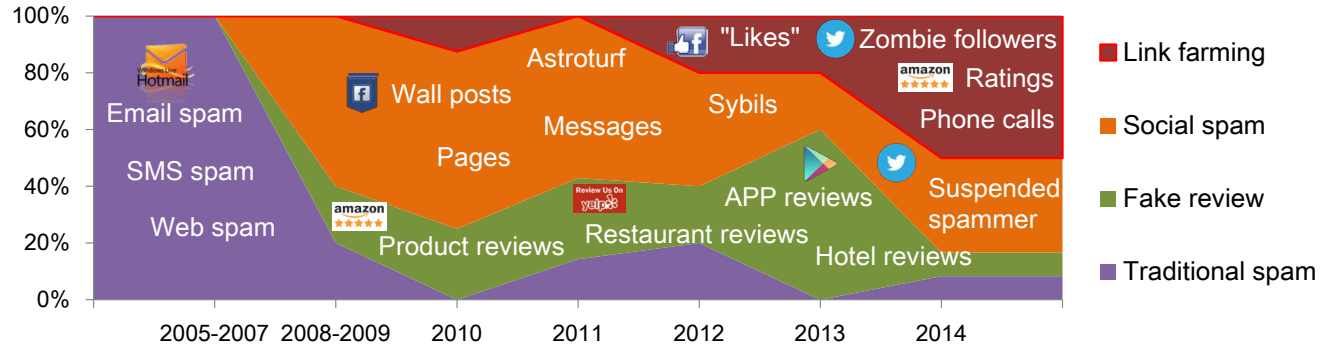
Suspicious Behavior Detection: Current Trends and Future Directions

Meng Jiang, *Tsinghua University*
Peng Cui, *Tsinghua University*
Christos Faloutsos, *Carnegie Mellon University*

In this article, the authors give an overview of suspicious behavior detection techniques, discuss current trends and present future directions.

Figure 1: This figure presents the percentages of the number of recent research works on detecting four main categories of suspicious behaviors. Lately, we have seen tremendous progress in link-farming detection systems.



Over the last decade, as Web applications such as Hotmail, Amazon, and Twitter have become important means of satisfying humans' behavioral needs in working, shopping, and information seeking, it has become a focus of suspicious users (*e.g.*, spammers, fraudsters, sybil accounts) attempting to grab dishonest interests (making money off Internet users, faking popularity in political campaign, and many other purposes). Suspicious behavior detection techniques are now commercially used to eliminate a large percentage of spams, frauds, and sybil attacks in popular platforms. These platforms want to ensure that behavior involves a real person interested in hearing from a specific Facebook Page, connecting to a specific Twitter account, and rating a specific Amazon product.

Here, we describe detection scenarios where the techniques are employed to ensure security and long-term growth of real-world systems, and then offer an overview of the various methods in use today. We find out current trends in application problems and solutions. As we move into the future it is important that we continue to identify successful methods of suspicious behavior detection at analytical, methodological, and practical levels that can be adapted to real applications in contexts.

Detection Scenarios

We surveyed over 100 advanced techniques on detecting suspicious behaviors during the last 10 years. We categorized the suspicious behaviors into four scenarios: traditional spam, fake review, social spam, and link farming. Figure 1 presents the percentages of the number of research work in these categories. These works gather different aspects of information such as content (C), network (N), and behavioral pattern (B) from behavioral data. Table 1 summarizes several experimentally successful detection techniques and their gathered information. From the figure and table, we can see tremendous progress in link-farming detection systems and trends of information integrations.

Traditional Spam

There exists a variety of spam detection methods to filter false/harmful information in traditional online systems such as e-mail or short message service (SMS).

E-mail spams may include malware or malicious links sent by a botnet having no current relationship with the recipient, and cause a waste of time, bandwidth, and

Table 1: This table summarizes experimentally successful suspicious behavior detection techniques and their gathered information from data (**C:** content, **N:** network, **B:** behavioral pattern).

	Traditional spam	**Fake review**	**Social spam**	**Link farming**
C	AFSD [25]		Astroturf [23], Decorate [19]	
N	MailRank [7]		SybilLimit [27], Truthy [24]	OddBall [2]
B	SMSF [26]			CopyCatch [5]
C+N			SSDM [13], SybilRank [6]	Collusionrank [10]
C+B		ASM [21], GSRank [22], OpinSpam [18], SBM [20]	URLSpam [4], Scavenger [9]	
N+B		FraudEagle [1]		Com2 [3], LockInfer [16]
C+N+B		LBP [8]		CatchSync [15]

money for the reader. A content-based approach AFSD extracts text features (bag-of-words) from character strings of an e-mail, develops a spam detector for a binary classification task (i.e., spam or regular e-mails), and shows promising accuracy in combating e-mail spams [25]. People tend to be strongly bothered by having desired e-mail blocked, i.e., high false positive rate (FPR). Thus, to take the FPR into consideration, AFSD gives the area under the receiver-operating-characteristics curve (AUC) score 0.991 on a dataset from NetEase, one of the largest e-mail service providers in China.

MailRank system studies e-mail networks using data gathered from the log files of a company-wide server, and addresses that users prefer to communicate with their acquaintances than other unknown users [7]. The system applies personalized PageRank algorithms with trusted e-mail addresses from different information sources to classify e-mails. It achieves stable, high-quality performance: the FPR is close to zero and it is smaller than 0.5% when the network is 50% sparser.

At the same time, text message spams often use the promise of free gifts, or product offers, or debt relief services to get users to reveal personal information. Due to the low cost of sending them, the massive count of SMS spam seriously harms the users' confidence in their telecom service providers. Some indicative keywords such as "GIFT CARD" or "CHEAP!!!" become important features that content-based spam filtering can use. However, text messages' contents are expensive or infeasible to obtain. A SMS filtering algorithm SMSF detects spam on the basis of behavioral features including static features, like total number of messages for seven days, and temporal features, like size of messages during every day [26]. On SMS data of 5 million senders from a Chinese telecom company, only using the static features to train SVM classifiers can reach a performance of AUC at 0.883, and by incorporating temporal features the AUC can get additional 7 percent improvements.

Researchers have developed various data mining approaches to detect e-mail spam [1] , SMS spam, 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. With rising new platforms such as shopping and social networking sites, researchers are devoted into more challenging problems.

Fake Review

Helpful customer reviews and review ratings promise many benefits to be derived from e-commerce systems such as Amazon, Yelp, and Google Play. At the same time, they are cracking down on sites that sell fake reviews to bolster products, restaurants, or Apps. Fake reviews are giving undeserving positive opinions to promote some target objects, or malicious negative opinions to damage some other objects' reputation, or irrelevant contents.

The first comprehensive study on trustworthiness of online reviews investigates 5.8 million reviews and 2.1 million reviewers crawled from Amazon [18] [2]. This work defines a large set of features including text features from reviews (e.g., length, opinion-bearing words), attribute features from products (e.g., price, sales rank), and rating related features from reviewers (e.g., average rating, standard deviation in rating) to characterize review, and uses logistic regression model to detect fake reviews (using 4,488 duplicate reviews as ground truth). Using only text features gives only 0.63 AUC, which shows that it is very difficult to identify fake reviews using text content alone. Combining all the features gives the best result - 0.78 AUC. A scoring method SBM models rating behavioral patterns of fake review spammers: they may target specific products to maximize their impact, and they tend to deviate from the other reviewers in their ratings of products [20]. The top 10 and the bottom 10 ranked reviewers are correctly spammers and non-spammers, respectively, in a small labelled Amazon dataset of 24 spam-

[1] One popular public dataset is from Kaggle competition: ADCG SS14 Challenge 02 - Spam Mails Detection.

[2] Datasets can be found here: http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#datasets

mers and 26 non-spammers.

In the last five years, researchers focus on discovering behavioral patterns of fake reviewers, and combine the new findings with text content to improve detection performance. GSRank studies fake review detection in the collaborative setting, and uses a frequent itemset mining method to find a set of fake reviewer groups [22]. Using group features can improve AUC from 0.75 to 0.95. ASM models review spammers' features in a latent space, receiving a 7% raise on accuracy [21]. The intuition is that spammers have different behavioral distributions from non-spammers, 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 [1]. LBP exploits the bursty nature of reviews to identify review spammers [8]. Though bursts of reviews can be either due to sudden popularity of products 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 network, and reviewer bursty 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 (SNS) such as Facebook, MySpace, or Twitter. Successfully defending against these social spammers is important to improve the quality of experience for SNS users.

The deployment of social honeypots harvests deceptive spam profiles from SNS, based on which statistical analysis of the properties of these spam profiles creates spam classifiers to actively filter social spammers. Decorate, an ensemble learner of classifiers, uses features from profiles (e.g., sexual content, advertisement content) to classify spammers and legitimates [19]. It obtains an accuracy of 0.9921, FPR of 0.007 on a MySpace dataset of 1.5 million profiles, and an accuracy of 0.8898 and FPR of 0.057 on a Twitter dataset of 210,000 profiles. URLSpam focuses on detecting Twitter spammers who post tweets containing typical words of a trending topic and URLs that lead users to unrelated websites [4]. It uses both content-based features (e.g., number of hashtags, number of URLs) and behavioral features (e.g., number of tweets posted per day, time between tweets) as attributes of a Support Vector Machine (SVM) classifier, and correctly classifies 70% of spammers and 96% of non-spammers. Scavenger is a clustering technique to group together Facebook wall posts that show strong similarities in advertised URL destination or text description [9]. It characterizes static and temporal properties of malicious clusters and identifies spam campaigns such as 'Get free ringtones" of 30,000 posts and "Checkout this cool video" of 11,000 posts from a large dataset composed of over 187 million posts. SSDM is a matrix factorization based model to integrate both social network information and content information for social spammer detection [13, 12, 11]. The unified model achieves 9.73% higher accuracy than those with only one kind of information on a Twitter dataset of 2,000 spammers and 10,000 normal users.

Social sybils refer to suspicious accounts creating multiple fake identities to unfairly increase the power or influence of a single user. With social networking information of n user nodes, SybilLimit accepts only $O(\log n)$ sybil nodes per attack edge [27]. 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 [6]. 90% of the 200,000 accounts that SybilRank designates as most likely to be fake on Tuenti, a SNS with 11 million users.

Social media has rapidly grown in importance as a forum for political, advertising, or religious activism. Astroturfing is one particular type of abuse disguised as spontaneous "grassroots" behavior that are in reality carried out by a single person or organization. Using content-based features such as hashtag, mentions, URLs, and phrases can achieve 0.96 accuracy on the detection of astroturfing content [23]. The Truthy system includes network-based information (e.g., degree, edge weight, clustering coefficient) to track political memes in Twitter and help detect astroturfing campaigns in the context of U.S. political elections [24].

The integration of information from social networks and behavioral patterns can significantly help detect spam content, astroturfing content, spammers, or sybil accounts.

Link Farming

Link farming previously referred to a form of spamming the index of a search engine by connecting all hyperlinks of a web page to every other page in a group. Now it is common in many graph-based applications within millions of nodes and billions of edges [3]. For example, in social networks, like Twitter's "who-follows-whom" graph, fraudsters are paid to make certain accounts seem more legitimate or famous through giving them many additional followers (zombie followers). In Facebook's "who-likes-what-page" graph, fraudsters make 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 by existing anti-spam techniques, link-farming fraudsters can

[3]KONECT has a large network collection within link farming datasets: http://konect.uni-koblenz.de/networks/

Table 2: This table classifies recent suspicious behavior detection methods in three main categories. Supervised methods have been well applied in detecting suspicious users (e.g., fake reviewers, sybil accounts, social spammers). As labeling data is hard and graph data is emerging, unsupervised methods including clustering methods and graph-based methods nicely overcome the limitations of labeled data access and generalize to the real world.

	Traditional spam	**Fake review**	**Social spam**	**Link farming**
Supervised methods	AFSD [25], SMSF [26]	LBP [8], OpinSpam [18], SBM [20]	Astroturf [23], Decorate [19], SSDM [13], URLSpam [4]	
Clustering methods		ASM [21], GSRank [22]	Scavenger [9], Truthy [24]	
Graph-based methods	MailRank [7]	FraudEagle [1]	SybilLimit [27], SybilRank [6]	CatchSync [15], Collusionrank [10], Com2 [3], CopyCatch [5], LockInfer [16], OddBall [2]

easily avoid content-based detection, e.g., the zombie followers do not have to post any suspicious content, but aim at distorting 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 [10]. LockInfer uncovers lockstep behaviors of the zombie followers and provides initialization scores by reading the social graph's connectivity patterns [16]. CatchSync exploits two of the tell-tale signs left in graphs by fraudsters: they are often required to perform some task together and have "synchronized" behavioral patterns; their patterns are "rare", very different from the majority [15]. Quantifying both concepts and using a distance-based outlier detection method, it can achieve 0.751 accuracy on detecting Twitter zombie followers and 0.694 accuracy on Tencent Weibo, one of the biggest microblogging platforms in China. CatchSync is complementary with content-based methods: combining both content information and behavioral information can improve the accuracy by 6% and 9%, respectively. It reports 3 million suspicious users who connect to around 20 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 at 20. Furthermore, removing them does leave only a smooth power law distribution on the remaining part of the graph, which is a strong evidence that recall on the full dataset is high.

CopyCatch detects lockstep Page Like patterns on Facebook by analyzing only the social graph between users and Pages and the times at which the edges in the graph (the Likes) were created [5]. It searches for temporally "bipartite cores", where the same set of users Like the same set of Pages, and add constraints on the relationship between edge properties (Like times) in this core. CopyCatch is actively in use at Facebook, searching for attacks on Facebook's social graph. Com2 leverages tensor decomposition on (caller, callee, day) triplets and MDL-based stopping criterion to find time-varying communities in a European Mobile Carrier dataset of 3.95 million users and 210 million phone calls in 14 days [3]. One of observations is that 5 users have received, on average, 500 phone calls each, on each of 4 consecutive days, from a single caller. OddBall spots anomalous donator nodes whose neighbors are very well connected ("near-cliques") or not connected ("stars") on a large graph of political donations [2].

Solving link farming problems in real world such as hashtag hijacking, retweet promoting, or false news spreading needs 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 methods, clustering methods, and graph-based methods. Supervised methods are inferring a function from labeled e-mail contents, web posts and reviews. Manual labeling the training data is often hard. Thus, 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. Graph-based methods have been very popular in detecting suspicious behavioral links (e.g., injected following links, ill-gotten Facebook Likes, strange phone calls).

Supervised Methods

The major approaches of supervised detection methods are linear/logistic regression models, naive Bayesian models, SVM, nearest neighbor algorithms (like k-NN), Least Squares, and ensembles of classifiers (like AdaBoost).

We take suspicious users, such as social spammers, $\{u_1, \ldots, u_N\}$,

where the i-th user is $u_i = (x_i, y_i)$, $x_i \in \mathbb{R}^{D\times 1}$ (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 "0". A supervised method seeks a function $g : X \to Y$, 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 used 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 the label and features. Naive Bayes classifiers assume strong independence between the features. With different assumptions on distributions of features, we have Gaussian, Multinomial and Bernoulli naive Bayes. A support vector machine constructs a hyperplane that represents the largest margin between the two classes (spammers and legitimates). SMSF applied Gaussian kernels and URLSpam applied a radial basis function kernel to maximum-margin hyperplanes so that they can efficiently perform a non-linear 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. 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

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

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

The key process to make supervised methods work better is feature engineering, i.e., 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 were fed with 36 proposed features of content and behavior information from three aspects (review, reviewer, product). All these features need knowledge from experts who are very familiar with Amazon and its review dataset.

The bottleneck of supervised methods is a lack of labeled training data in large-scale real-world applications. As CopyCatch says, unfortunately, there is no ground truth to whether any individual Facebook Page Like is legitimate or not. CatchSync argues that labeling Twitter accounts as zombie followers or normal users can be difficult with only a subtle red flags raising eyebrows. Each account, on its own, raises a few small suspicions, but CatchSync shows that, collectively, these accounts raise many more suspicions. Therefore, many researchers have realized the power of unsupervised methods like clustering and graph-based methods that look for suspicious behaviors from millions of users and objects.

Clustering Methods

For detecting suspicious objects (e.g., users, wall posts, products), clustering is the task of grouping a set of objects so that objects in the same cluster are more similar to each other than to those in other clusters. Scavenger extracts URLs from Facebook wall posts, builds the wall post similarity graph, and then cluster together wall posts that share similar URLs [9]. The next step is to identify which clusters are likely to represent the results of spam campaigns. Clustering methods can reveal hidden structure in countless unlabeled data, and seek to summarize key features of the data.

Latent variable models are statistical models where a set of latent variables exist to represent unobserved variables like the spamicity of Amazon reviewers. Spamicity here means is the degree of being spamming. Author Spamicity Model (ASM) is an unsupervised Bayesian inference framework which formulates fake review detection as a clustering problem [21]. Based on the hypothesis that fake reviewers differ from others on behavioral dimensions, ASM models the population distributions of two clusters, fake reviewers and normal reviewers, in a latent space. The accurate classification results give a good confidence that unsupervised spamicity models can be effective.

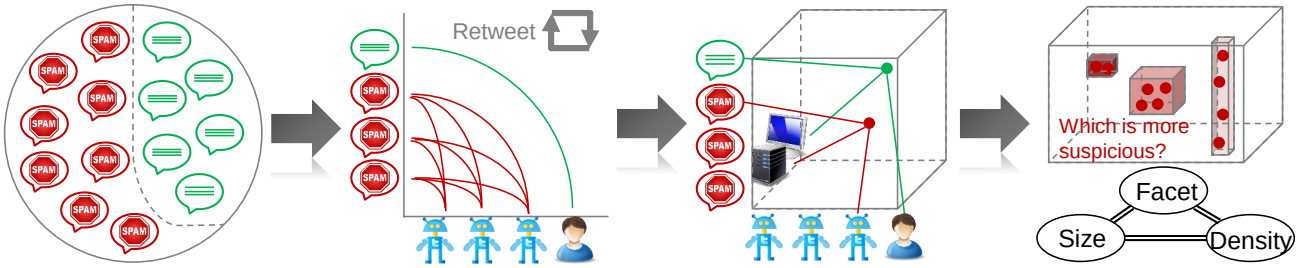
Graph-based Methods

Graphs (e.g., directed/undirected graphs, binary/weighted graphs, static/time-evolving graphs) naturally represent the inter-dependencies by the links or edges between the objects from network information in social spam and behavioral information in link farming scenarios. Graph-based suspicious detection methods can be categorized into two: PageRank-like approaches and density-based methods.

PageRank-like approaches such as MailRank, SybilRank, CollusionRank, and FraudEagle, have been proposed to deal with the problem of ranking suspicious nodes such as spam ranking, sybil ranking, and fraud ranking: given a graph $G = (U, E)$ where $U = u_1, \ldots, u_N$ is the set of nodes and E_{ij} is the edge from node u_i to u_j, and initial spamicity scores (PageRank values) of the set of nodes $\mathbf{R}^0 = [R(u_1) \ldots R(u_N)]^{\mathrm{T}}$, find the nodes that are most likely to be suspicious. The iterative equation of the solution is

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

Figure 2: This figure illustrates three future directions in suspicious behavior detection. To detect retweet hijacking requires comprehensively analyzing suspicious behavioral patterns (e.g., lockstep, synchronized). Detection techniques should integrate multi-faceted behavioral information such as user, content, device and timestamp. Thus, it is important to define metrics for generally evaluating the suspiciousness from perspectives of facet, size and density.



(a) Behavior-driven suspicious pattern analysis

(b) Multi-faceted behavioral information integration

(c) General metrics for suspicious behaviors

where $\mathbf{T} \in \mathbb{R}^{N \times N}$ denotes the adjacency matrix of the graph or transition matrix. Note that the initial scores could be empty, but if they were determined with some heuristic rules or former observations, the performance would be better. LockInfer works on a 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 find suspicious followers and followee who perform lockstep behaviors.

Density-based detection methods in graphs share the same idea with density-based clustering: they are looking for areas of higher density than the remainder of the graphs/data. The task is given a graph G, to find all subgraphs $\mathcal{G}_{sub}$ (e.g., near-cliques, bipartite cores, communities) that have anomalous patterns (unexpectedly high density). OddBall extracts features such as number of neighbors (degrees) of nodes, number of edges in the subgraph, total weight of the subgraph, and principal eigenvalue of the weighted adjacency matrix of the subgraph. 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, i.e., 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 lockstep behaviors (e.g., group attacks, 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. One of the subgraphs contains 3 million accounts connecting to 20 random targets from a group of 1,500 "famous" users such as actors, musicians, politicians and e-magazines. Many of the 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. Most research on this domain has focused on spam content analysis. In Figure 2 we offer three points of future directions by answering the following questions: (1) What is the nature of suspicious behaviors? (2) How to model behavioral information from real data? (3) How to quantify the suspiciousness of strange behavioral patterns?

Behavior-driven Suspicious Pattern Analysis

In recent years, a number of approaches were proposed for fake account detection in social networks. Most of them learned content-based features from duplicate tweets, malicious URLs, misleading contents, and user profiles. Although they tried to capture different aspects of spam behaviors, the behavior patterns of these fake accounts can be easily varied by changing the fake-account creating scripts, whose updating speed can be always faster than these learning-based spam detection algorithms. Meanwhile, these methods heavily rely on side information (i.e. the information except social graph and user profile, for example, the tweet contents) which are not always available, and are mostly after-the-fact (i.e. after the account having published a number of malicious information). It is necessary to change the focus from understanding how these fake accounts appear to behave (e.g. publish duplicate tweets, or malicious URLs, etc.) to conceal their fake identities or carry out attacks, to discovering how they have to behave (e.g. follow and get followed) to attain the monetary purpose [15]. The simple fact is that the fake accounts consistently follow a group of suspi-

cious followees, so that the companies who host these fake accounts can earn money from these suspicious followees, and the suspicious followees can realize rapid increasing on number of followers.

Moreover, as shown in Figure 2(a), existing retweet hijacking detection methods analyzed text features of the tweets and classified them into spam content and normal content. To comprehensively capture the hijacking behaviors, we need behavior-driven approaches on analyzing retweeting behavioral links, assuming that retweet hijacking often forms "user-tweet" bipartite cores.

Therefore, the nature of suspicious behaviors is not appearance like contents but monetary incentives of spammers, fraudsters, fake accounts, and many other kinds of suspicious users, and their strange behavioral patterns. Behavior-driven suspicious pattern analysis has become a direction of suspicious behavior detection.

Multi-faceted Behavioral Information Integration

User behavior is the product of a multitude of interrelated factors. The factors such as physical environment, social interaction, and social identity, affect how the behavior takes place with users' monetary incentives or normal motivations. For example, if a group of Twitter accounts aim at retweet hijacking, they operate together on a cluster of machines (maybe, in the same building, the same city), promoting a small group of tweets of advertising content during the same time period (e.g., retweeting every night in one week). Figure 2(b) represents retweeting behaviors as "user-tweet-IP-. . . " multi-dimensional tensors. User behaviors naturally evolve with the changing of both endogenous factors (e.g., intentions) and exogenous factors (e.g., environments), resulting in different multi-faceted, dynamic behavioral patterns. However, there is a lack of research to support suspicious behavior analysis with multi-faceted and temporal information.

Flexible Evolutionary Multi-faceted Analysis (FEMA) utilizes a flexible and dynamic high-order tensor factorization scheme for analyzing user behavioral data sequences, which can integrate various knowledge embedded in multiple aspects of behavioral information [17]. This method sheds light on behavioral pattern discovery in real-world applications. Integrating multi-faceted behavioral information provides deep understanding of how to distinguish between suspicious and normal behaviors.

General Metrics for Suspicious Behaviors

Suppose we use "user-tweet-IP", a 3-order tensor to represent a retweeting dataset, Figure 2(c) gives three subtensors: the first two are dense 3-order subtensors of different sizes, and the third one is a 2-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 tweet, 10 to 15 times each, using 2 IP addresses; the second one has 2,000 Twitter users, retweeting the same 30 tweets, 3 to 5 times each, using 30 IP addresses; the third subtensor 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, considered facet (e.g., user, tweet, 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" 3-mode blocks and zombie followers on Twitter create dense "follower-followee" 2-mode blocks. Density is worth inspecting but how to evaluate the suspiciousness? In other words, can we find a general metric for the suspicious behaviors?

A recent fraud detection work provides a set of basic axioms that a good metric must meet to detect dense blocks in multi-faceted data (e.g., if two blocks are the same size, the denser one is more suspicious) [14]. It demonstrates that while simple, meeting all the criteria is non-trival. The authors derive a metric from the probability of "dense-block" events that meet the specified criteria. 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, they have to face a challenging issue: how to quantify the suspiciousness of a suspicious behavioral pattern is still an open issue.

Meng Jiang is a Ph.D. candidate in the Department of Computer Science and Technology of Tsinghua University, Beijing. Contact him at jm06@mails.tsinghua.edu.cn.

Peng Cui is an assistant professor in the Department of Computer Science and Technology of Tsinghua University, Beijing. Contact him at cuip@tsinghua.edu.cn.

Christos Faloutsos is a professor at the School of Computer Science of Carnegie Mellon University. Contact him at christos@cs.cmu.edu.

References

[1] L. Akoglu, R. Chandy, and C. Faloutsos. Opinion fraud detection in online reviews by network effects. In *Int'l Conf. Web and Social Media*, 2013.

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

[3] M. Araujo, S. Papadimitriou, S. Günnemann, C. Faloutsos, P. Basu, A. Swami, E. E. Papalexakis, and D. Koutra. Com2: Fast automatic discovery of temporal ('comet') communities. In *Advances in Knowledge Discovery and Data Mining*, pages 271–283. 2014.

[4] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida. Detecting spammers on twitter. In *Collaboration, Electronic messaging, Anti-abuse and Spam Conference*, volume 6, page 12, 2010.

[5] 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'l Conf. World Wide Web*, pages 119–130, 2013.

[6] Q. Cao, M. Sirivianos, X. Yang, and T. Pregueiro. Aiding the detection of fake accounts in large scale social online services. In *Proc. 9th USENIX Conf. Networked Systems Design and Implementation*, pages 15–28, 2012.

[7] P.-A. Chirita, J. Diederich, and W. Nejdl. Mailrank: Using ranking for spam detection. In *Proc. 14th ACM Int'l Conf. Information and Knowledge Management*, pages 373–380, 2005.

[8] 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'l AAAI Conf. Weblogs and Social Media*, 2013.

[9] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Y. Zhao. Detecting and characterizing social spam campaigns. In *Proc. 10th ACM SIGCOMM Conf. Internet Measurement*, pages 35–47, 2010.

[10] 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 *Proc. 21st Int'l Conf. World Wide Web*, pages 61–70, 2012.

[11] X. Hu, J. Tang, H. Gao, and H. Liu. Social spammer detection with sentiment information. In *Proc. Int'l Conf. on Data Mining*, pages 180–189, 2014.

[12] X. Hu, J. Tang, and H. Liu. Online social spammer detection. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[13] X. Hu, J. Tang, Y. Zhang, and H. Liu. Social spammer detection in microblogging. In *Proc. 23rd Int'l Joint Conf. Artificial Intelligence*, pages 2633–2639, 2013.

[14] M. Jiang, A. Beutel, P. Cui, B. Hooi, S. Yang, and C. Faloutsos. A general suspiciousness metric for dense blocks in multi-modal data. In *Proc. IEEE Int'l Conf. on Data Mining*, 2015.

[15] 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'l Conf. Knowledge Discovery and Data Mining*, pages 941–950, 2014.

[16] M. Jiang, P. Cui, A. Beutel, C. Faloutsos, and S. Yang. Inferring strange behavior from connectivity pattern in social networks. In *Advances in Knowledge Discovery and Data Mining*, pages 126–138. 2014.

[17] M. Jiang, P. Cui, F. Wang, X. Xu, W. Zhu, and S. Yang. Fema: Flexible evolutionary multi-faceted analysis for dynamic behavioral pattern discovery. In *Proc. 20th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, pages 1186–1195, 2014.

[18] N. Jindal and B. Liu. Opinion spam and analysis. In *Proc. 1st Int'l Conf. Web Search and Data Mining*, pages 219–230, 2008.

[19] K. Lee, J. Caverlee, and S. Webb. Uncovering social spammers: Social honeypots + machine learning. In *Proc. 33rd Int'l ACM SIGIR Conf. Research and Development in Information Retrieval*, pages 435–442, 2010.

[20] E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw. Detecting product review spammers using rating behaviors. In *Proc. 19th ACM Int'l Conf. Information and Knowledge Management*, pages 939–948, 2010.

[21] A. Mukherjee, A. Kumar, B. Liu, J. Wang, M. Hsu, M. Castellanos, and R. Ghosh. Spotting opinion spammers using behavioral footprints. In *Proc. 19th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, pages 632–640, 2013.

[22] A. Mukherjee, B. Liu, and N. Glance. Spotting fake reviewer groups in consumer reviews. In *Proc. 21st Int'l Conf. World Wide Web*, pages 191–200, 2012.

[23] J. Ratkiewicz, M. Conover, M. Meiss, B. Gonçalves, A. Flammini, and F. Menczer. Detecting and tracking political abuse in social media. In *Proc. 5th Int'l AAAI Conf. Weblogs and Social Media*, 2011.

[24] J. Ratkiewicz, M. Conover, M. Meiss, B. Gonçalves, S. Patil, A. Flammini, and F. Menczer. Truthy: Mapping the spread of astroturf in microblog streams. In *Proc. 20th Int'l Conf. Comp. World Wide Web*, pages 249–252, 2011.

[25] C. Xu, B. Su, Y. Cheng, and W. Pan. An adaptive fusion algorithm for spam detection. *IEEE Intelligent Systems*, 29(4):2–8, 2014.

[26] Q. Xu, E. W. Xiang, Q. Yang, J. Du, and J. Zhong. Sms spam detection using noncontent features. *IEEE Intelligent Systems*, 27(6):44–51, 2012.

[27] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao. Sybillimit: A near-optimal social network defense against sybil attacks. *IEEE/ACM Trans. Netw.*, 18(3):885–898, 2010.