

Suspicious Behavior Detection: Current Trends and Future Directions

Meng Jiang and Peng Cui, *Tsinghua University*

Christos Faloutsos, *Carnegie Mellon University*

As Web applications such as Hotmail, Facebook, Twitter, and Amazon have become important means of satisfying working, social, information-seeking, and shopping tasks, suspicious users (such as spammers, fraudsters, and other types of attackers) are increasingly attempting to engage in dishonest

Applications have different definitions of suspicious behaviors. Detection methods often look for the most suspicious parts of the data by optimizing scores, but quantifying the suspiciousness of a behavioral pattern is still an open issue.

activity, such as scamming money out of Internet users and faking popularity in political campaigns. Fortunately, commercially available suspicious behavior-detection techniques can eliminate a large percentage of spam, fraud, and sybil attacks on popular platforms. Naturally, the owners of these platforms want to ensure that any behavior happening on them involves a real person interested in interacting with a specific Facebook page, following a specific Twitter account, or rating a specific Amazon product.

Here, we describe detection scenarios that use different techniques to ensure security and long-term growth of real-world systems; we also offer an overview of the various methods in use today. As we move into the future, it's important that we continue to identify successful methods of suspicious behavior detection at analytical, methodological, and practical levels—especially those methods that can be adapted to real applications.

Detection Scenarios

We surveyed more than 100 advanced techniques for detecting suspicious behaviors that have existed over the past 10 years, dividing suspicious behaviors into four categories: traditional spam, fake reviews, social spam, and link farming. Figure 1 shows the percentages of research works that focus on these categories. The various works gather different aspects of information, such as content (C), network (N), and behavioral (B) patterns from behavioral data. Table 1 summarizes several experimentally successful detection techniques.^{1–23} From the figure and table, we can see tremendous progress in link-farming detection systems and trends in information integration.

Traditional Spam

A variety of spam-detection methods filter false or harmful information for traditional systems such as email or short message service (SMS).

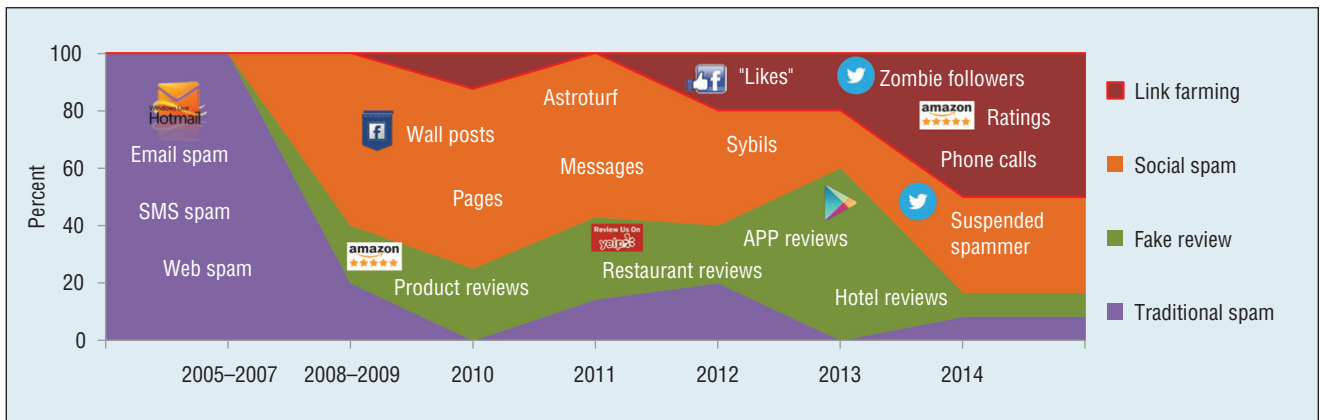


Figure 1. Percentage of recent research works on four main categories of suspicious behaviors: traditional spam, fake review, social spam, and link farming. Lately, we’ve seen tremendous progress in link-farming detection systems.

Table 1. Experimentally successful suspicious behavior detection techniques and their gathered information from data.*

Information aspects	Traditional spam	Fake reviews	Social spam	Link farming
C	AFSD ¹	—	Astroturf ² and Decorate ⁴	—
N	MailRank ⁴	—	SybilLimit ⁵ and Truthy ⁶	OddBall ⁷
B	SMSF ⁸	—	—	CopyCatch ⁹
C+N	—	—	SSDM ¹⁰ and SybilRank ¹¹	Collusionrank ¹²
C+B	—	ASM, ¹³ GSRank, ¹⁴ OpinSpam, ¹⁵ and SBM ¹⁶	URLSpam ¹⁷ and Scavenger ¹⁸	—
N+B	—	FraudEagle ¹⁹	—	Com2 ²⁰ and LockInfer ²¹
C+N+B	—	LBP ²²	—	CatchSync ²³

* C: content, N: network, and B: behavioral pattern.

Email spam can include malware or malicious links from botnets with no current relationship to the recipient, wasting time, bandwidth, and money. A content-based approach called Adaptive Fusion for Spam Detection (AFSD) extracts text features (bag-of-words) from an email’s character strings, develops a spam detector for a binary classification task (spam versus regular message), and shows promising accuracy in combating email spams.¹ People don’t want legitimate email blocked, so to take false-positive rates (FPRs) into consideration, AFSD gives the accuracy score mea-

sured by the area under the receiver-operating-characteristics curve (AUC) of 0.991 (with the highest score being 1) on a dataset from NetEase, one of the largest email service providers in China, indicating an almost-perfect performance in stopping text-rich spam for email services.

The MailRank system studies email networks using data gathered from the log files of a company-wide server and the email addresses that users prefer to communicate with (typically acquaintances rather than other unknown users).⁴ The system applies personalized PageRank algorithms with trusted

addresses from different information sources to classify emails. It achieves stable, high-quality performance: the FPR is close to zero, smaller than 0.5 percent when the network is 50 percent sparser.

Text message spams often use the promise of free gifts, product offers, or debt relief services to get users to reveal personal information. Due to the low cost of sending them, the resulting massive amount of SMS spam seriously harms users’ confidence in their telecom service providers. Content-based spam filtering can use indicative keywords in this space, such as “gift card” or “Cheap!!!,” but obtaining a text message’s content is expensive and often infeasible. An algorithm for SMS filtering (SMSF) detects spam using static features such as the total number of messages in seven days and temporal features such as message size every day.⁸ On SMS data from 5 million senders on a Chinese telecom platform, we can use static features to train support vector machine (SVM) classifiers and get the AUC to 0.883—incorporating temporal features gives an additional 7 percent improvement.

Researchers have developed various data mining approaches to detect email, SMS, and Web spam. High accuracy (near-1 AUC and near-0 FPR) makes these methods applicable in real systems, and some achieve a certain

degree of success. But with rising new platforms such as shopping and social networking sites, researchers have new challenges to tackle.

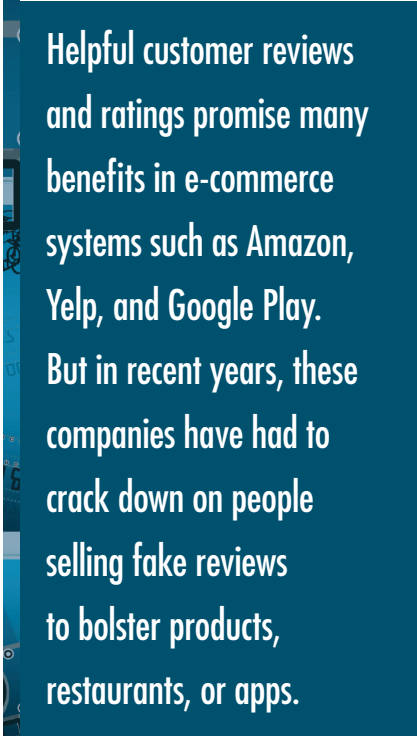
Fake Reviews

Helpful customer reviews and ratings promise many benefits in e-commerce systems such as Amazon, Yelp, and Google Play. But in recent years, these companies have had to crack down on people selling fake reviews to bolster products, restaurants, or apps. Fake reviews give undeserving positive opinions to promote a target object—or conversely, malicious negative opinions that damage an object’s reputation.

The first comprehensive study on trustworthiness in online reviews investigated 5.8 million reviews and 2.1 million reviewers on Amazon.¹⁵ This work had to define text features from reviews (length, opinion-bearing words), attribute features from products (price, sales rank), and rating-related features from reviewers (average rating, standard deviation in rating); it also had to use a logistic regression model to detect fake reviews (4,488 duplicate reviews served as ground truth). Using only text features gives only 0.63 on the AUC, indicating that it’s difficult to identify fake reviews using text content alone. Combining all the features gives the best result: 0.78 AUC. A scoring method called Spaming Behavior regression Model (SBM) simulates the behavioral patterns of fake review spammers: they target specific products to maximize their impact, and they tend to deviate from other reviewers in their ratings.¹⁶ SBM correctly identified the top and bottom 10 ranked reviewers as spammers and nonspammers, respectively, in a small labeled Amazon dataset of 24 spammers and 26 nonspammers.

In past five years, researchers have focused on discovering fake reviewers’ behavioral patterns and

combining these findings with text content to improve detection performance. GSRank studies fake review detection in a collaborative setting and uses a frequent itemset-mining method to find a set of fake reviewer groups.¹⁴ Using group features can improve the AUC from 0.75 to 0.95. The Author Spamicity Model (ASM) reviews spammers’ features in a latent space, receiving a 7 percent improvement in accuracy.¹³ The intuition is that spammers



Helpful customer reviews and ratings promise many benefits in e-commerce systems such as Amazon, Yelp, and Google Play. But in recent years, these companies have had to crack down on people selling fake reviews to bolster products, restaurants, or apps.

have different behavioral distributions than nonspammers, creating a divergence between the latent population distributions of the two reviewer clusters. FraudEagle spots fraudsters and fake reviews in online review datasets by exploiting the network effect among reviewers and products,¹⁹ and Loopy Belief Propagation (LBP) exploits the bursty nature of reviews to identify review spammers.²² Although review bursts can be due to either sudden product popularity or spam attacks,

spammers tend to work with other spammers, and genuine reviewers tend to work with other genuine reviewers. LBP incorporates review content, co-occurrence networks, and reviewer burstiness into a probabilistic graphical model (PGM). The “content + network + behavior” combination significantly increases accuracy from 0.58 to 0.78 in the binary classification task.

Social Spam

Social spam is unwanted user-generated content (UGC) such as messages, comments, or tweets on social networking services (SNSs) such as Facebook, MySpace, or Twitter. Successfully defending against social spammers is important for improving the quality of experience for SNS users.

The deployment of social honeypots harvests deceptive spam profiles from an SNS; from here, statistical analysis on these spam profiles creates spam classifiers that actively filter social spammers. Decorate, an ensemble learner of classifiers, uses features from profiles (such as sexual or advertisement content) to classify spammers and legitimate users.³ It obtains an accuracy of 0.9921 and an FPR of 0.007 on a MySpace dataset of 1.5 million profiles, and an accuracy of 0.8898 and an FPR of 0.057 on a Twitter dataset of 210,000 profiles. URLSpam focuses on detecting Twitter spammers who post tweets containing words found in trending topics and URLs that lead users to unrelated websites.¹⁷ It uses both content-based features (such as the number of hashtags or URLs) and behavioral features (number of tweets posted per day or time between tweets) as attributes of an SVM classifier, correctly classifying 70 percent of spammers and 96 percent of nonspammers. Scavenger is a clustering technique to group Facebook wall posts that show strong similarities in advertised URL destinations or text descriptions.¹⁸ Specifically, it characterizes static and temporal

properties of malicious clusters and identifies spam campaigns such as, “Get free ringtones” in 30,000 posts and “Check out this cool video” in 11,000 posts from a large dataset composed of more than 187 million posts. The proposed Social Spammer Detection in Microblogging (SSDM) approach is a matrix factorization-based model that integrates both social network information and content information for social spammer detection.^{10,24,25} The unified model achieves 9.73 percent higher accuracy than those with only one kind of information on a Twitter dataset of 2,000 spammers and 10,000 legitimate users.

Social sybils refer to suspicious accounts creating multiple fake identities to unfairly increase a single user’s power or influence. With social networking information of n user nodes, SybilLimit accepts only $O(\log n)$ sybil nodes per attack edge.⁵ The intuition is that if malicious users create too many sybil identities, the graph will have a small set of attack edges whose removal disconnects a large number of nodes (all the sybil identities). SybilRank relies on social graph properties to rank users according to their perceived likelihood of being fake sybils.¹¹ SybilRank found 90 percent of 200,000 accounts as most likely being fake on Tuenti, an SNS with 11 million users.

Social media has rapidly grown in importance as a forum for political, advertising, and religious activism. Astroturfing is a particular type of abuse disguised as spontaneous “grassroots” behavior, but that is in reality carried out by a single person or organization. Using content-based features such as hashtag, mentions, URLs, and phrases can help determine astroturfing content with 0.96 accuracy.² The Truthy system includes network-based information (degree, edge weight, and clustering coefficient) to track political memes in Twitter and detect

astroturfing campaigns in the context of US political elections.⁶

Link Farming

Link farming previously referred to a form of spamming on search engine indexes that connected all of a webpage’s hyperlinks to every other page in a group. Today, it’s grown to include many graph-based applications within millions of nodes and billions of edges. For example, in Twitter’s “who-follows-whom” graph, fraudsters are paid to make certain accounts seem more

Astroturfing is a particular type of abuse disguised as spontaneous “grassroots” behavior, but that is in reality carried out by a single person or organization.

legitimate or famous by giving them additional followers (zombies). In Facebook’s “who-likes-what-page” graph, fraudsters create ill-gotten page likes to turn a profit from groups of users acting together, generally liking the same pages at around the same time. Unlike spam content that can be caught via existing antispam techniques, link-farming fraudsters can easily avoid content-based detection: zombie followers don’t have to post suspicious content, they just distort the graph structure. Thus, the problem of combating link farming is rather challenging.

With a set of known spammers and a Twitter network, a PageRank-like approach can give high Collusionrank scores to zombie followers.¹² LockInfer

uncovers lockstep behaviors in zombie followers and provides initialization scores by reading the social graph’s connectivity patterns.²¹ CatchSync exploits two of the tell-tale signs left in graphs by fraudsters: they’re often required to perform some task together and have “synchronized” behavioral patterns, meaning their patterns are rare and very different from the majority.²³ Quantifying concepts and using a distance-based outlier detection method, CatchSync can achieve 0.751 accuracy in detecting zombie followers on Twitter and 0.694 accuracy on Tencent Weibo, one of the biggest microblogging platforms in China. CatchSync works well with content-based methods: combining content and behavioral information can improve accuracy by 6 and 9 percent, respectively. So far, it has found 3 million suspicious users who connect to around 20 users from a set of 1,500 celebrity-like accounts on the 41-million-node Twitter network, creating a big spike on the out-degree distribution of the graph. Furthermore, removing the suspicious user nodes leaves a smooth power law distribution on the remaining part of the graph, strong evidence that recall on the full dataset is high.

CopyCatch detects lockstep Facebook page-like patterns by analyzing the social graph between users and pages and the times at which the edges in the graph (the likes) were created.⁹ Specifically, it searches for temporally “bipartite cores,” where the same set of users like the same set of pages, and adds constraints on the relationship between edge properties (like times) in this core. CopyCatch is actively in use at Facebook, searching for attacks on its social graph. Com2 leveraged tensor decomposition on (caller, callee, and day) triplets and minimum description length (MDL)-based stopping criterion to find time-varying communities in a European Mobile Carrier dataset

Table 2. Recent suspicious behavior-detection methods in three main categories.

Methods	Traditional spam	Fake reviews	Social spam	Link farming
Supervised methods	AFSD ¹ and SMSF ⁸	LBP, ²² OpinSpam, ¹⁵ and SBM ¹⁶	Astroturf, ² Decorate, ³ SSDM, ¹⁰ and URLSpam ¹⁷	—
Clustering methods	—	ASM ¹³ and GSRank ¹⁴	Scavenger ¹⁸ and Truthy ⁶	—
Graph-based methods	MailRank ⁴	FraudEagle ¹⁹	SybilLimit ⁵ and SybilRank ¹¹	CatchSync, ²³ Collusionrank, ¹² Com2, ²⁰ CopyCatch, ⁹ LockInfer, ²¹ and OddBall ⁷

of 3.95 million users and 210 million phone calls over 14 days.²⁰ One observation was that five users received, on average, 500 phone calls each, on each of four consecutive days, from a single caller. Similarly, OddBall spots anomalous donator nodes whose neighbors are very well connected (“near-cliques”) or not connected (“stars”) on a large graph of political donations.⁷

Solving link-farming problems in the real world, such as hashtag hijacking, retweet promoting, or false news spreading, requires deep understanding for their specific suspicious behavioral patterns. Only through the integration of content, network, and behavioral information can we find multi-aspect clues for effective and interpretable detection.

Detection Methods

Suspicious behavior detection problems can be formulated as machine learning tasks. Table 2 presents three main categories of detection methods: supervised, clustering, and graph-based methods. Supervised methods infer a function from labeled email content, webposts, and reviews. Labeling the training data manually is often hard, so large-scale real systems use clustering methods on millions of suspicious users and identify spammer and fraudster clusters. Social network information and behavioral information are often represented as graph data, so graph-based methods have been quite popular in detecting suspicious behavioral links (injected following links, ill-gotten Facebook likes, and strange phone calls). Supervised methods are often applied to detect suspicious users

(fake reviewers, sybil accounts, and social spammers). Because labeling data is difficult and graph data is emerging, unsupervised methods (which include both clustering and graph-based methods) nicely overcome the limitations of labeled data access and generalize to the real world.

Supervised Methods

The major approaches in supervised detection methods are linear/logistic regression models, naive Bayesian models, SVM, nearest-neighbor algorithms (such as k -NN), least squares, and ensembles of classifiers (AdaBoost).

We start with suspicious users, such as social spammers, $\{u_1, \dots, u_N\}$, where the i th user is $u_i = (x_i, y_i)$, $x_i \in R^{D \times 1}$ (D is the number of features) is the feature representation of a certain user (a training example), and $y_i \in \{0, 1\}$ is the label denoting whether the user is spammer 1 or legitimate user 0. A supervised method seeks a function $g: X \rightarrow Y$, where X is the input feature space and Y is the output label space. Regressions and naive Bayes are conditional probability models, where g takes the form of $g(x) = P(y|x)$. Linear regression in SBM models the relationship between a scalar dependent variable (label) y and one or more independent variables (features) x . Logistic regression applied in AFSD, Decorate, and OpinSpam assumes a logistic function to measure the relationship between labels and features. Naive Bayes classifiers assume strong independence between features. Gaussian, multinomial, and Bernoulli naive Bayes have different assumptions on distributions of features. An SVM

constructs a hyperplane that represents the largest margin between the two classes (spammers and legitimate users). SMSF applies Gaussian kernels and URLSpam applies a radial basis function kernel to maximum-margin hyperplanes to efficiently perform a nonlinear classification. According to the distance measure, instead of the margin, the k -NN algorithms find the top k nearest neighbors of training instances from test instances. The least squares method in SSDM learns a linear model to fit the training data. This model can use an l_1 -norm penalization to control the sparsity. The classification task can be performed by solving the optimization problem

$$\min_{\mathbf{W}} \frac{1}{2} \| \mathbf{X}^T \mathbf{W} - \mathbf{Y} \|_F^2 + \lambda \| \mathbf{W} \|_1,$$

where $X \in R^{D \times N}$ denotes the feature matrix, $Y \in R^{N \times C}$ denotes the label matrix (C is the number of categories/labels, and $C = 2$ if we focus on classifying users as spammers or legitimate users), and λ is a positive regularization parameter. AdaBoost is an algorithm for constructing a strong classifier as a linear combination of simple classifiers. Statistical analysis has demonstrated that ensembles of classifiers can improve the performance of individual classifiers.

The key process in making supervised methods work better is feature engineering—that is, using domain knowledge of the data to create features. Feature engineering is much more difficult and time-consuming than feature selection (returning a subset of relevant features). Logistic regression models in OpinSpam, for

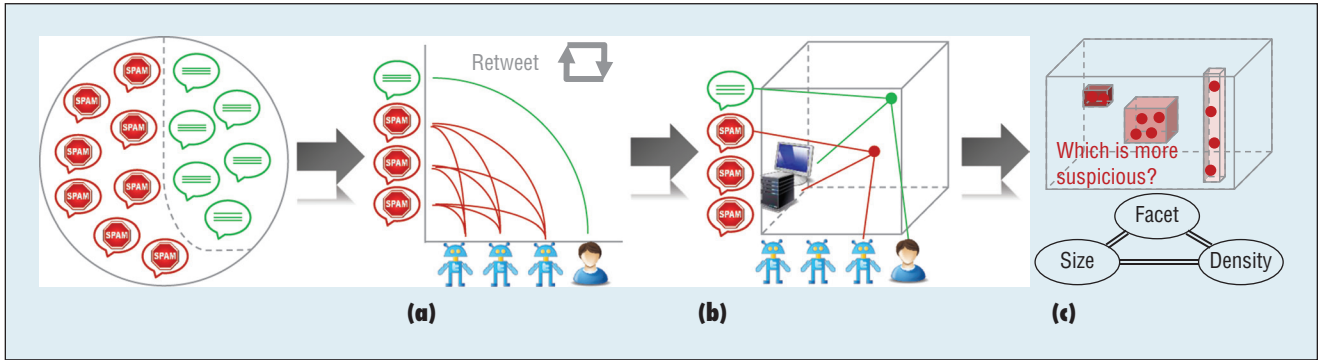


Figure 2. Three future directions for suspicious behavior detection: (a) behavior-driven suspicious pattern analysis, (b) multifaceted behavioral information integration, and (c) general metrics for suspicious behaviors. Detecting retweet hijacking requires comprehensively analyzing suspicious behavioral patterns (such as lockstep or synchronized). Detection techniques should integrate multifaceted behavioral information such as user, content, device, and time stamp.

example, were fed with 36 proposed features of content and behavior information from three aspects (review, reviewer, and product), all of which required knowledge from experts who are familiar with Amazon and its review dataset.

The bottleneck in supervised methods is the lack of labeled training data in large-scale, real-world applications. As CopyCatch says, unfortunately, there’s no ground truth about whether any individual Facebook page “like” is legitimate. Similarly, CatchSync argues that labeling Twitter accounts as zombie followers or normal users can be difficult with only subtle red flags—each account, on its own, generates a few small suspicions, but collectively, they raise many more. Researchers have come to realize the power of unsupervised methods such as clustering and graph-based methods that look for suspicious behaviors from millions of users and objects.

Clustering Methods

Clustering is the task of grouping a set of objects so that those in the same cluster are more similar to each other than to those in other clusters. To detect suspicious users in this manner, Scavenger extracts URLs from Facebook wall posts, builds a wall post similarity graph, and then clusters

wall posts that share similar URLs.¹⁸ The next step is to identify which clusters are likely to represent the results of spam campaigns. Clustering methods can reveal hidden structures in countless unlabeled data and summarize key features.

Latent variable models use a set of latent variables to represent unobserved variables such as the spamicity of Amazon reviewers, where spamicity is the degree of something being spam. The Author Spamicity Model (ASM) is an unsupervised Bayesian inference framework that formulates fake review detection as a clustering problem.¹³ Based on the hypothesis that fake reviewers differ from others on behavioral dimensions, ASM looks at the population distributions of two clusters—fake reviewers and normal reviewers—in a latent space. Its accurate classification results give good confidence that unsupervised spamicity models can be effective.

Graph-Based Methods

Graphs (directed/undirected, binary/weighted, static/time-evolving) represent interdependencies through the links or edges between objects via network information in social spam and behavioral information in link-farming scenarios. Graph-based suspicious detection methods can be categorized into PageRank-

like approaches and density-based methods.

PageRank-like approaches solve suspicious node detection problem in large graphs from the ranking perspective, such as MailRank for spam ranking, SybilRank and Collusion-Rank for sybil ranking, and FraudEagle for fraud ranking. For example, given a graph $G = (U, E)$, where $U = u_1, \dots, u_N$ is the set of nodes, E_{ij} is the edge from node u_i to u_j , and initial spamicity scores (PageRank values) of the set of nodes $R^0 = [R(u_1), \dots, R(u_N)]^T$, find the nodes that are most likely to be suspicious. The iterative equation of the solution is

$$R(t + 1) = dTR(t) + \frac{1 - d}{N} \mathbf{1},$$

where $T \in R^{N \times N}$ denotes the adjacency matrix of the graph or transition matrix. Note that the initial scores could be empty, but if they were determined with heuristic rules or former observations, the performance would improve. LockInfer works on Twitter’s “who-follows-whom” graph and infers strange connectivity patterns from subspace plots. With seed nodes of high scores from observations on the plots, the PageRank-like (trust propagation) algorithm accurately finds suspicious followers and followees who perform lockstep behaviors.

Density-based detection methods in graphs share the same idea with density-based clustering: they're looking for areas of higher density than the remainder of the graphs/data. The task is, given a graph G , to find all subgraphs G_{sub} (near-cliques, bipartite cores, and communities) that have anomalous patterns (unexpectedly high density). OddBall extracts features such as the number of node neighbors (degrees), number of subgraph edges, total subgraph weight, and principal eigenvalue of the subgraph's weighted adjacency matrix. With these features, OddBall uses traditional outlier detection methods for near-cliques that indicate malicious posts and fake donations. Com2 applies incremental tensor decomposition on a "caller-callee-time" dataset—that is, a phone call time-evolving graph—to find anomalous temporal communities. CopyCatch offers a provably-convergent iterative algorithm to search temporally coherent bipartite cores (dense "user-page" subgraphs) that indicate suspicious lock-step behaviors (such as group attacks or ill-gotten likes) in a million-node Facebook graph. CatchSync finds synchronized behavioral patterns of zombie follower on Twitter-style social networks and reports anomalous "who-follows-whom" subgraphs. Many graph-based approaches assume that the data exhibits a power-law distribution. CatchSync successfully removes spikes on the out-degree distributions by deleting the subgraphs.

Future Directions

For more than a decade, there has been tremendous growth in our understanding of suspicious behavior detection, with most research in this domain focusing on spam content analysis. Figure 2 offers three ideas for future directions that involve answering the following questions: What's the nature of suspicious behaviors?

How can we model behavioral information from real data? How can we quantify the suspiciousness of strange behavioral patterns?

Behavior-Driven Suspicious Pattern Analysis

Several approaches have been proposed for detecting fake accounts in social networks. Most learn content-based features from duplicate tweets,

Although they try to capture different aspects of spam behaviors, the patterns in these fake accounts easily can be varied by changing the scripts that create them, the updating speed of which is almost always faster than learning-based spam-detection algorithms.

malicious URLs, misleading content, and user profiles. Although they try to capture different aspects of spam behaviors, the patterns in these fake accounts easily can be varied by changing the scripts that create them, the updating speed of which is almost always faster than learning-based spam-detection algorithms. These detection methods also rely heavily on side information (such as, say, tweet content) that isn't always available and

is mostly available after the fact (after the account publishes malicious information). We need to change the focus from understanding how these fake accounts appear to behave (publishing duplicate tweets, malicious URLs, and so on) to conceal their fake identities or launch attacks to discovering how they must behave (follow or be followed) for monetary purposes.²³ The simple fact is that fake accounts consistently follow a group of suspicious followees so the companies hosting these fake accounts can earn money from them.

As Figure 2a shows, existing retweet hijacking-detection methods analyze text features in tweets and classify them as spam or normal content. To comprehensively capture hijacking behaviors, we need behavior-driven approaches that analyze retweeting behavioral links, assuming that retweet hijacking often forms "user-tweet" bipartite cores.

The nature of suspicious behaviors isn't content but monetary incentives, fraudsters, fake accounts, and many other kinds of suspicious users and their behavioral patterns. Behavior-driven suspicious pattern analysis can become a driving force in suspicious behavior detection.

Multifaceted Behavioral Information Integration

User behavior is the product of a multitude of interrelated factors. Some of these factors, such as physical environment, social interaction, and social identity, can affect how the behavior is related to monetary incentives or other motivations. For example, if a group of Twitter accounts engage in retweet hijacking, they operate together on a cluster of machines (maybe in the same building or city), promoting a small group of tweets during the same time period (retweeting every night in one week,

for example). Figure 2b represents retweeting behaviors as “user-tweet-IP-...” multidimensional tensors. User behaviors naturally evolve with the changing of both endogenous (intention) and exogenous (environment) factors, resulting in different multifaceted, dynamic behavioral patterns. However, there’s a lack of research to support suspicious behavior analysis with multifaceted and temporal information.

Flexible Evolutionary Multifaceted Analysis (FEMA) uses a flexible and dynamic high-order tensor factorization scheme to analyze user behavioral data sequences, integrating various bits of knowledge embedded in multiple aspects of behavioral information.²⁶ This method sheds light on behavioral pattern discovery in real-world applications, and integrating multifaceted behavioral information provides a deeper understanding of how to distinguish suspicious and normal behaviors.

General Metrics for Suspicious Behaviors

Suppose we use “user-tweet-IP,” a three-order tensor, to represent a retweeting dataset. Figure 2c gives us three subtensors: the first two are dense three-order subtensors of different sizes, and the third is a two-order subtensor that takes all the values on the third mode. For example, the first subtensor has 225 Twitter users, all retweeting the same 5 tweets, 10 to 15 times each, using 2 IP addresses; the second has 2,000 Twitter users, retweeting the same 30 tweets, 3 to 5 times each, using 30 IP addresses; and the third has 10 Twitter users, retweeting all the tweets, 5 to 10 times each, using 10 IP addresses. If our job at Twitter is to detect when fraudsters are trying to manipulate the most popular tweets, given time pressure and considering facet

(user, tweet, and IP), size, and density, which subtensor is more worthy of our investigation?

Dense blocks (subtensors) often indicate suspicious behavioral patterns in many detection scenarios. Purchased page likes on Facebook result in dense “user-page-time” three-mode blocks, and zombie followers on Twitter create dense “follower-followee” two-mode blocks. Density is worth inspecting, but how do we evaluate the suspiciousness? In other words, can we

If our job at Twitter is to detect when fraudsters are trying to manipulate the most popular tweets, given time pressure and considering facet (user, tweet, and IP), size, and density, which subtensor is more worthy of our investigation?

find a general metric for the suspicious behaviors?

A recent fraud detection study provides a set of basic axioms that a good metric must meet to detect dense blocks in multifaceted data (if two blocks are the same size, the denser one is more suspicious).²⁷ It demonstrates that while simple, meeting all the criteria is nontrivial. The authors derived a metric from the probability of “dense-block” events that meet the specified criteria. Their

experimental results show that a search algorithm based on this metric can catch hashtag promotion and retweet hijacking.

Different real-world applications have different definitions of suspicious behaviors. Detection methods often look for the most suspicious parts of the data by optimizing (maximizing) suspiciousness scores. However, quantifying the suspiciousness of a behavioral pattern is still an open issue. ■

References

1. C. Xu et al., “An Adaptive Fusion Algorithm for Spam Detection,” *IEEE Intelligent Systems*, vol. 29, no. 4, 2014, pp. 2–8.
2. J. Ratkiewicz et al., “Detecting and Tracking Political Abuse in Social Media,” *Proc. 5th Int’l AAAI Conf. Weblogs and Social Media*, 2011; www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/view/2850.
3. K. Lee, J. Caverlee, and S. Webb, “Uncovering Social Spammers: Social Honeypots + Machine Learning,” *Proc. 33rd Int’l ACM SIGIR Conf. Research and Development in Information Retrieval*, 2010, pp. 435–442.
4. P.-A. Chirita, J. Diederich, and W. Nejdl, “Mailrank: Using Ranking for Spam Detection,” *Proc. 14th ACM Int’l Conf. Information and Knowledge Management*, 2005, pp. 373–380.
5. H. Yu et al., “Sybillimit: A Near-Optimal Social Network Defense against Sybil Attacks,” *IEEE/ACM Trans. Networking*, vol. 18, no. 3, 2010, pp. 885–898.
6. J. Ratkiewicz et al., “Truthy: Mapping the Spread of Astroturf in Microblog Streams,” *Proc. 20th Int’l Conf. Comp. World Wide Web*, 2011, pp. 249–252.
7. L. Akoglu, M. McGlohon, and C. Faloutsos, “Oddball: Spotting

THE AUTHORS

Meng Jiang is a PhD candidate in the Department of Computer Science and Technology of Tsinghua University, Beijing. His research interests include behavioral modeling and social media mining. Jiang has a BS in computer science from Tsinghua University. Contact him at mjiang89@gmail.com.

Peng Cui is an assistant professor in the Department of Computer Science and Technology of Tsinghua University, Beijing. His research interests include social network analysis and social multimedia computing. Contact him at cui@tsinghua.edu.cn.

Christos Faloutsos is a professor in the School of Computer Science at Carnegie Mellon University. His research interests include large-scale data mining with emphasis on graphs and time sequences. Faloutsos has a PhD in computer science from University of Toronto. He's an ACM Fellow. Contact him at christos@cs.cmu.edu.

- Anomalies in Weighted Graphs,” *Advances in Knowledge Discovery and Data Mining*, LNCS 6119, Springer, 2010, pp. 410–421.
8. Q. Xu et al., “SMS Spam Detection Using Noncontent Features,” *IEEE Intelligent Systems*, vol. 27, no. 6, 2012, pp. 44–51.
 9. A. Beutel et al., “Copycatch: Stopping Group Attacks by Spotting Lockstep Behavior in Social Networks,” *Proc. 22nd Int’l Conf. World Wide Web*, 2013, pp. 119–130.
 10. X. Hu et al., “Social Spammer Detection in Microblogging,” *Proc. 23rd Int’l Joint Conf. Artificial Intelligence*, 2013, pp. 2633–2639.
 11. Q. Cao et al., “Aiding the Detection of Fake Accounts in Large Scale Social Online Services,” *Proc. 9th Usenix Conf. Networked Systems Design and Implementation*, 2012, pp. 15–28.
 12. S. Ghosh et al., “Understanding and Combating Link Farming in the Twitter Social Network,” *Proc. 21st Int’l Conf. World Wide Web*, 2012, pp. 61–70.
 13. A. Mukherjee et al., “Spotting Opinion Spammers Using Behavioral Footprints,” *Proc. 19th ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining*, 2013, pp. 632–640.
 14. A. Mukherjee, B. Liu, and N. Glance, “Spotting Fake Reviewer Groups in Consumer Reviews,” *Proc. 21st Int’l Conf. World Wide Web*, 2012, pp. 191–200.
 15. N. Jindal and B. Liu, “Opinion Spam and Analysis,” *Proc. 1st Int’l Conf. Web Search and Data Mining*, 2008, pp. 219–230.
 16. E.-P. Lim et al., “Detecting Product Review Spammers Using Rating Behaviors,” *Proc. 19th ACM Int’l Conf. Information and Knowledge Management*, 2010, pp. 939–948.
 17. F. Benevenuto et al., “Detecting Spammers on Twitter,” *Proc. Collaboration, Electronic Messaging, Anti-Abuse and Spam Conf.*, vol. 6, 2010, p. 12.
 18. H. Gao et al., “Detecting and Characterizing Social Spam Campaigns,” *Proc. 10th ACM SIGCOMM Conf. Internet Measurement*, 2010, pp. 35–47.
 19. L. Akoglu, R. Chandy, and C. Faloutsos, “Opinion Fraud Detection in Online Reviews by Network Effects,” *Proc. Int’l Conf. Web and Social Media*, 2013; www.aaai.org/ocs/index.php/ICWSM/ICWSM13/paper/viewFile/5981/6338.
 20. M. Araujo et al., “Com2: Fast Automatic Discovery of Temporal (‘Comet’) Communities,” *Advances in Knowledge Discovery and Data Mining*, LNCS 8444, Springer, 2014, pp. 271–283.
 21. M. Jiang et al., “Inferring Strange Behavior from Connectivity Pattern in Social Networks,” *Advances in Knowledge Discovery and Data Mining*, LNCS 8443, Springer, 2014, pp. 126–138.
 22. G. Fei et al., “Exploiting Burstiness in Reviews for Review Spammer Detection,” *Proc. 7th Int’l AAI Conf. Weblogs and Social Media*, 2013; www.aaai.org/ocs/index.php/ICWSM/ICWSM13/paper/viewFile/6069/6356.
 23. M. Jiang et al., “Catchsync: Catching Synchronized Behavior in Large Directed Graphs,” *Proc. 20th ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining*, 2014, pp. 941–950.
 24. X. Hu et al., “Social Spammer Detection with Sentiment Information,” *Proc. Int’l Conf. Data Mining*, 2014, pp. 180–189.
 25. X. Hu, J. Tang, and H. Liu, “Online Social Spammer Detection,” *Proc. 28th AAAI Conf. Artificial Intelligence*, 2014; www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/viewFile/8467/8399.
 26. M. Jiang et al., “FEMA: Flexible Evolutionary Multi-Faceted Analysis for Dynamic Behavioral Pattern Discovery,” *Proc. 20th ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining*, 2014, pp. 1186–1195.
 27. M. Jiang et al., “A General Suspiciousness Metric for Dense Blocks in Multi-Modal Data,” to be published in *Proc. IEEE Int’l Conf. Data Mining*, 2015; www.meng-jiang.com/pubs/crossspot-icdm15/crossspot-icdm15-paper.pdf.

 Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.