Spotting Suspicious Link Behavior with fBox: An Adversarial Perspective

Neil Shah
Carnegie Mellon University
Pittsburgh, PA
neilshah@cs.cmu.edu

Alex Beutel
Carnegie Mellon University
Pittsburgh, PA
abeutel@cs.cmu.edu

Brian Gallagher
Lawrence Livermore Lab
Livermore, CA
bgallagher@llnl.gov

Christos Faloutsos
Carnegie Mellon University
Pittsburgh, PA
christos@cs.cmu.edu

***Abstract*—How can we detect suspicious users in large online networks? Online popularity of a user or product (via follows, page-likes, etc.) can be monetized on the premise of higher ad click-through rates or increased sales. Web services and social networks which incentivize popularity thus suffer from a major problem of fake connections from link fraudsters looking to make a quick buck. Typical methods of catching this suspicious behavior use spectral techniques to spot large groups of often blatantly fraudulent (but sometimes honest) users. However, small-scale, stealthy attacks may go unnoticed due to the nature of low-rank eigenanalysis used in practice.**

In this work, we take an adversarial approach to find and prove claims about the weaknesses of modern, state-of-the-art spectral methods and propose FBOX, an algorithm designed to catch small-scale, *stealth attacks* that slip below the radar. Our algorithm has the following desirable properties: (a) it has theoretical underpinnings, (b) it is shown to be highly effective on real data and (c) it is scalable (linear on the input size). We evaluate FBOX on a large, public 41.7 *million* node, 1.5 *billion* edge who-follows-whom social graph from Twitter in 2010 and with high precision identify many suspicious accounts which have persisted without suspension even to this day.

I. INTRODUCTION

In an online network, how can we distinguish honest users from deceptive ones? Since many online services rely on machine learning algorithms to recommend relevant content to their users, it is crucial to their performance that user feedback be legitimate and indicative of true interests. "Fake" links via the use of sockpuppet/bot accounts can enable arbitrary (frequently spammy or malicious) users and products of varying nature seem credible and popular, thus degrading the online experience of users. Unsurprisingly, numerous sites such as `buy1000followers.co`, `boostlikes.com` and `buyamazonreviews.com` exist to provide services such as fake Twitter followers, Facebook page-likes and Amazon product reviews for typically just a few dollars per one-thousand fake links.

Here we focus exactly on the link-fraud problem. We take an adversarial approach to illustrate when and how current methods fail to detect fraudsters and design a new complementary algorithm, FBOX, to spot attackers who evade these state-of-the-art techniques. Figure 1 showcases several suspicious accounts spotted by FBOX– we elaborate on three of them, marked using the triangle, square and star glyphs. All three are identified as outliers in the FBOX Spectral Reconstruction Map (SRM) shown in Figure 1b. The corresponding Twitter profiles are shown in Figure 1c, and further manual inspection shows that all three accounts exhibit suspicious behavior:

- triangle: it has only 2 tweets but over 1000 followers
- square: it is part of a 50-clique with suspicious names
- star: it posts tweets advertising a link fraud service

Our main contributions are the following:

1) **Theoretical analysis:** We prove limitations of the detection range of spectral-based methods.
2) **FBOX algorithm:** We introduce FBOX, a *scalable* method that *boxes-in* attackers, since it spots small-scale, stealth attacks which evade spectral methods.
3) **Effectiveness on real data:** We apply FBOX to a real, 41.7 *million* node, 1.5 *billion* edge Twitter who-follows-whom social graph from 2010 and identify many still-active accounts with suspicious follower/followee links, spammy Tweets and otherwise strange behavior.

Reproducibility: Our code is available at http://www.cs.cmu.edu/~neilshah/code/. The Twitter dataset is also publicly available as cited in [8].

II. BACKGROUND AND RELATED WORK

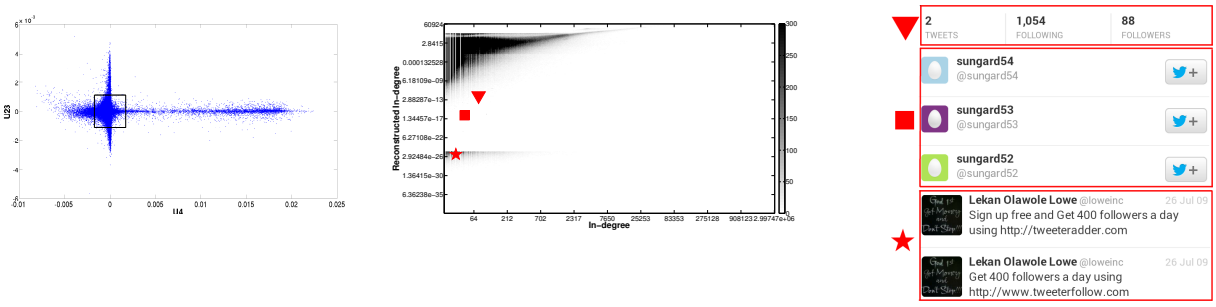
The related work forms three groups: spectral methods, graph traversal methods, and feature-based methods.

A. Spectral methods

We classify techniques that cluster the latent factors produced in graph-based spectral (eigendecomposition or singular value decomposition) analysis of the adjacency matrix as spectral methods. They include Prakash et al's work on the SpokEn algorithm for the EigenSpokes pattern [13] and Jiang et al's work on spectral subspaces of social networks [7].

These works both use the Singular Value Decomposition (SVD) of the input graph's adjacency matrix to group similar users and objects based on their projections. Recall that the SVD of a $u \times o$ matrix $\mathbf{A}$ is defined as $\mathbf{A} = \mathbf{U\Sigma V^T}$, where $\mathbf{U}$ and $\mathbf{V}$ are $u \times u$ and $o \times o$ matrices respectively, containing the left and right singular vectors, and $\mathbf{\Sigma}$ is a $u \times o$ diagonal matrix, containing the singular values of $\mathbf{A}$. Both papers try to spot unusual patterns and microclusters in the projected subspaces, retrieve the corresponding nodes and chip out communities of similar users.

Both methods have shown use in finding large dense communities, but as we show in Section III, they are likely to miss smaller and stealthier link-fraud attacks.



(a) Spectral subspace plot (b) Proposed ISRM (c) Suspicious accounts

Fig. 1: FBOX catches stealth attacks which are missed by spectral methods. (a) shows a spectral subspace plots on the Twitter social graph which identifies blatant attacks but ignores stealth attackers (at the origin). (b) portrays how the proposed FBOX ISRM (In-link Spectral Reconstruction Map) can describe these users by their *reconstruction degree* and identifies several with improbably poor reconstruction. (c) shows their suspicious profiles with matching glyphs (see text for details).

B. Graph-traversal based methods

Shrivastava et al [15] use random walks to detect random-link attacks. Ghosh et al [5] proposes a PageRank-like approach to penalizing promiscuous users on Twitter, Beutel et al propose the CopyCatch algorithm [2] which uses graph traversal to find lock-step behavior and thus dense bipartite cores of Facebook Page-Likes.

One major caveat with clustering methods is the non-trivial identification of appropriate minimal detectable attack parameters which must be finely tuned to avoid incurring false positives while catching most true negatives.

C. Feature-based methods

Spam and fraud detection has classically been framed as a feature-based classification problem, e.g. based on the words in spam email or URLs in tweets. However, [6] focuses on malicious Tweets and finds that blacklisting approaches are too slow to stem the spread of Twitter spam. OddBall [1] proposes features based on egonets to find anomalous users on weighted graphs. In [3] and [10] the authors take a game theoretic approach to learning simple classifiers over generic features to detect spam. While related in the adversarial perspective, these approaches focus on general feature-based classification as used for spam email, rather than graph analysis.

III. An Adversarial Analysis - Our Perspective

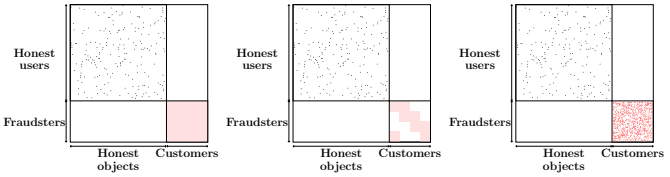
Here we examine the state-of-the-art methods from an adversarial point-of-view and present lemmas and theorems showing their vulnerability to intelligent attackers. Table I gives the list of symbols and definitions.

Formally, we pose the following adversarial problem:

Problem 1 (Evasion)**.**

- **Given:** *an input graph adjacency matrix* **A***, with rows/-columns corresponding to users/objects,*
- **Engineer** *a stealth attack*
- **to evade** *detection by modern spectral methods.*

Current methods can find large bipartite cores or cliques in the input graph, by studying the first k singular values and



(a) Naïve (b) Staircase (c) Random graph

Fig. 2: (a), (b) and (c) show the different types of adversarial attacks we characterize.

the corresponding subspaces. Therefore, if an adversary knows the minimum size attack that detection methods will catch, he can carefully engineer attacks to fall just below that threshold. For clustering approaches, this threshold is clearly set based on input parameters, and the attacker can simply use fewer accounts than specified to avoid detection.

However, for spectral methods like SpokEn, the possible attack size for an adversary is unclear. We argue that these spectral methods have a detection threshold based on the graph's singular values. For a rank k decomposition used in these methods, this threshold is governed by σ_k (kth largest singular value). An adversary could estimate σ_k from experimental attacks of varying sizes or by conducting analysis on publicly available data. Then, as we show below, the adversary can easily design attacks that will lie below this threshold, not change the top k singular values/vectors and thus avoid detection.

To analyze what type of attacks can evade detection by spectral methods, let us consider that there are c customers who have each bought s fraudulent actions (page likes, followers, etc.) from an attacker that has f fake accounts in his botnet, with $s \leq f$. More formally, the attacker can control the contents of an $f \times c$ submatrix $\mathbf{S}$ of the full adjacency matrix, where the f rows correspond to the f available attacker nodes and the c columns correspond to the customers, each of which should receive s links.

As mentioned earlier, an attack will only be detected by a spectral algorithm if it appears in the top k singular

TABLE I: Frequently used symbols and definitions

Symbol	Definition
u and o	Number of user and object nodes described by the input graph
$\mathbf{A}$	$u \times o$ input graph adjacency matrix where $\mathbf{A}_{x,y} = 1$ if a link exists between user node x and object node y
f and c	Number of attacker and customer nodes described by the attack graph
s	Number of fraudulent actions each customer node has paid commission for in the attack graph
$\mathbf{S}$	$f \times c$ attack graph adjacency matrix where $\mathbf{S}_{x,y} = 1$ if a link exists between attacker node x and customer node y
k	Decomposition rank parameter used by spectral methods
λ_k and σ_k	kth largest eigenvalue and singular value of a given matrix (largest values for $k = 1$)

values/vectors. Then, the goal of the attacker becomes to understand the spectral properties of his attack so that he can deliver on his promise with the customer without impacting the top k decomposition.

We find that the attacker can achieve his goals even if the submatrix $\mathbf{S}$ is disconnected from the rest of the graph. It is well known that a disconnected subgraph retains its own singular values in the singular spectrum of the overall graph. As a result, we arrive at the following observation:

Observation 1. *An $f \times c$ attack submatrix $\mathbf{S}$, will evade spectral methods that use threshold σ_k, if the attack has a sufficiently small leading singular value σ':*

$$\sigma' < \sigma_k$$

Thus, we know what the attacker has to do: design an attack of f rows/accounts to c columns/customers, with s links per customer, so that the leading singular value of $\mathbf{S}$ is below σ_k. What are his choices, and which choices will allow him to achieve this goal?

We explore three attack strategies: "naïve", "staircase" and "random". Figure 2 gives a pictorial representation of each of these. We evaluate the suitability of each attack for an adversary on the basis of the leading singular value that the pattern generates.

A. Naïve Injection

The naïve injection distributes the sc total fraudulent actions into an $s \times c$ submatrix of $\mathbf{S}$, creating a $s \times c$ complete bipartite core as shown in Figure 2a.

Lemma 1. *The leading singular value of an $s \times c$ bipartite core injection is $\sigma_1 = \sqrt{cs}$.*

Proof: Omitted for brevity. ■

B. Staircase Injection

The staircase injection (discovered in [7]) evenly distributes cs fraudulent actions over f attacker nodes. However, unlike in the naïve method, where each node that performs any fraudulent actions does so for each of the c customers, the staircase method links different subsets of nodes with different subsets of customers. The resulting $\mathbf{S}$ sub-matrix resembles staircase, as shown in Figure 2b.

We restrict our analysis here to staircase injections in which all users have equal out-degrees o and equal in-degrees i, though o need not equal i. When out degrees and in degrees are not equal, users and objects do not have uniform connectivity properties which complicates calculations. In particular, we assume that the periodicity of the staircase pattern, given by $t = lcm(s, f)/s$ is such that $t|c$ to ensure this criteria. However, for large values of c/t, $\sigma_1 \approx s\sqrt{c/f}$ given LLN.

Theorem 1. *The leading singular value of an s, c, f staircase injection is $\sigma_1 = s\sqrt{c/f}$.*

Proof: (Sketch): By noting that the staircase injection is equivalent to a random graph-injection of $f \times c$ with edge probability $p = s/f$. ■

C. Random Graph Injection

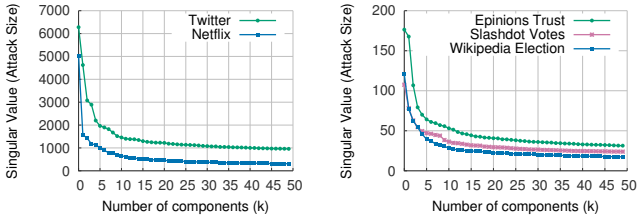
The random graph injection distributes roughly sc fraudulent actions over the f attacker nodes approximately evenly. Figure 2c shows a visual representation of such an attack. This approach assigns each node a fixed probability $p = s/f$ of performing a fraudulent operation associated with one of the c customers. The random graph injection is similar to the Erdös-Rényi model defined by $G(n, p)$ [4], except we consider a directed graph scenario with cf possible edges.

Theorem 2. *The leading singular value of an s, c, f directed random bipartite graph is $\sigma_1 \sim s\sqrt{c/f}$.*

Proof: (Sketch): by computing the expected row sums of $\mathbf{SS^T}$ and applying the Perron-Frobenius theorem. ■

All proofs are given in more detail in [14].

D. Implications and Empirical Analysis



(a) Twitter Followers (b) Smaller Graphs

Fig. 3: Skewed singular value distribution in real networks — spectral (k-rank SVD) approaches suffer from stealth attacks. (a) and (b) show distributions for corresponding networks which allow stealth attacks capable of signficantly impacting local network structure to go undetected.

Our analysis shows that two of the attack patterns, the staircase and random graph injections, produce leading singular values of $s\sqrt{c/f}$ respectively, whereas naïve injection results in a leading singular value of $\sigma_1 = \sqrt{cs}$. Thus, it is apparent

TABLE II: Graphs used for empirical analysis

Graph	Nodes	Edges
Twitter [8]	41.7 million	1.5 billion
Netflix [12]	480k users & 17k videos	99 million
Epinions [9]	131,828	841,372
Slashdot [9]	82,144	549,202
Wikipedia [9]	8274	114,040

that naïve injection is the least suitable for an adversarial use, since it will necessarily produce a larger singular value than the other two methods. Our results beget two important conclusions – firstly, that smarter means of attack than naïve exist and must be considered by detection algorithms and secondly, attackers can easily engineer attacks of scale up to *just below* thresholds without consequence by characterizing the singular values of their attacks.

To demonstrate that this leaves a significant opening for attackers, we analyze the distribution of singular values for a variety of real world graphs listed in Table II and show the results in Figure 3. Specifically, we use the Twitter who-follows-whom, Netflix product ratings, Epinions who-trusts-whom, Slashdot friend/foe and Wikipedia's administrative election graphs.

For a rank $k = 50$ decomposition, we observe the following: An attacker controlling 960 Twitter accounts could use them to follow 960 other accounts without being caught by existing spectral methods. In the Netflix scenario, an attacker could introduce 300 fake reviews to 300 movies. The same analyses can be extended to Epinions (30 trust links to 30 users, where the average number of links per user is only 6), Slashdot (23 ratings for 23 users where the average number of ratings per user is only 6) and Wikipedia (17 users voting on 17 elections, enough to win 31% of elections automatically). From these examples across a variety of networks, we see that using spectral approaches for catching fraud leaves a wide opening for attackers to manipulate online graphs.

IV. Proposed Algorithm

Thus far, we have seen how existing state-of-the-art techniques have firm effective detection thresholds and are entirely ineffective in detecting stealth attacks that fall below this threshold. Given this problem, it is natural to consider the following question — how can we identify the many numerous small scale attacks that are prone to slipping below the radar of existing techniques? In this section, we formalize our problem definition and propose fBox as a suitable method for addressing this problem.

A. Problem Formulation

We identify the major problem to be addressed as follows:

Problem 2. Given *an input graph adjacency matrix* $\mathbf{A}$*, with rows/columns corresponding to users/objects,* **identify** *stealth attackers which are undetectable given a desired decomposition rank-*k *for* $\mathbf{A}$.

Note that Problem 2 is an exact foil to Problem 1. In this paper, we primarily focus on smart attacks which fall below a practitioner-defined spectral threshold, given that a number of previous works mentioned have tackled the problem of discovering blatant attacks. Given that this body of work is effective in detecting such attacks, we envision that the best means of boxing in attackers is a *complementary* approach to existing methods.

Require: Input graph adjacency matrix $\mathbf{A}$,
Decomposition rank k,
Threshold τ

userCulprits = {}
objectCulprits = {}
outDegrees = $rowSum(\mathbf{A})$
inDegrees = $colSum(\mathbf{A})$
$[\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}] = svd(\mathbf{A}, k)$
for each row i in $\mathbf{U\Sigma}$ **do**
    reconstructedOutDegrees = $\|(\mathbf{U\Sigma})_\mathbf{i}\|_2^2$
end for
for each row j in $\mathbf{V\Sigma}$ **do**
    reconstructedInDegrees = $\|(\mathbf{V\Sigma})_\mathbf{j}\|_2^2$
end for
for each unique od in outDegrees **do**
    nodeSet = $find$(outDegrees $== od$)
    reconstructedOutDegreeSet = reconstructedOutDegrees(nodeSet)
    recThreshold = $percentile$(reconstructedOutDegreeSet, τ)
    for each node n in nodeSet **do**
        if reconstructedOutDegrees(n) $\leq$ recThreshold **then**
            userCulprits = userCulprits + n
        end if
    end for
end for
for each unique id in inDegrees **do**
    nodeSet = $find$(inDegrees $== id$)
    reconstructedInDegreeSet = reconstructedInDegrees(nodeSet)
    recThreshold = $percentile$(reconstructedInDegreeSet, τ)
    for each node n in nodeSet **do**
        if reconstructedInDegrees(n) $\leq$ recThreshold **then**
            objectCulprits = objectCulprits + n
        end if
    end for
end for
return userCulprits,
objectCulprits

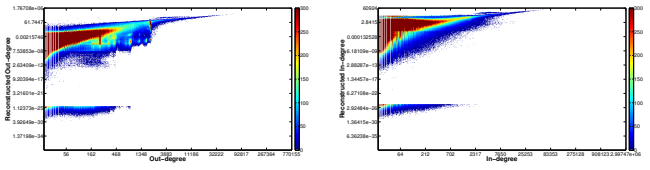
Algorithm 1: fBox algorithm pseudocode

B. Description

As per the problem formulation, we seek to develop a solely graph-based method, which will be able to complement existing fraud detection techniques by discerning previously undetectable attacks. In Section III, we demonstrated that smaller attacks do not appear in the singular vectors given by a rank k decomposition. Assuming an isolated attack which has been engineered to fall below the detection threshold, the users/objects comprising the attack will have absolutely *no* projection onto any of the top-k left and right singular vectors respectively. If the attack is only near-isolated, projection of the culprit nodes may increase slightly, but should still be near-zero given sparsity of connection to the honest graph. Given this observation, two questions naturally arise: (a) how can we effectively capture the extent of projection of a user or object? and (b) is there a pattern to how users or objects project into low-rank subspaces?

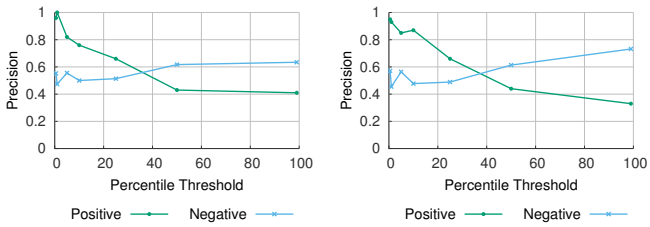
In fact, we can address the first question by taking advantage of the norm-preserving property of SVD, which states that the row vectors of a full rank decomposition and associated projection will retain the same l_2 norm or vector length as in the original space. That is, for $k = rank(\mathbf{A})$,

$$\|\mathbf{A_i}\|_2 = \|(\mathbf{U\Sigma})_\mathbf{i}\|_2 \text{ for } i \leq u$$



(a) OSRM for Twitter fans (b) ISRM for Twitter idols

Fig. 4: SRMs show correlation between the reconstruction degree and suspiciousness of nodes. (a) and (b) show the SRMs produced from analysis on the Twitter social graph.



(a) Fans (b) Idols

Fig. 5: (a) and (b) show FBOX's strong predictive value and low false-discovery rate in identifying suspicious accounts.

In the same fashion, one can apply the norm-preserving property to decomposition of $\mathbf{A^T}$ to show

$$\|\mathbf{A^T_j}\|_2 = \|(\mathbf{V\Sigma})_\mathbf{j}\|_2 \text{ for } j \leq o$$

Since the l_2 norms are preserved in a full rank decomposition, it is obvious that the sum of squares of components are also preserved. Note that for the 0-1 adjacency matrix $\mathbf{A}$ we consider here, the sum of squares of components of the ith row vector corresponds to the out-degree of user i and the sum of squares of components of the jth column vector corresponds to the in-degree of object j — given these considerations, we define the true/reconstructed degree of a node in a given subspace as the squared l_2 norm of its vector in that space.

We conjecture that due to the different graph connectivity patterns of dishonest and honest users/objects, their projections in terms of reconstructed degrees should also vary. Intuitively, dishonest users who either form isolated components or link to dishonest objects will project poorly whereas honest users who are well-connected to real products and brands should project well. In fact, we find that in real data, users and objects have certain ranges in which they commonly reconstruct in the latent space. Figure 4 shows the OSRM and ISRM (*Out-link/In-link Spectral Reconstruction Maps*) for a large, multi-million node and multi-billion edge social graph from Twitter, where we model follower (fan) and followee (idol) behavior. The data is represented in heatmap form to indicate the concentration of projection. The SRMs indicate that for each true degree, there is a tailed distribution with most nodes reconstructing in a common (reddish color) range and few nodes reconstructing as we move away from this range in either direction (bluish color). Most notably, there are a large number of nodes with degrees up to the thousands which project several orders of magnitude lower than the rest, well-depicted by a clearly isolated point cloud at the bottom of both SRMs.

These observations serve to substantiate our conjecture that poorly reconstructing nodes are suspicious, but what about the well reconstructing nodes? Interestingly, we find that nodes which reconstruct on the high range of the spectrum for a given degree have many links to popular and verified Twitter accounts. We do not classify such behavior as suspicious, as it is common for Twitter users to be connected with popular actors, musicians, brands, etc. At the bottom of the reconstruction spectrum, however, we most commonly find accounts which demonstrate a number of notably suspicious behaviors in the context of their followers/followees and the content of their Tweets – we focus our algorithm on identifying these. The FBOX algorithm pseudocode is given in Algorithm 1.

V. EXPERIMENTS

A. Datasets

For our experiments we primarily use two datasets: the who-follows-whom Twitter graph and the who-rates-what Amazon graph. The Twitter graph was scraped by Kwak et al. in 2010 and contains 41.7 million users with 1.5 billion edges [8]. The Amazon ratings graph was scraped in March 2013 by McAuley and Leskovec [11] and contains 29 million reviews from 6 million users about 2 million products. Our analysis is conducted both directly and via synthetic attacks.

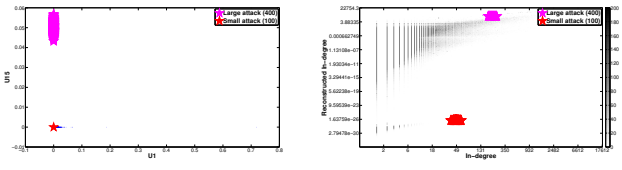
B. FBOX *on real Twitter accounts*

To show our effectiveness in catching smart link fraud attacks on real data, we conducted a classification experiment on data from the Twitter graph. Specifically, we collected the culprit results for suspicious fans and idols with degree at least 20 (to avoid catching unused accounts) for seven different values of the detection threshold τ, at 0.5, 1, 5, 10, 25, 50 and 99 percentile. For each combination of τ value and user type (fan or idol), we randomly sampled 50 accounts from the "culprit-set" of accounts classified as suspicious by FBOX and another 50 accounts from the remainder of the graph in a 1:1 fashion, for a total of 1400 accounts. We randomly organized and labeled these accounts as suspicious or honest based on several criteria including suspension since data collection, spammy tweets, suspicious usernames, and sparse profiles/few tweets but large numbers of suspicious followers.

Figure 5 shows how the performance of FBOX varies with the threshold τ for Twitter fans and idols. As evidenced by the results, FBOX is able to correctly discern suspicious accounts with 0.93+ precision for $\tau \leq 1$ for both fans and idols. Since recall is impossible to calculate given unbounded false negatives, we observe that negative precision increases as we increase τ. With these considerations, we recommend conservative threshold values for practitioner use. On Twitter data, we found roughly 150 thousand accounts classified as suspicious between the SRMs for $\tau = 1$.

C. Complementarity of FBOX

As mentioned before, FBOX is complementary to spectral techniques and is effective in catching smart attacks that



(a) Spectral subspace plot (b) ISRM

Fig. 6: FBOX and SPOKEN are complementary, with FBOX detecting smaller stealth attacks missed by SPOKEN. (a) shows how spectral subspace plots identify blatant attacks but ignore smaller ones. (b) shows the ISRM plot clearly identifying the suspiciousness of the small attack.

adversaries could engineer to avoid detection by these techniques. We demonstrate this claim using both synthetically formulated attacks on the Amazon network as well as comparing the performance of both FBOX and SPOKEN on the Twitter network. In the first experiment, we inject random attacks of scale 100 and 400, each with density $p = 0.5$ into the Amazon graph and compare the effectiveness of spectral subspace plots and SRMs in spotting these attacks. Figure 6a shows the spectral subspace plot for the 1st and 15th singular vectors, corresponding to a naturally existing community and the blatant attack, respectively. The plot clearly shows nodes involved in the blatant attack as a spoke, but groups nodes involved in the small attack along with many other honest nodes at the origin. However, in Figure 6b, we see that the ISRM distinguishes the attack from other legitimate behavior.

In our second experiment, we compared the performance of both FBOX and SPOKEN on a sample of 66K accounts selected from the Twitter graph. For each of these accounts, we queried the Twitter API to collect information regarding whether the account was suspended or had posted Tweets promoting adware/malware (checked via Google SafeBrowsing), and if so we marked the account as fraudulent. This ground truth marking allows us to unbiasedly measure the complementarity of FBOX and SPOKEN in catching users that are surely malicious. Of these users, 4K were marked as fraudulent via Twitter and Google SafeBrowsing. For rank $k = 50$, SPOKEN produced 8K suspicious accounts whereas FBOX (with $\tau = 1$) produced 150K. The user sets identified by both methods were found to be completely distinct, suggesting that the methods are indeed complementary. Furthermore, FBOX identified 1.1K suspicious accounts from the sampled dataset, of which only 350 were caught via Twitter and Google SafeBrowsing, suggesting that roughly 70% of FBOX-classified suspicious accounts are missed by Twitter.

D. Scalability of FBOX

The running time of FBOX is dominated by the (linear) large matrix-vector multiplication per iteration of the Lanczos algorithm to compute SVD for large, sparse matrices.

VI. CONCLUSIONS

In this work, we focused on spotting fraudsters and their customers in online social networks and web services. Our main contributions are:

1) **Theoretical analysis:** in order to examine spectral characteristics of certain attacks and identify limitations of existing detection methods
2) **FBOX algorithm:** a complementary method to existing spectral approaches that detects *stealth* attacks which previous methods miss
3) **Effectiveness on real data:** we apply FBOX to a large Twitter who-follows-whom dataset from 2010 and discover many tens of thousands of suspicious users

Our experiments show that our method is scalable, effective in detecting small-scale attacks on real data and catches a class of fraudsters previously undetected by existing approaches.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant Nos. IIS-1217559 and DGE-1252522 as well as a Facebook Fellowship. Prepared by LLNL under Contract DE-AC52-07NA27344.

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 National Science Foundation, DARPA, or other funding parties. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

REFERENCES

[1] L. Akoglu, M. McGlohon, and C. Faloutsos. Oddball: Spotting anomalies in weighted graphs. *Advances in Knowledge Discovery and Data Mining*, pages 410–421, 2010.

[2] A. Beutel, W. Xu, V. Guruswami, C. Palow, and C. Faloutsos. Copycatch: stopping group attacks by spotting lockstep behavior in social networks. In *WWW*, pages 119–130. ACM, 2013.

[3] N. Dalvi, P. Domingos, S. Sanghai, D. Verma, et al. Adversarial classification. In *SIGKDD*, pages 99–108. ACM, 2004.

[4] P. Erdős and A. Rényi. On random graphs. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.

[5] S. Ghosh, B. Viswanath, F. Kooti, N. K. Sharma, G. Korlam, F. Benevenuto, N. Ganguly, and K. P. Gummadi. Understanding and combating link farming in the twitter social network. In *WWW*, pages 61–70. ACM, 2012.

[6] C. Grier, K. Thomas, V. Paxson, and M. Zhang. @ spam: the underground on 140 characters or less. In *CCS*. ACM, 2010.

[7] M. Jiang, P. Cui, A. Beutel, C. Faloutsos, and S. Yang. Inferring strange behavior from connectivity pattern in social networks. In *PAKDD*, 2014.

[8] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a social network or a news media? In *WWW*, pages 591–600. ACM, 2010.

[9] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Signed networks in social media. In *SIGCHI*, pages 1361–1370. ACM, 2010.

[10] D. Lowd and C. Meek. Adversarial learning. In *SIGKDD*, pages 641–647. ACM, 2005.

[11] J. McAuley and J. Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *RecSys*, pages 165–172. ACM, 2013.

[12] Netflix. Netflix competition. 2006.

[13] B. Prakash, M. Seshadri, A. Sridharan, S. Machiraju, and C. Faloutsos. Eigenspokes: Surprising patterns and community structure in large graphs. *PAKDD*, 84, 2010.

[14] N. Shah, A. Beutel, B. Gallagher, and C. Faloutsos. Spotting suspicious link behavior with fbox: An adversarial perspective, 2014. arXiv preprint 1410.3915.

[15] N. Shrivastava, A. Majumder, and R. Rastogi. Mining (social) network graphs to detect random link attacks. In *ICDE*. IEEE, 2008.