is, popular products and unpopular products should use different bin sizes.

Belief propagation (BP) [18] is another common approach for fraud detection that can incorporate some specific edge attributes, such as rating and sentiments of reviews associated with edges [19]. It can also use seeds discovered from suspicious patterns [26] as a prior knowledge. However, its robustness against adversaries that try to hide themselves is not well understood. Based on a similar idea, CopyCatch [17] detected lockstep behavior by maximizing the number of edges in blocks constrained within time windows. However, this approach ignores the distribution of edge creation times within the window, and it does not capture bursts and drops directly.

Finally, we summarize the previous baselines compared to our HoloScope in Table 1. Camouflage means that fraudsters rate some normal or popular objects to behave like honest users, besides the fraudulent rating. Here we use "camouflage" to indicate that the detection algorithms are not sensitive to camouflage, or resistance as proved in [6], [7]. We use "hy-community" to indicate whether the method can avoid detecting the naturally-formed hyperbolic topology that is unnecessary (false positive) for fraud detection. Therefore, HoloScope is the only one that considers the complete property list, especially including temporal spikes (sudden bursts and drops, and multiple bursts), hyperbolic topology, and rating deviation in a unified suspiciousness framework.

3 Proposed Approach

The definition of our problem is as follows.

Problem 1 (Informal definition). Given *quadruplets* $(user, object, timestamp, \#stars)$, *where* $timestamp$ *is the time when a* $user$ *rates an* $object$, *and* $\#stars$ *is the categorical rating scores:*

- Find *a group of suspicious users, and suspicious objects or a rank with suspiciousness scores, and*
- to optimize *the metric under the common knowledge of suspiciousness from topology, rating time and scores.*

To make the problem more general, $timestamp$ and $\#stars$ are optional. For example, in Twitter, we have $(user, object, timestamp)$ triples, where a $user$ retweets a message $object$ at $timestamp$. In a static following network, we have pairs $(user, object)$, with $user$ following $object$.

Our metric should capture the following basic traits.

First, the fraudsters need to create as many fake reviews as they can to boost fraudulent products.

Trait 1 (Engagement). Fraudsters engage as much firepower as possible to boost customers' objects, i.e., suspicious objects.

Second, as [6] suggested, a popular object is not likely a fraudulent object. In other words, suspicious objects attract less attention from ordinary users due to their low quality. Then we have:

Trait 2 (Less Involvement). Suspicious objects seldom attract non-fraudulent users to connect with them.

Third, fraudsters conduct their attacks in a short period of time, creating temporal spikes with bursts and sudden drops, as reported in previous works [12], [13].

Trait 3 (Spikes: Bursts and Drops). Fraudulent attacks are concentrated in time, sometimes over multiple waves of attacks, creating bursts of activity. Conversely, the end of an attack corresponds to sudden drops in activity.

Finally, the rating distribution of fraudsters differs greatly from those of typical users, as observed by [15]. This occurs because fraudsters are aiming to manipulate the rating of products.

Trait 4 (Rating Deviation). The rating behavior of fraudsters deviates greatly from the rating behavior of normal users.

In the following sections, we will show that our proposed metric can make holistic use of several signals, namely topology, temporal spikes, and rating deviation, to locate suspicious users and objects satisfying the above traits. That is why we call our method as HoloScope.

3.1 HoloScope Metric

To give a formal definition of our metric, we describe the quadruplets ($user$, $object$, $timestamp$, $\#stars$) as a bipartite and directed graph $\mathcal{G} = \{U, V, E\}$. U is the source node set corresponding to users, V is the sink node set corresponding to objects, and E contains the directed edges from U to V. Generally, graph $\mathcal{G}$ is a multigraph, i.e., multiple edges can be present between two nodes. Multiple edges mean that a user can repeatedly comment or rate on the same product at a different time, as is common in practice. Users can also retweet messages multiple times in Sina Weibo. Each edge can be associated with rating scores ($\#stars$), and timestamp, for which the data structure is introduced in Section 3.1.2.

Our HoloScope metric detects fraud from three perspectives: topology connection, timestamp, and rating score. To easily understand the framework, we first introduce the HoloScope in a perspective of topology connection. Afterwards, we show how we aggregate the other two perspectives into HoloScope. We first view $\mathcal{G}$ as a weighted adjacency matrix $\mathbb{M}$, with the number of multiple edges (i.e., edge frequency) as matrix elements.

Lockstep behaviors are always a strong signal of fraud, in which a group of suspicious source nodes $A \subset U$ intensively act on a group of sink nodes $B \subset V$. Based on Trait 1, the total engagement of source nodes A with sink nodes B can be basically measured via density measures. There are many density measures, such as arithmetic and geometric average degree. Our HoloScope allows for any such measure. However, as the average degree metrics have a bias toward including too many nodes, we use a measure denoted by $D(A, B)$ as the basis of the HoloScope, defined as

$$D(A, B) = \frac{\sum_{v_i \in B} f_A(v_i)}{|A| + |B|}, \tag{1}$$

where $f_A(v_i)$ is the total edge frequency from source nodes A to a sink node v_i. $f_A(v_i)$ can also be viewed as an engagement from A to v_i, or A's lockstep on v_i, which is defined as

$$f_A(v_i) = \sum_{(u_j, v_i) \in E \wedge u_j \in A} \sigma_{ji} \cdot e_{ji}, \tag{2}$$

where constant σ_{ji} is the global suspiciousness on an edge, which can be equal to 1 if no extra global suspiciousness is assigned to a node pair (u_j, v_i). We will propose a way to assign the suspiciousness in Section 3.1.2. e_{ji} is the element of adjacency matrix $\mathbb{M}$, i.e., the edge frequency between a node pair (u_j, v_i). The edge frequency e_{ji} becomes a binary in a simple graph. The global suspiciousness as a prior information can come from the degree, and from additional domain knowledge, such as duplicated review sentences and unusual behaving time.

To maximize $D(A, B)$, the suspicious source nodes A and the suspicious sink nodes B are mutually dependent. Therefore, we introduce an informal definition of *contrast suspiciousness*:

Definition 1 (Contrast suspiciousness). *The contrast suspiciousness denoted as* $P(v_i \in B|A)$ *is defined as the conditional likelihood of a sink node* v_i *that belongs to* B *(the suspicious object set), given the suspicious source nodes* A.

A visualization of the contrast suspiciousness is given in Fig. 3. The intuitive idea behind contrast suspiciousness is that in most cases, we need to judge the suspiciousness of objects by currently chosen suspicious users A, e.g., an object is more suspicious if very few users not in A are connected to it (see Trait 2); the sudden burst of an object is mainly caused by A (see Trait 3); or the rating scores from A to an object are

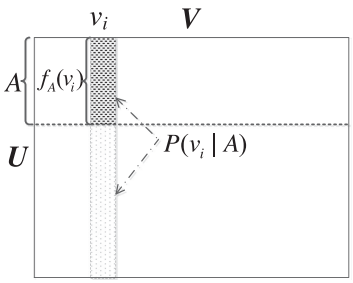
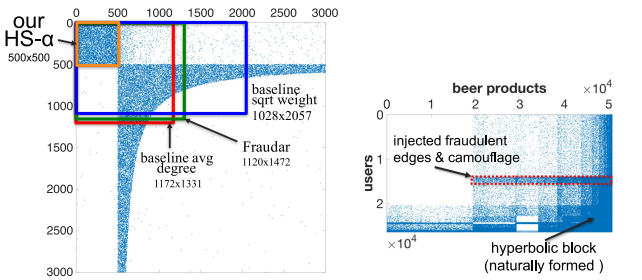


Fig. 3. An intuitive view of our definitions in the HoloScope.



(a) HS-α finds the exact dense block in the synthetic data (b) Hyperbolic community in BeerAdvocate data

Fig. 4. (a) The synthetic data consists of hyperbolic and rectangular blocks, with volume density around 0.84 and 0.60, respectively. The camouflage is randomly biased towards columns of high degree. Existing methods without considering contrast suspiciousness, i.e. Trait 2, do not detect the most suspicious area precisely. They sometimes even miss smaller density rectangular areas, due to the existence of hyperbolic connections. (b) A real data of naturally-formed hyperbolic community, and injected dense block. The injection is 2000×200 with biased camouflage.

quite different from those of other users (see Trait 4). Therefore, such suspiciousness makes use of the contrasts between users in A and users not in A, or the whole set.

Finally, instead of maximizing $D(A, B)$, we maximize the following expectation of suspiciousness $D(A, B)$ over the probabilities $P(v_i \in B|A)$

$$\max_A HS(A) := \mathbb{E}[D(A,B)] = \frac{1}{|A| + \sum_{k \in V} P(v_k|A)} \sum_{i \in V} f_A(v_i) P(v_i|A), \quad (3)$$

where for brevity we write $P(v_i|A)$ as meaning $P(v_i \in B|A)$. $1 - P(v_i|A)$ is the probability of v_i being a normal sink node. We dynamically calculate the contrast suspiciousness for all objects, after every choice of source nodes A.

Using this overall framework for our proposed metric $HS(A)$, we next show how to satisfy the remaining traits. To do this, we define contrast suspiciousness $P(v_i|A)$ in a way that takes into account various edge attributes. This will allow greater accuracy particularly in detecting low-density blocks.

3.1.1 HS-α: Less Involvement from Others

As previous work [24] has shown, large communities form hyperbolic structures, which we generate in our synthetic data (see the lower-right block in Fig. 4a), and we also find in real BeerAdvocate data (see Fig. 4b). Thus, if we compare the rectangular dense block and the hyperbolic core (i.e., the upper left part of hyperbolic block in Fig. 4a), which one is more suspicious? Fig. 4b also shows the comparison between the naturally formed hyperbolic community and the injected fraudulent block with random camouflage on popular objects. By examination in the scenario of online reviews, the products in a hyperbolic core are also rated by many other people not in the core, with high scores. In a rectangular dense block, the products seldom attract other people to give high scores, which may not be really a good products. So a rectangular dense block is more suspicious.

Existing algorithms including Fraudar and those based on average degree or square-root (sqrt) weight always find a dense block overlapping both the rectangular block at the top left and the hyperbolic block, to maximize their objectives (see Fig. 4a).

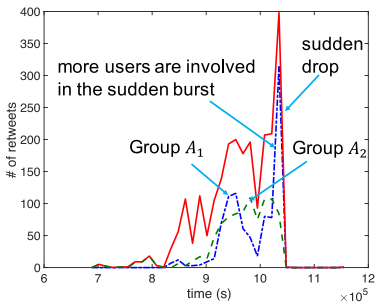
Based on Trait 2, a sink node should be more suspicious if it attracts connections mostly from the suspicious source nodes A and rarely from other nodes. Mathematically, we capture this by defining

$$P(v_i|A) \propto q(\alpha_i), \quad \text{where} \quad \alpha_i = \frac{f_A(v_i)}{f_U(v_i)}, \quad (4)$$

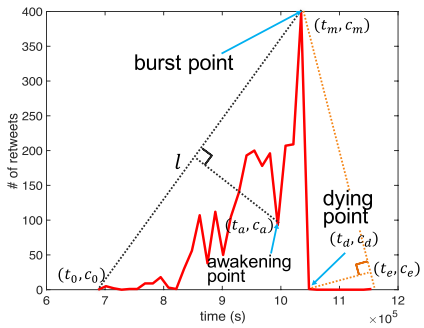
where $f_U(v_i)$ is the weighted in-degree of sink node v_i. Similar to $f_A(v_i)$, the edges are weighted by global suspiciousness. $\alpha_i \in [0, 1]$ measures the involvement ratio of A in the activity of sink node v_i. The scaling function $q(\cdot)$ is our belief about how this ratio relates to suspiciousness, and we choose the exponential form $q(x) = b^{x-1}$, where base $b > 1$. Such a function form can rapidly boost the ratio for a high x, and keep the ratio in a steady and low level for a low x. Since $x \in [0, 1]$, the scaling function $q(\cdot)$ can be viewed as a variant of half Sigmoid function. The details for function $q(x)$ are described in Section 3.3.

For clarity, our HoloScope method is denoted as HS-α when it is applied only on adjacency matrix of a graph. In Fig. 4a, the results of the synthetic data show that HS-α detected the exact dense rectangular block ($b = 128$), whereas the other methods included numerous non-suspicious nodes from the core part of the hyperbolic community, resulting in low accuracy. In the beer review data from the BeerAdvocate website, testing on different fraudulent density (see Fig. 1a), our HS-α remained at high accuracy, whereas the other methods' accuracy dropped quickly when the density fell below 70 percent. The reason for these results is that the existing metrics assign static weights, rather than using our beliefs about which users are suspicious to dynamically update the weights.

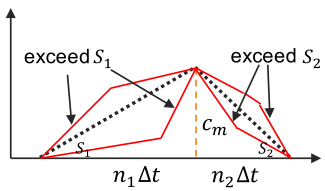
The main idea is that HS-α can do better because it dynamically adjusts the weights for sink nodes, penalizing those sink nodes that also have many connections from other source nodes not in A. The weights are used for every edge connected to those sink nodes. In contrast, although Fraudar proposed to penalize popular sink nodes based their in-degree, these penalties also scaled down the weights of suspicious edges. The Fraudar (green box) improved the unweighted "average degree" method (red box) by only a very limited amount. Moreover, with a heavier penalty, the "sqrt weight" method (blue box) achieved better accuracy on source nodes but poorer accuracy on sink nodes, since those methods used globally fixed



(a) Group A_1 is more suspicious than A_2



(b) Detection of burst and drop



(c) Time cost obstruction

Fig. 5. (a) and (b) are the time series (histogram) of a real message being retweeted in Sina Weibo. The horizontal axis is the number of seconds after $2013-11-1$. In (a), A_1 and A_2 have the same number of retweets of the message. A_1 is more suspicious than A_2 since A_1 has more users involved in the sudden burst, which causes the strange burst and drop. (c) is the auxiliary for the proof of time cost obstruction.

weights, and the weights of suspicious nodes were penalized as well. Hence the hyperbolic structure pushes those methods to include more nodes from its core part.

In summary, our HS-α using dynamic contrast suspiciousness can improve the accuracy of fraud detection in 'noisy' graphs (containing hyperbolic communities), even with low fraudulent density.

3.1.2 Temporal Bursts and Drops

Timestamps for edge creation are commonly available in most real settings. If two subgroups of microblog users have the same number of retweets of a message, can we say that they have the same suspiciousness? In the example shown in Fig. 5a, the red line is the time series of the total retweets of a message in Sina Weibo. The blue dotted line and green dashed line are the retweeting time series, respectively, from user groups A_1 and A_2. The two series have the same area under the time series curves, i.e., the same number of retweets. However, considering that fraudsters tend to surge to retweet a message to reduce the time cost, the surge should create one or more sudden bursts, along with sudden drops. Therefore, the suspiciousness of user group A_1 and that of A_2 become quite different even though they have the same number of retweets, which cannot be detected solely based on connections in the graph. Thus we include the temporal attribute in our HoloScope framework for defining contrast suspiciousness.

To include the temporal attribute, we introduce a simple but effective data structure, the *property indexing matrix* (PIM). Each kind of property is stored in a list of property entries. Each property entry is associated with a pair composed of a source node and a sink node. The timestamps of multiple edges form a list and are associated with the corresponding entry. The PIM has a similar structure to the weighted adjacency matrix of $\mathcal{G}$, except that it stores the index of a property entry as an element, instead of the connection frequency. we can easily find the timestamps of edges from source nodes in A via "slicing" operations on the PIM.

Denote the list of timestamps of edges connected to a sink node v as T_v. To simplify our notation, we use T without subscript when talking about a single given sink node v. Let $\mathcal{T} = \{(t_0, c_0), (t_1, c_1), \ldots, (t_e, c_e)\}$ as the *time series* of T, i.e., the histogram of T. The count c_i is the number of timestamps in the time bin $[t_i - \Delta t/2, t_i + \Delta t/2)$, with bin size Δt. The bin size of the histogram is calculated according to the maximum of Sturges criteria and the robust Freedman-Diaconis' criteria [41]. Note that the HoloScope can *tune* different bin sizes for different sink nodes, e.g., popular objects need fine-grained bins to explore detailed patterns. Hence, the HoloScope is more flexible than tensor-based methods, which use a globally fixed bin size. Moreover, the HoloScope can update the time series at a low cost when T is increasing.

To consider the burst and drop patterns described in Trait 3, we need to decide the start point of a burst and the end point of a drop in time series $\mathcal{T}$. Let the burst point be (t_m, c_m), having the maximum value c_m. According to the definition in the previous "Sleeping Beauty" work, we use an auxiliary straight line from the beginning to the burst point to decide the start point, named the *awakening point* of the burst. Fig. 5b shows the time series $\mathcal{T}$ (red polygonal line) of a message from Sina Weibo, the auxiliary straight line l (black dotted line) from the lower left point (t_0, c_0) to the upper right point (t_m, c_m), and the awakening point for the maximum point (t_m, c_m), which is defined as the point along the time series $\mathcal{T}$ which maximizes the distance to l. As the dotted line perpendicular to l suggests in this figure, the awakening point (t_a, c_a) satisfies

$$t_a = \underset{(c,t)\in\mathcal{T}, t<t_m}{\arg\max} \frac{|(c_m - c_0)t - (t_m - t_0)c + t_m c_0 - c_m t_0|}{\sqrt{(c_m - c_0)^2 + (t_m - t_0)^2}}. \quad (5)$$

Finding the awakening point for one burst is not enough, as multiple bursts may be present. Thus, sub-burst points and the associated awakening points should be considered. The algorithm is summarized in Algorithm 1. The *MultiBurst* algorithm detects other bursts by recursively searching the left and right parts of time series after removing the current burst period (t_a, t_m), until no time series are left. Finally, a list of bursts including awakening and burst time, slops, and altitude differences are returned by the algorithm.

Thus the contrast suspiciousness of burst awareness satisfies $P(v_i|A) \propto q(\varphi_i)$, where defining φ_i as *the involvement ratio of source nodes from A in multiple bursts*. Let the collection of timestamps from A to sink node v_i be T_A. Then,

$$\varphi_i = \frac{\Phi(T_A)}{\Phi(T_U)}, \text{ and } \Phi(T) = \sum_{(t_a,t_m)} \Delta c_{am} \cdot s_{am} \sum_{t\in T} \mathbf{1}(t \in [t_a, t_m]), \quad (6)$$

where s_{am} is the slope from the output of the $MultiBurst$ algorithm. Here s_{am} is used as a weight based on how steep the current burst is. The key point is that $\varphi_i \in [0, 1]$ measures how many of the total users involved in the bursts of an object are from the candidate set A. Considering the framework of dense block detection, only those common users that are always and synchronously involved in steep bursts of many objects can be detected as fraudsters, where the fraudulent attacking satisfies Trait 3. In another word, normal users that are diversely involved in the burst of popular objects will not be detected as false positive cases with our contrast suspiciousness of burst.

Algorithm 1. $MultiBurst$ Algorithm

Input Time series $\mathcal{T}$ of sink node v, beginning index i, end index j
Output A list of awakening-burst point pairs,
s_{am}: slope of the line passing through each point pair,
Δc_{am}: altitude difference of each point pair.
If $j - i < 1$ **then return**
(t_m, c_m) = point of maximum altitude in $\mathcal{T}_{ij}$.
(t_a, c_a) = find awakening point as Eq. (5) in $\mathcal{T}_{ij}$.
$\Delta c_{am} = c_m - c_a$, and $s_{am} = {}^{\Delta c_{am}}/_{(t_m - t_a)}$
Append $\{(t_a, c_a), (t_m, c_m)\}$, s_{am}, and Δc_{am} into the output.
$MultiBurst(\mathcal{T}, i, a-1)$
k = Find the first local min position from indices $m+1$ to j
$MultiBurst(\mathcal{T}, k, j)$

In fact, sudden drops are also a prominent pattern of fraudulent behavior as described in Trait 3, since after creating the attack is complete, fraudsters usually stop their activity suddenly. To make use of the suspicious pattern of a sudden drop, we define the *dying point* as the end of a drop. As Fig. 5b suggests, another auxiliary straight line is drawn from the highest point (t_m, c_m) to the last point (t_e, c_e). The dying point (t_d, c_d) can be found by maximizing the distance to this straight line. Thus we can discover the "sudden drop" by the absolute slope value $s_{bd}={}^{(c_m - c_d)}/_{(t_d - t_m)}$ between the burst point and the dying point. Since there may be several drops in a fluctuated time series $\mathcal{T}$, we choose the drop with the maximum fall, as Algorithm 2.

Algorithm 2. $MaxDrop$ Algorithm

Input Time series $\mathcal{T}$ of sink node v, beginning index i, end index j
Output Δc_{bd}: altitude difference of the point pair,
s_{bd}: slope of the line passing through the point pair.
If $j - i < 1$ **then return**
(t_m, c_m) = point of maximum altitude in $\mathcal{T}_{ij}$.
(t_d, c_d) = find dying point in $\mathcal{T}_{ij}$ by definition.
$\Delta c_{bd} = c_m - c_d$, and $s_{bd} = {}^{\Delta c_{bd}}/_{(t_d - t_m)}$
If Δc_{bd} is larger than the drop found in the output **then**
Overwrite output with Δc_{bd} and s_{bd}.
$MaxDrop(\mathcal{T}, i, m-1)$
$MaxDrop(\mathcal{T}, d, j)$

$MaxDrop$ algorithm detects the next possible drops by recursively searching in the left and right part of time series after removing the current drop period ($t_m and t_d$), until no time series are left. As a result, $MaxDrop$ algorithm returns the maximum drop fall Δc_{bd}, and its drop slope s_{bd} among those detected drops. Such a sudden drop of created edges to an object is unusual and indicates the suspiciousness of the object. Therefore, we use the weighted drop slope as a measure of global suspiciousness in Equation (2), namely,

$$\sigma = \Delta c_{bd} \cdot s_{bd}, \tag{7}$$

for every edge connected to the object v_i, and we omit the subscripts of edges for simplicity.

With this approach of detecting bursts and drops, we now show that this provides a provable time obstruction for fraudsters.

Theorem 1. *Let N be the number of edges that fraudsters want to create for an object. If the fraudsters use an amount of time less than $\tau \geq \sqrt{\frac{2N\Delta t(S_1+S_2)}{S_1 \cdot S_2}}$, then they will be tracked by a suspicious burst or drop, where Δt is the size of time bins and S_1 and S_2 are the slopes of normal rise and decline, respectively.*

Proof. The most efficient way to create N edges is to have one burst and one drop; otherwise, more time is needed. As shown in Fig. 5c, to minimize the slope, every point in the time series should be in line with the two auxiliary straight lines to the highest point c_m, separately from the awakening and dying points. Otherwise, the slopes will exceed the normal values S_1 and S_2. Hence, we consider only the triangle with the auxiliary lines as its two edges. Note that a trapezoid whose legs have the same slopes as the triangle's edges cannot have a shorter time cost. Then

$$\frac{c_m}{n_1 \Delta t} = S_1, \qquad \frac{c_m}{n_2 \Delta t} = S_2, \qquad (n_1 + n_2) \cdot c_m = 2N'.$$

Here n_1 and n_2 are the number of time bins before and after the burst. N' is the total number of rating edges, and $N' \geq N$ consider some edges from normal users. Thus, solving the above equations, we have

$$\tau = (n_1 + n_2)\Delta t = \sqrt{\frac{2N'\Delta t(S_1 + S_2)}{S_1 \cdot S_2}} \geq \sqrt{\frac{2N\Delta t(S_1 + S_2)}{S_1 \cdot S_2}}.$$

□

We also have the height of the burst $c_m \geq \sqrt{\frac{2N\Delta t S_1 S_2}{S_1 + S_2}}$. Thus, the maximum height of time series $\mathcal{T}$ cannot be larger by far than that of a normal sink node. That is why we use the altitude differences Δc_{am} and Δc_{bd} separately in Equations (6) and (7) as weights.

3.1.3 Rating Deviation and Aggregation

We now consider edges with categorical attributes such as rating scores, text contents, etc. For each sink node v_i, we use the KL-divergence κ_i of the rating distribution from the suspicious source nodes A given that from the other nodes, i.e., $U \backslash A$.

$$\kappa_i = \sum_{k \leq K} F_k(v_i) \log \frac{F_k(v_i)}{F'_k(v_i)},$$

where K is the number of categories, e.g., different rating scores, words or topics. Let $F_k(v_i)$ and $F'_k(v_i)$ are the frequencies for the kth category separately from users in A and $U \setminus A$ to object v_i. To avoid being divided by a zero

frequency, we add 1 to each category as a multinomial prior. Here, we use the distribution from other nodes $U \setminus A$, as opposite to that from the whole source nodes U, aiming to avoid a trivial case where most of the rating scores are from A. The rating deviation κ_i is scaled into $[0, 1]$ by the maximum value before being passed into function $q(\cdot)$ to compute contrast suspiciousness. Notice that rating deviation is meaningful when both A and $U \setminus A$ have comparable numbers of ratings. Thus, we weighted κ_i by a balance factor, i.e.,

$$\kappa_i = \min\left\{\frac{f_A(v_i)}{f_{U\setminus A}(v_i)}, \frac{f_{U\setminus A}(v_i)}{f_A(v_i)}\right\} \cdot \kappa_i.$$

Moreover, the neutral scores can be ignored in the KL-divergence for the purpose of detecting fraudulent boosting or defamation.

To make holistic use of different signals, i.e., topology, temporal spikes, and rating deviation, we need a way to aggregate those signals together. As far as we know, there are few approaches that can be used for aggregation in an unsupervised framework. We have tried to use RRF (reciprocal rank fusion) scores from Information Retrieval. Compared to the RRF score, we found that joint probability, i.e., multiplying those signals together, was the most effective way to aggregate

$$P(v_i|A) = q(\alpha_i)q(\varphi_i)q(\kappa_i) = b^{\alpha_i+\varphi_i+\kappa_i-3}, \tag{8}$$

In this way, we can consider the absolute suspiciousness value of each signal, as opposed to only using ranking order. Moreover, being wrapped with $q(x)$, the signal values cannot be canceled out by multiplying a very small value from other signals. As a concrete example, a suspicious spike can retain a high suspiciousness score by multiplying a very small score from low fraudulent density.

3.2 Algorithm and Time Complexity

Before designing the full algorithm for large scale datasets, we first introduce the most important sub-procedure, *GreedyShaving*, in Algorithm 3.

Algorithm 3. *GreedyShaving* Procedure

Given bipartite multigraph $\mathcal{G}(U, V, E)$,
  initial source nodes $A_0 \subset U$.
Output suspect source nodes $A^* \subset U$,
  target sink probabilities P^*.
Initialize:
  $A = A_0$
  $\mathcal{P}$= calculate contrast suspiciousness given A_0
$\mathcal{S}$ = calculate suspiciousness scores of source nodes A.
MT = build priority tree of A with scores $\mathcal{S}$.
while A is not empty **do**
  u = pop the source node of the minimum score from MT.
  $A = A \setminus u$, delete u from A.
  Update $\mathcal{P}$ with respect to new source nodes A.
    {(takes $O(d_u \cdot (|A| - 1))$ time)}
  Update MT with respect to new $\mathcal{P}$.
  {(takes $O(d_u \cdot (|A| - 1)\log|A_0|)$ time)}
  A^* = source nodes A that has the best objective $HS(A^*)$ so far.
end while
return A^* and $P(v|A^*)$, $v \in V$.

At the beginning, this greedy shaving procedure starts with an initial set $A_0 \subset U$ as input. It then greedily deletes source nodes from A, according to users' scores $\mathcal{S}$

$$\mathcal{S}(u_j \in A) = \sum_{v_i:(u_j,v_i)\in E} \sigma ji \cdot e_{ji} \cdot P(v_i|A),$$

which can be interpreted as how many suspicious nodes that user u_j is involved in. So the user is less suspicious if having a smaller score, with respect to the current contrast suspiciousness $\mathcal{P}$, where we use $\mathcal{P}$ to denote a vector of contrast suspiciousness of all sink nodes. We build a priority tree to help us efficiently find the user with the minimum score. The priority tree updates the users' scores and maintains the new minimum as the priorities change. With the removal of a source node in A, the contrast suspiciousness $\mathcal{P}$ changes, after which we then update users' scores $\mathcal{S}$. The algorithm keeps reducing A until it is empty. The best A^* maximizing objective HS and $P(v|A^*)$ are returned at the end.

The awakening and burst points only need to be detected once for each sink node by *MultiBurst* algorithm (see Algorithm 1), which is linear of the time series size for sink node v, that is, $O(d_v)$, where d_v is the degree of v. Then for all sink nodes, the detection cost is $O(|E|)$, which is considered as the initial step before the *GreedyShaving* procedure. Thus the calculation of the contrast suspicious $P(v|A)$ for a sink node v needs only $O(|A|)$ amount of time. With source node j as the jth one removed from A_0 by the *GreedyShaving* procedure, $|A_0| = m_0$, and the out-degree as d_i, the complexity is

$$\sum_{j=2,\ldots,m_0} O(d_j \cdot (j-1) \cdot \log m_0) = O(m_0|E_0|\log m_0), \tag{9}$$

where E_0 is the set of edges connected to source nodes A_0.

With the *GreedyShaving* procedure, our scalable algorithm can be designed so as to generate candidate suspicious source node sets. In our implementation, we use singular vector decomposition for our algorithm. Each top singular vector gives a spectral view of high connectivity communities. However, those singular vectors are not associated with suspiciousness scores. Thus combined with the top singular vectors, our fast greedy algorithm is given in Algorithm 4.

Algorithm 4. *FastGreedy* Algorithm for Fraud Detection

Given bipartite multigraph $\mathcal{G}(U, V, E)$.
Output suspect source nodes $A^* \subset U$, and the probabilities of being target sinks, P^*.
$\mathbb{L}$ = get first several left singular vectors
for all $L^{(k)} \in \mathbb{L}$ **do**
  Rank source nodes U decreasingly on $L^{(k)}$
  $\tilde{U}^{(k)}$ = truncate $u \in U$ when $L^{(k)}_u \leq \frac{1}{\sqrt{|U|}}$
  GreedyShaving with initial $\tilde{U}^{(k)}$.
end for
return the best A^* with maximized objective $HS(A^*)$,
  and the rank of $v \in V$ by $f_{A^*}(v) \cdot P(v|A^*)$.

Theorem 2 (Algorithm complexity). *In the graph $\mathcal{G}(U, V, E)$, given $|V| = O(|U|)$ and $|E| = O(|U|^{\epsilon_0})$, the complexity of FastGreedy algorithm is subquadratic, i.e., $o(|U|^2)$ in little-o notation, if the size of truncated user set $|\tilde{U}^{(k)}| \leq |U|^{1/\epsilon}$, where $\epsilon > \max\{1.5, \frac{2}{3-\epsilon_0}\}$.*

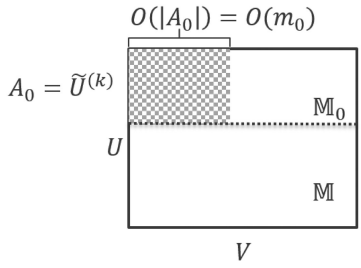


Fig. 6. Auxiliary for the proof of algorithm complexity.

Proof. The *FastGreedy* algorithm executes *GreedyShaving* in constant iterations. A_0 is assigned to $\tilde{U}^{(k)}$ in every *GreedyShaving* procedure. Then $m_0 = |A_0| = |\tilde{U}^{(k)}|$. In the adjacency matrix $\mathbb{M}$ of the graph, we consider the submatrix $\mathbb{M}_0$ with A_0 as rows and V as columns in Fig. 6. The worst case of computing complexity is that the fraudulent dense block is in submatrix $\mathbb{M}_0$. Thus we assume that the fraudulent dense block has at most m_0 rows and $O(m_0)$ columns. Excluding the dense block, the remaining part of $\mathbb{M}_0$ is assumed to have the same density with the whole matrix $\mathbb{M}$. Therefore, the total number of edges in $\mathbb{M}_0$ is

$$\begin{aligned} O(|E_0|) &= O\left(m_0^2 + \frac{m_0 \cdot |E| \cdot (|V| - m_0)}{|U| \cdot |V|}\right) \\ &= O\left(m_0^2 + \frac{m_0 \cdot |E|}{|U|}\right) \\ &= O(|U|^{2/\epsilon} + |U|^{1/\epsilon - 1}|E|). \end{aligned} \tag{10}$$

Then based on Equation (9), the algorithm complexity is

$$O(m_0|E_0|\log m_0) = O((|U|^{3/\epsilon} + |U|^{2/\epsilon - 1 + \epsilon_0})\log|U|).$$

Therefore, if $\epsilon > \max\{1.5, \frac{2}{3-\epsilon_0}\}$, then the complexity is subquadratic $o(|U|^2)$. □

In a real-life graph, $\epsilon_0 \leq 1.6$, so if $\epsilon > 1.5$ the complexity of the *FastGreedy* algorithm is subquadratic. Therefore, without loss of performance and efficiency, we can limit $|\tilde{U}^{(k)}| \leq |U|^{1/1.6}$ to truncate an ordered U in the *FastGreedy* algorithm for a large dataset. The time complexity is then $O(|U|^{1.875})$.

In the *FastGreedy* algorithm for HS-α, SVD on adjacency matrix $\mathbb{M}$ is used to generate initial blocks for the *GreedyShaving* procedure. Although we can still use SVD on matrix $\mathbb{M}$ for HS on holistic attributes, nevertheless considering attributes of timestamps and rating scores may bring more benefits. As we know, a multi-mode tensor is an intuitive way to represent a graph with multiple edge attributes, i.e., bucketized timestamps (time bins), and # of stars are another two modes besides users and products. Fig. 7 illustrates a tensor with three modes. Observing that not every combination of # of stars, time bins and products has a value in the tenor representation, we hence can only choose every existing triplets (*object*, *timestamp*, *#stars*) as one column and each *user* as a row, to form a new matrix. The above transformation is called the *matricization* of a tensor, which outputs a new matrix as the figure shows. With proper time bins, e.g., one hour or day, and with re-clustering of *#stars*, the flattening matrix becomes more dense and contains more attribute information. Therefore, we use such a flattened

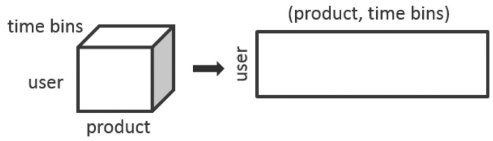


Fig. 7. An example of a tensor matricization to a matrix.

matrix with each column weighted by the sudden-drop suspiciousness of the corresponding product for users' singular vectors $L_{(k)}$ in the *FastGreedy* algorithm.

As an alternative, K-core can also be used as the start of the *GreedyShaving* procedure, which is also reported as a complementary success of SVD [23].

3.3 Theoretical Analysis and Scaling Base

We will now show how the scaling base b of $q(\cdot)$ affects our algorithm and objective function. In a step of the *GreedyShaving* algorithm, we assume $A' = A \setminus \{u_j\}$ by shaving a source u. Let $\tilde{HS}(A) = \sum_{i \in V} f_A(v_i)P(v_i|A)$, i.e., the numerator of $HS(A)$. To simplify, we let global the suspiciousness $\sigma_{ji} = 1$ for all edges, and denote $P(v_i|A)$ and $P(v_i|A')$ as P_i and P'_i respectively. Then

$$\tilde{HS}(A') = \tilde{HS}(A \setminus u_j) = \sum_{i \in V}(f_A(v_i) - e_{ji})P'_i.$$

And

$$\begin{aligned} \Delta\tilde{HS} &= \tilde{HS}(A) - \tilde{HS}(A') \\ &= \sum_{l:(j,l)\in E j':j'\in A \wedge j' \neq j \wedge (j',l)\in E} e_{j'l}\Delta P_l + \sum_{l:(j,l)\in E} e_{jl}P_l, \end{aligned}$$

where $\Delta P_l = P_l - P'_l$.

Denoting $\Delta\tilde{HS}/(\sum_l \Delta P_l + 1)$ as ω_j, we have the following theorem:

Theorem 3. *Given that $A' = A \setminus \{u_j\}$, the objective value increases, i.e., $HS(A') > HS(A)$ if and only if user u_j has smaller ω_j than $HS(A)$.*

Proof. Since

$$HS(A') = \frac{\tilde{HS}(A) - \Delta\tilde{HS}}{|A| - 1 + \sum_k P'_k} > HS(A) = \frac{\tilde{HS}(A)}{|A| + \sum_k P_k},$$

we can derive that

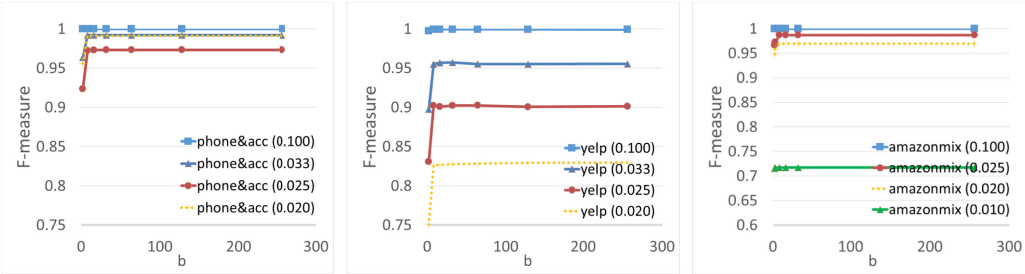
$$HS(A) > \frac{\Delta\tilde{HS}}{\sum_l \Delta P_l + 1} = \omega_j.$$

This proves the necessity half of the theorem. We can also derive that $HS(A') > HS(A)$ based on the above inequality, which proves the sufficiency part of the theorem. □

Corollary 1. *All users u_j from the optimal source nodes A^*, satisfy $\omega_j \geq HS(A^*)$.*

This corollary holds since if $\omega_j < HS(A^*)$, we can use Theorem 3 to increase $HS(A^*)$, which contradicts the fact that $HS(A^*)$ is the maximum solution. We know that

$$\omega_j = \frac{\sum_{l,j'\neq j} e_{j'l}\Delta P_l + \sum_l e_{jl}P_l}{\sum_l \Delta P_l + 1}.$$



(a) Amazon Phone & Acc (b) Yelp (c) Amazon mix category

Fig. 8. HS is not sensitive to scaling base b in some interval.

Let's consider the HoloScope HS-α on connection graphs as an example. For the users who have the same ΔP_l and the same degree, the user u_j is likely to be shaved, if connecting to the sinks that few users in A connect to, i.e., smaller involving ratio $P_l = b^{\alpha_l - 1}$ from A. The larger base b can make the interval growing quickly between high and low involving ratios, so that the user with the lower involving ratio is easily to be shaved. More importantly, we have $\Delta P_l = b^{\alpha_l - 1} - b^{\alpha_l - \Delta\alpha_l - 1} = (1 - b^{-\Delta\alpha_l})P_l$. Therefore, when the users u_j have the same degree, and the same $\Delta\alpha_l$ for each connected sink node l, a larger b offers the shaving algorithm more bias toward keeping the users who cover the sinks that have larger α_l, while shaving those who cover the less involved sinks.

In fact, let the derivative of P_l be no less than 1. We conclude that when $\alpha_l \geq 1 - \frac{\ln \ln b}{\ln b}$, the scaling function $q(\cdot)$ amplifies the difference $\Delta\alpha_l$. Thus, small b will shrink $\Delta\alpha_l$, while large b helps amplify the difference from a smaller α_l.

4 Experiments

In our experiments, we only consider the significant multiple bursts for fluctuated time series of sink nodes. We keep those awakening-burst point pairs with the altitude difference Δc more than half of the largest altitude difference in time series. Table 3 gives the statistics of our six datasets which are publicly available for academic research, including BeerAdvocate data [42], Yelp data,[3] Amazon review data in categories [43], and Amazon reviews with mixed categories [44].

4.1 Parameter Sensitivity

As we have theoretically analyzed the scaling base b in function $q(x)$, we test the sensitivity of b on two datasets with different fraudulent density by choosing b from 2 to 256. The results of the F-measure on users are shown in Fig. 8. The fraudulent density is given after the name of data in a bracket. The curves of "phone&acc (0.100)" and "yelp (0.100)" are overlapped in the figure, since they have almost the same F-measure. The AUCs on objects are all 1.0 for those testing b. Therefore, generally speaking, our performance is not sensitive to b, as the empirical results shows that around 5 percent difference between the choices of b. As b increases, the performance quickly becomes stable after $b = 8$, which agrees with the theoretical analysis that larger b helps in shaving false positive users. We suggest that scaling base $b \in [8, 32]$ in practice, since a very large b requires an extremely high ratio of fraudulent connections to objects, which may fail to detect those suspicious users if some normal users have the similar connection. In the following experiments, we choose $b = 32$.

We also compare the different forms of scaling function $q(x)$, and we run the results on the BeerAdvocate dataset with fraudulent density 0.1. As shown in Table 2, the RRF scores [39] is computed

$$RRF(v|A) = \sum_{r \in R} \frac{1}{k + r(v|A)},$$

where $r(v|A)$ is the order of v in a rank list by contrast suspiciousness $P(v|A)$. R is a collection of rank lists obtained by all suspiciousness values of holistic signals. The constant k is essentially used to harmonize the differences among the top ranks. For example, if $k = 0$, the distance between ranks 1 and 2 is 0.5, while the rest of infinity items sharing the residual 0.5. So we choose $k = 60$ as [39] suggested, implying that the top 60 items share the same distance with the rest. As for "POW", we use the power form as scaling function

$$q(v|A) = (\alpha \cdot \varphi \cdot \kappa)^b.$$

We tried $b = 2, 4, 8$ and all the parameters had lower F-measures, so we only list $b = 2$ in the table.

As we can see, the joint probability of the exponential scaling function (8) performs in a promising manner. RRF throws away the suspiciousness scores, which provides helpful information in fraud detection. The product of different

TABLE 2
Performances of Different Scaling Functions

	source nodes			sink nodes
	F-measure	Precision	Recall	AUC
RRF [39]	0.0418	0.0749	0.0290	0.7849
POW	0.1166	0.1116	0.1220	0.5245
EXP	1.0000	1.0000	1.0000	1.0000

TABLE 3
Data Statistics

Data Name	#nodes	#edges	time span
BeerAdvocate [42]	26.5 K × 50.8 K	1.07 M	Jan 08-Nov 11
Yelp	686 K × 85.3 K	2.68 M	Oct 04-Jul 16
Amazon Phone & Acc [43]	2.26 M × 329 K	3.45 M	Jan 07-Jul 14
Amazon Electronics [43]	4.20 M × 476 K	7.82 M	Dec 98-Jul 14
Amazon Grocery [43]	763 K × 165 K	1.29 M	Jan 07-Jul 14
Amazon mix category [44]	1.08 M × 726 K	2.72 M	Jan 04-Jun 06

3. The Yelp dataset is from https://www.yelp.com/dataset_challenge

TABLE 4
Experimental Results on Real Data with Injected Labels

Data Name	metrics*	source nodes				sink nodes			
		M-Zoom	D-Cube	CrossSpot	HS	M-Zoom	D-Cube	CrossSpot	HS
BeerAdvocate	auc	0.7280	0.7353	0.2259	**0.9758**	0.6221	0.6454	0.1295	**0.9945**
	F≥90%	0.5000	0.5000	–	**0.0333**	0.5000	0.5000	–	**0.0333**
Yelp	auc	0.9019	0.9137	0.9916	**0.9925**	0.9709	0.8863	0.0415	**0.9950**
	F≥90%	0.2500	0.2000	0.0200	**0.0143**	0.0250	1.0000	–	**0.0100**
Amazon	auc	0.9246	0.8042	0.0169	**0.9691**	0.9279	0.8810	0.0515	**0.9823**
Phone & Acc	F≥90%	0.1667	0.5000	–	**0.0200**†	0.1429	0.1000	–	**0.0200**†
Amazon	auc	0.9141	0.9117	0.0009	**0.9250**	0.9142	0.7868	0.0301	**0.9385**
Electronics	F≥90%	0.2000	0.1250	–	**0.1000**	**0.1000**	0.5000	–	0.1250
Amazon	auc	0.8998	0.8428	0.0058	**0.9250**	0.8756	0.8241	0.0200	**0.9621**
Grocery	F≥90%	0.1667	0.5000	–	**0.1000**	0.1250	0.2500	–	**0.1000**
Amazon	auc	0.9001	0.8490	0.5747	**0.9922**	0.9937	0.9346	0.0157	**0.9950**
mix category	F≥90%	0.2500	0.5000	0.2000†	**0.0167**	**0.0100**	0.2000	–	**0.0100**

* *We use the two metrics: the area under the curve (abbrev as lower-case "auc") of the accuracy curve as drawn in Fig. 1b, and the lowest* detection density *that the method can detect with high accuracy (F measure* $\geq 90\%$*).*
† *One of the above fraudulent densities not detected with high accuracy.*

suspiciousness cores in POW, may cancel each other, if some score is not striking and has a very low value closed to 0.

4.2 Evaluation of Different Injection Density

In these experiments, we mimic the fraudsters' behaviors and randomly choose 200 objects with in-degree no more than 100 as the fraudsters' customers, since less popular objects are more susceptible to manipulation. Since all user accounts have a risk to be hijacked as fraudulent accounts, we then uniformly choose a number of users as controlled accounts by fraudsters from the whole user set. We randomly inject $200 \times 200 = 40,000$ fraudulent edges between the fraudulent accounts and the 200 target objects, ensuring that each target object receives 200 ratings as imaginary customers required. And to make the fraudulent accounts behave as the ordinaries, the camouflage edges are also randomly injected to the other objects biased to the popular ones, with the same amount of fraudulent edges [6].

To test on different fraudulent densities, the number of fraudulent accounts are from 200 to 20,000. As a results, the fraudulent density ranges from 1.0 to 0.01 for testing.

To mimic the surge of fraudsters' attacks, we generate the rating time for each fraudulent edge. First, find the earliest time t_0 and latest time t_e of the original edges in data, and collect the intervals of edge creation time for an object, as set $\mathcal{C}$. Second, randomly choose a start time t between t_0 and t_e. Third, add to t a randomly and biased time interval sampled from $\mathcal{C}$, and assign it to a fraudulent edge. Shorter intervals have higher probability, since fraudsters would like to attacks in a high frequency.

In addition, a biased rating score, e.g., 4 or 4.5, is randomly chosen for a fraudulent edge.[4]

Fig. 1a shows the results of HS-α on the BeerAdvocate data, compared with Fraudar and SpokEn which consider only topology information as HS-α does. To detect fraudsters of a low density is much harder than those of a high density, so the better methods are able to detect fraudsters of lower density with high accuracy. Since HS-α considers only the topology information in our novel contrast suspiciousness, we compare HS-α with the baselines based on graph topology. When the fraudulent density decreases from the right to the left along the horizontal axis, HS-α can detect fraudulent density as low as 0.125 in a high F-measure, much better than 0.8 which is the best of the baselines.

Fig. 1b shows the results of HoloScope HS, which uses topology, temporal, and rating attributes. Compared to those baselines on the same kinds of attributes, HS can keep as high an F-measure as more than 90 percent of them before reaching 0.033 in density, better by far than the baseline methods (0.50 in density). In other words, if the same amount of fraudulent edges is present, HS can detect fraud with high accuracy even when fraudsters use 6,000 source nodes (user accounts). On the other hand, the best of the baselines detects fraud with a low level of accuracy (less than 50 percent), even when only 600 fraudulent accounts attacked, which is easier to detect by HS. Besides, HS using several signals further improves over HS-α with only the topology signal (compared with Fig. 1a) by decreasing density from 0.125 to 0.033 with high detection accuracy.

We compare the baselines on all six datasets with different injection density, and propose to use two metrics: a lower-case "auc" and the lowest *detection density*, as described in the notes of Table 4. The table reports the fraud detection results of our HoloScope and the baselines on the six datasets. Since the accuracy curve stops at 0.01 (the minimum testing density), and since we add zero accuracy at zero density, the ideal value of auc is 0.995. The auc on source and sink nodes are reported separately. In terms of sink nodes, HS outputs the rank list by suspiciousness scores. We use the area under the upper-case AUC (similar to F-measure) accuracy curve along all testing density. As the table suggests, our HS achieved the best auc among the baselines, and it even reached the ideal auc in two cases.

Furthermore, we compare the lowest *detection density* in Table 4. The better a method is, the lower density it should be able to detect well. As we can see, HS has the smallest detection density in most cases, which can be as small as $^{200}/_{14000} = 0.0143$ on source nodes, and it reached the minimum testing density of 0.01 on sink nodes. That means that

4. The injection code is also open-sourced for reproducibility: https://github.com/shenghua-liu/HoloScope

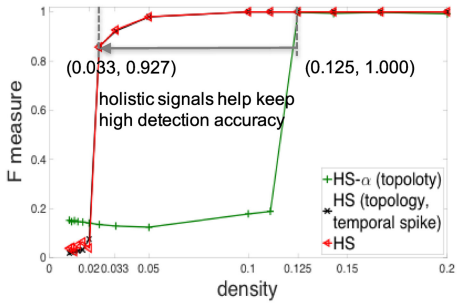


Fig. 9. HS with holistic signals has high detection accuracy for the low fraudulent density. HS with two signals achieves very close performance, but missing 30 of 6,000 fraudulent accounts. Default b is used.

we can detect fraudsters with high accuracy even when they use 14,000 accounts to create fake edges for each of 200 objects, due to the holistic use of signals in the contrast suspiciousness framework. The fraudulent objects can also be detected accurately.

4.3 Evaluation on Sina Weibo with Real Labels

We also performed experiments on a large real dataset from Sina Weibo, which had 2.75 million users, 8.08 million messages, and 50.1 million edges in Dec 2013. The user names and IDs, and the message IDs are from the online system. Thus we can check their existence status in the system to evaluate the experiments. If the messages or the users were deleted from the system, we treat them as the basis for identifying suspicious users and messages. It is impossible to check all of the users and messages, so we first collected a candidate set, which is the union of the output sets from the HS and the baseline methods. The real labels are examined in the candidate set by checking their status as to whether they still existed in Sina Weibo (checked in February 2017, i.e. more than 3 years later). If the users or messages cannot be accessed, then we label them as suspicious nodes, since they were probably removed by the operation team from Sina Weibo. We used a program on the API service of Sina Weibo to check the candidate user and message ID lists, resulting in 3,957 labeled users and 1,615 labeled messages.

The experimental results in Fig. 1c show that HS achieved a high F-measure on accuracy, which detected 3,781 labeled users, higher than M-Zoom's 1,963 labeled users. The F-measure of HS was about 30 and 60 percent higher than M-Zoom and D-Cube, respectively. CrossSpot was biased to include a large amount of users ($> 500,000$) in its detection results, which recalled fewer than 150 extra labeled users, getting a very low F-measure, less than 1.5 percent. In terms of messages, HS achieved around 0.8 in AUC from the ranking list of the results, while M-Zoom and D-Cube attained lower recall and CrossSpot still suffered from a very low F-measure with many false positive messages. Therefore, our HoloScope outperformed the baselines on real-labeled data as well.

4.4 Effectiveness of Combining Multiple Signals

Fig. 9 illustrates the results of HS-α using only topology information, HS (topology, temporal spike) using topology and temporal signals, and HS using of three signals on BeerAdvocate data. Default scaling base b is used for the above three cases. We can see that HS improves the performance of HS-α by retaining a high detection accuracy (≥ 0.92) up to a very low fraudulent density (0.033), whereas HS-α results in a very low accuracy (< 0.20) when the density is no more than 0.111. Besides, HS combining two signals of topology and temporal spikes (black-star line) improves the performance close to the best one reported (HS, red-triangle line), yet missing 30 of 6,000 fraudsters. Therefore, with our uniform metric *contrast suspiciousness*, the combination of multiple signals improves the performance of fraud detection over using one signal or two signals.

4.5 Scalability

To verify the complexity, we choose two representative datasets: BeerAdvocate data, which has the highest volume density, and Amazon Electronics, which has the most edges. We truncated the BeerAdvocate data according to different time ranges, from Jan 2012, Jan 2011, ..., Oct 2004 to Nov 2011. We then ran our algorithm on each resulting dataset. The larger Amazon Electronics data was also tested, truncated by ranges from Dec 1998, Jan 2003, Jan 2005, ..., Mar and Jun 2014 to Jul 2014. Our algorithm was implemented in Python, and ran in a single node with 256G memory, and 26 Cores and 2.00 GHz Intel Xeon CPU E7. As shown in Fig. 2, the running time of our algorithm increased almost linearly with the number of the edges.

5 Discussion and Conclusion

We now address several questions about the HoloScope method.

Q1. *Could fraudsters slow down the rate of creating fraudulent edges to avoid spikes?* Although it is technically easy for fraudsters to reduce their pace to create edges, the total time cost may not be acceptable due to the requirement of a large amount of edges from fraudulent objects. For example, in Twitter, if a message is retweeted fewer than thousands or tens of thousands of times in one day, people may not pay attention to it. Moreover, if creating fake ratings takes a long time, the fraudsters' *operating costs* of hiring people and maintaining accounts and IPs will be dramatically increased. Therefore, if a fraud detection algorithm like HoloScope can provide a robust temporal obstruction for fraudulent edge creation, fraudsters may give up due to the high cost involved.

Q2. *Isn't there a danger of a post from an ordinary user that goes viral?* HoloScope is based on block detection; that is, a group of suspicious users are connected to a group of suspicious objects, instead of an individual post or user. Moreover, the novel idea of examining *contrast suspiciousness*, instead of the speed by which a post becomes popular or large degree, considers the ratio of suspicious users participating in bursting periods or the total degree. Contrast suspiciousness is dynamically calculated based on current suspicious users in Algorithm 3. An intuitive way to see the result is that HS-α detects objects with suspicious high degrees in the top left corner of Fig. 4a, and avoids the ordinary popular (high-degree) objects in the hyperbolic area.

Q3. *Is the algorithm sensitive to camouflage?* Generally speaking, our algorithm can reduce the effects of camouflage connections that are created to normal objects by fraudsters. Since the contrast suspiciousness $P(v|A)$ will be very low for popular objects, objective (3) will not be

decreased much by adding small numbers to both the numerator and the denominator of the objective. In addition, by connecting to a large number of low-popularity objects, our objective value may greatly decrease.

Moreover, in the future, we expect to see improvement of the performance, by introducing more properties into our HoloScope, such as the text of reviews, and even the psychological elements of deceptive reviews [45].

In conclusion, we have proposed a contrast metric for fraud detection in attributed and bipartite graphs. HoloScope has the following advantages: *1) Novel suspiciousness metric:* With contrast metric, our HoloScope emphasizes the contrast in behavior between fraudsters and honest users in terms of topology, temporal spikes, and rating deviation. *2) Robustness and theoretical analysis of fraudsters' obstruction:* We showed that if the fraudsters use less than a lower bound of time to rate an object, they will cause a suspicious drop or burst. HoloScope can obstruct fraudsters and increases their time cost. *3) Effectiveness:* We achieved higher accuracy on both semi-real and real datasets than the baselines. *4) Scalability*. The algorithm is scalable and runs in near linear number of edges. At last, HoloScope is open-sourced.

Acknowledgments

This material is based upon work supported by the Strategic Priority Research Program of CAS (XDA19020400), National 973 Program of China (2014CB340401), NSF of China (61772498, 61872206), the Beijing NSF (4172059), and the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053. Shenghua Liu is also supported by a scholarship from the China Scholarship Council. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding parties. The work that led to this paper was performed while Shenghua Liu was a visiting researcher at Carnegie Mellon University

References

[1] W. Ops, "The methbot operation," 2016. [Online]. Available: http://go.whiteops.com/rs/179-SQE-823/images/WO_Methbot_Operation_WP.pdf, Accessed on: Jan. 7, 2017.

[2] N. N. Dalvi, P. M. Domingos, S. K. Sanghai, and D. Verma, "Adversarial classification," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2004, pp. 99–108.

[3] M. Charikar, "Greedy approximation algorithms for finding dense components in a graph," in *Proc. Int. Workshop Approximation Algorithms Combinatorial Optimization*, 2000, pp. 84–95.

[4] 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*, 2016, pp. 264–280.

[5] 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*, 2017, pp. 681–689.

[6] 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*, 2016, pp. 895–904.

[7] B. Hooi, K. Shin, H. A. Song, A. Beutel, N. Shah, and C. Faloutsos, "Graph-based fraud detection in the face of camouflage," *ACM Trans. Knowl. Discovery Data*, vol. 11, no. 4, pp. 44:1–44:26, 2017. [Online]. Available: http://doi.acm.org/10.1145/3056563

[8] K. Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval," *J. Documentation*, vol. 28, no. 1, pp. 11–21, 1972.

[9] S. Robertson, "Understanding inverse document frequency: On theoretical arguments for IDF," *J. Documentation*, vol. 60, no. 5, pp. 503–520, 2004.

[10] M. Jiang, A. Beutel, P. Cui, B. Hooi, S. Yang, and C. Faloutsos, "A general suspiciousness metric for dense blocks in multimodal data," in *Proc. IEEE Int. Conf. Data Mining*, 2015, pp. 781–786.

[11] S. Günnemann, N. Günnemann, and C. Faloutsos, "Detecting anomalies in dynamic rating data: A robust probabilistic model for rating evolution," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 841–850.

[12] 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.

[13] M. Giatsoglou, D. Chatzakou, N. Shah, C. Faloutsos, and A. Vakali, "Retweeting activity on twitter: Signs of deception," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*, 2015, pp. 122–134.

[14] M. Jiang, A. Beutel, P. Cui, B. Hooi, S. Yang, and C. Faloutsos, "Spotting suspicious behaviors in multimodal data: A general metric and algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 8, pp. 2187–2200, Aug. 2016.

[15] N. Shah, A. Beutel, B. Hooi, L. Akoglu, S. Gunnemann, D. Makhija, M. Kumar, and C. Faloutsos, "EdgeCentric: Anomaly detection in edge-attributed networks," *IEEE Int. Conf. Data Mining (ICDM) Workshop on Data Mining for Cyber Security*, 2016.

[16] 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*, 2010, pp. 435–448.

[17] 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.

[18] S. Pandit, D. H. Chau, S. Wang, and C. Faloutsos, "Netprobe: A fast and scalable system for fraud detection in online auction networks," in *Proc. 16th Int. Conf. World Wide Web*, 2007, pp. 201–210.

[19] 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. 2–11.

[20] M. Jiang, P. Cui, A. Beutel, C. Faloutsos, and S. Yang, "Inferring strange behavior from connectivity pattern in social networks," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*, 2014, pp. 126–138.

[21] L. Akoglu, M. McGlohon, and C. Faloutsos, "Oddball: Spotting anomalies in weighted graphs," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*, 2010, pp. 410–421.

[22] D. Gibson, R. Kumar, and A. Tomkins, "Discovering large dense subgraphs in massive graphs," in *Proc. 31st Int. Conf. Very Large Data Bases*, 2005, pp. 721–732.

[23] 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*, 2016, pp. 469–478.

[24] M. Araujo, S. Günnemann, G. Mateos, and C. Faloutsos, "Beyond blocks: Hyperbolic community detection," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2014, pp. 50–65.

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

[26] 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*, 2014, pp. 941–950.

[27] M. Jiang, P. Cui, A. Beutel, C. Faloutsos, and S. Yang, "Catching synchronized behaviors in large networks: A graph mining approach," *ACM Trans. Knowl. Discovery Data*, vol. 10, no. 4, pp. 35:1–35:27, 2016.

[28] E. Kirkos, C. Spathis, and Y. Manolopoulos, "Data mining techniques for the detection of fraudulent financial statements," *Expert Syst. Appl.*, vol. 32, no. 4, pp. 995–1003, 2007.

[29] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *Proc. Eur. Symp. Artif. Neural Netw. Comput. Intell. Mach. Learn.*, 2015, Art. no. 89.

[30] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins, "On the bursty evolution of blogspace," *World Wide Web*, vol. 8, no. 2, pp. 159–178, 2005.

[31] Q. Ke, E. Ferrara, F. Radicchi, and A. Flammini, "Defining and identifying sleeping beauties in science," *Proc. Nat. Academy Sci. United States America*, vol. 112, no. 24, pp. 7426–7431, 2015.
[32] J.-I. Takeuchi and K. Yamanishi, "A unifying framework for detecting outliers and change points from time series," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 4, pp. 482–492, Apr. 2006.
[33] J. Yang and J. Leskovec, "Patterns of temporal variation in online media," in *Proc. 4th ACM Int. Conf. Web Search Data Mining*, 2011, pp. 177–186.
[34] S. A. Myers and J. Leskovec, "The bursty dynamics of the twitter information network," in *Proc. 23rd Int. Conf. World Wide Web*, 2014, pp. 913–924.
[35] A. Paranjape, A. R. Benson, and J. Leskovec, "Motifs in temporal networks," in *Proc. 10th ACM Int. Conf. Web Search Data Mining*, 2017, pp. 601–610.
[36] N. Günnemann, S. Günnemann, and C. Faloutsos, "Robust multivariate autoregression for anomaly detection in dynamic product ratings," in *Proc. 23rd Int. Conf. World Wide Web*, 2014, pp. 361–372.
[37] H. Li, G. Fei, S. Wang, B. Liu, W. Shao, A. Mukherjee, and J. Shao, "Bimodal distribution and co-bursting in review spam detection," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 1063–1072.
[38] G. Fei, A. Mukherjee, B. Liu, M. Hsu, M. Castellanos, and R. Ghosh, "Exploiting burstiness in reviews for review spammer detection," in *Proc. 7th Int. AAAI Conf. Weblogs Social Media*, pp. 175–184, 2013.
[39] G. V. Cormack, C. L. Clarke, and S. Buettcher, "Reciprocal rank fusion outperforms condorcet and individual rank learning methods," in *Proc. 32nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2009, pp. 758–759.
[40] H. A. Sturges, "The choice of a class interval," *J. Amer. Statistical Assoc.*, vol. 21, no. 153, pp. 65–66, 1926.
[41] D. Freedman and P. Diaconis, "On the histogram as a density estimator: L2 theory," *Probability Theory Related Fields*, vol. 57, no. 4, pp. 453–476, 1981.
[42] J. J. McAuley and J. Leskovec, "From amateurs to connoisseurs: Modeling the evolution of user expertise through online reviews," in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 897–908.
[43] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: Understanding rating dimensions with review text," in *Proc. 7th ACM Conf. Recommender Syst.*, 2013, pp. 165–172.
[44] A. Mukherjee, B. Liu, and N. Glance, "Spotting fake reviewer groups in consumer reviews," in *Proc. 21st Int. Conf. World Wide Web*, 2012, pp. 191–200.
[45] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock, "Finding deceptive opinion spam by any stretch of the imagination," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics: Human Lang. Technol.-Volume 1*, 2011, pp. 309–319.

Shenghua Liu received the PhD degree from the Computer Science and Technology Department, Tsinghua University. He is an associate professor with the Institute of Computing Technology, Chinese Academy of Sciences. He was a visiting scholar with the University of California, Los Angeles and Carnegie Mellon University, respectively. His current research interests are designing intelligent and automated algorithms for big data mining problems, related to big graphs and series.

Bryan Hooi received the BS and MS degrees from Stanford University. He is working toward the PhD degree jointly enrolled in the Machine Learning Department and the Department of Statistics, Carnegie Mellon University. His main research interests include graph mining, anomaly detection, and time series data analysis.

Christos Faloutsos is a professor at Carnegie Mellon University. He received the Presidential Young Investigator Award from the National Science Foundation (1989), the Research Contributions Award in ICDM 2006, the SIGKDD Innovations Award (2010), 25 "best paper" awards (including five "test of time" awards), and four teaching awards. Eight of his advisees or co-advisees have received KDD or SCS dissertation awards. He is an ACM fellow and has served as a member of the executive committee of SIGKDD; he has published more than 350 refereed articles, 17 book chapters, and two monographs. He holds seven patents (with two pending), and he has given more than 40 tutorials and more than 20 invited distinguished lectures. His research interests include large-scale data mining with emphasis on graphs and time sequences, anomaly detection, tensors, and fractals.