FraudVis: Understanding Unsupervised Fraud Detection Algorithms

Jiao Sun[1*] Qixin Zhu[1†] Zhifei Liu[1‡] Xin Liu[1§] Yueming Wang[1¶] Jihae Lee[1‖] Lei Shi[2**]
Ling Huang[1††] Wei Xu[1‡‡]

[1]Institute of Interdisciplinary Information Sciences, Tsinghua University
[2]SKLCS, Institute of Software, Chinese Academy of Sciences

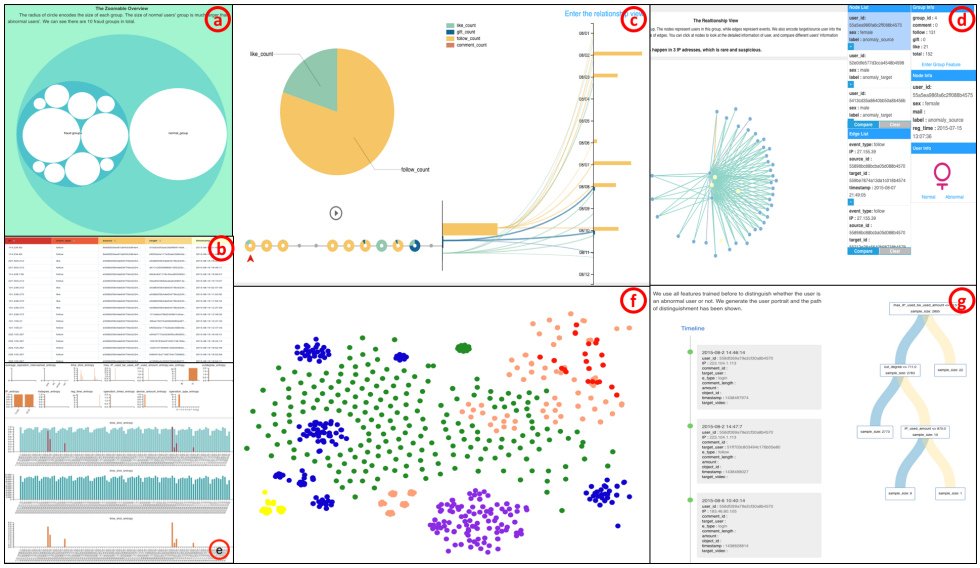


Figure 1: FraudVis interfaces: (a) Global fraud groups, (b) Part of raw data with title encoding the feature importance, (c) Group activity view for temporal sequence of fraud users' behavior, (d) User interaction view indicating relationships among users within a group, (e) The comparison of features that contribute the most to the detection result in overall and group scale, (f) Inter-group comparison for users in five most-similar-sized groups, (g) Tree view that decides the result and timeline of a certain user.

Abstract

Discovering fraud user behaviors is vital to keeping online websites healthy. Fraudsters usually exhibit grouping behaviors, and researchers have effectively leveraged this behavior to design unsupervised algorithms to detect fraud user groups. In this work, we propose a visualization system, FraudVis, to visually analyze the unsupervised fraud detection algorithms from temporal, intra-group correlation, inter-group correlation, feature selection, and the individual user perspectives. FraudVis helps domain experts better understand the algorithm output and the detected fraud behaviors. Meanwhile, FraudVis also helps algorithm experts to fine-tune the algorithm design through the visual comparison. By using the visualization system, we solve two real-world cases of fraud detection, one for a social video website and another for an e-commerce website. The results on both cases demonstrate the effectiveness of FraudVis in understanding unsupervised fraud detection algorithms.

*e-mail: j-sun16@mails.tsinghua.edu.cn
†e-mail:bruce.waynezu@gmail.com
‡e-mail:liuzhifei0@gmail.com
§e-mail:liuxin16@mails.tsinghua.edu.cn
¶e-mail:lynn.yueming.wang@gmail.com
‖e-mail:lizhihui17@mails.tsinghua.edu.cn
**e-mail:shijim@gmail.com
††e-mail:huang.ling@gmail.com
‡‡e-mail:weixu@tsinghua.edu.cn

1 Introduction

Online frauds are the well-known dark side of the modern Internet, causing millions of worldwide complaints [11] and billions of economic losses each year globally [4]. For example, on content-based social networks, fraudsters can pay to promote certain merchandise, or spread spams [25,27]. On Internet-based finance, criminals exploit fake identities to apply for loans [7], purchase things with hijacked credit cards [22], or even conduct money laundering. Increasingly, it has become crucial to deploy anti-fraud technologies on the Internet business.

While there are many ways to identify frauds on the Internet, in this work, we focus on the fraud detection through analyzing user logs (aka. the clickstreams). It is believed that the fraud users have rather different features on their logs, compared with the other legitimate users. In particular, fraudsters exhibit a *grouping behavior* in that they can be partitioned into a few clusters with high intra-cluster consistency, e.g., a tiny set of IP addresses, a short list of cell phone numbers, or synchronized timestamps on their activities [1]. In contrast, the legitimate users often have their logs randomly distributed in the feature space. This grouping behavior is hard to avoid as fraudsters.

People have developed many fraud detection algorithms based on this grouping behavior, in particular, the unsupervised learning methods such as clustering [8,12,21]. However, designing the cluster algorithm and evaluating its result are challenging: 1) the user log data contains many dimensions describing a single user's behavior, and it is difficult to select the most related dimensions to their fraud behavior; 2) the selection of features and algorithms depends heavily on the characteristics of the log data and scenario; and 3) in most cases, there are no fraud labels for evaluation, as the fraud behaviors can be confirmed only when a visible damage is noticed, often after

a long time. Given these challenges, we believe that visualization is an indispensable component of a fraud detection system, as it helps keep users in the loop of actual detecting, evaluating and confirming various types of fraud behaviors.

In this paper, we propose FraudVis, as shown in Figure 1, a comprehensive visualization system to assist both algorithm experts and domain experts to better understand the fraud detection results. For algorithm experts, FraudVis presents detailed *in situ* information to let them fine-tune the feature selection and detection algorithms. For example, we answer their questions such as: *what are the distributions of important features?*, *do users in the same group share the same pattern?*, *why does this false positive happen?*.

For domain experts, FraudVis opens the algorithm black box, deepens their understandings of the detection mechanism, and finally helps them to verify the detected frauds by combining their domain knowledge.

We summarize the contribution of this work as follows.

Comprehensive visualization of fraud detection results. Based on the in-depth investigation of the underground black market, the grouping patterns of the fraud users and state-of-the-art fraud detection algorithms, we design the visualization interfaces of FraudVis as coordinated multiple views, which display the temporal feature of user logs, the correlation pattern of the intra-group and inter-group features, as well as the detection results on individual users.

Visual interpretation of algorithm results through customized interactions. By interweaving commodity visualization components, FraudVis allows users to explore the fraud behaviors and the algorithmic detection process in many interactive ways. In particular, we highlight the feature selection process to visually interpret the fraud detection algorithm. We provide customized interactions for separate classes of users (i.e., algorithm experts and domain experts) to navigate through the interfaces till their desired level of details.

Evaluation through real-world data sets, algorithms, and cases. We demonstrate the power of FraudVis on two real-world cases with different fraud types. We visualize results from two categories of fraud detection algorithms, demonstrating the flexibility of FraudVis interfaces in visually analyzing the detection result and explaining the working mechanism of algorithms in each case.

2 Fraud Behavior and Detection

2.1 Frauds and the Black Market

The key to conducting online frauds lies in collecting lots of disposable accounts on the target website (aka, fake). To break into the defense of website providers with low cost, fraudsters established a professional chain of *fraud services* through the dark web (aka, the black market) [28]. For example, there are people selling phone verification services on the dark web for very low prices, say \$140$\sim$420 for 1000 mobile SIMs.

Though economically feasible, this resource sharing mechanism enables a fundamental way to detect the fraud behaviors based on these resources. Compared with legitimate accounts, fraudsters exhibit unusually similar behaviors in certain aspects, e.g., re-use of phone numbers, similar phone access durations, highly recurrent IP ranges, and the regular frequency of activities. The goal of fraud detection algorithms is efficiently discovering these grouping behaviors so we can stop these shared resources from further trading on the black market.

2.2 Fraud Detection Algorithms

Nevertheless, it is non-trivial to detect these grouping behaviors. People often prefer unsupervised detection algorithms as there are few labels for fraudsters. Besides, since the fraud behaviors alter frequently to avoid detection, the historical data are not as useful as in other scenarios such as recommendation. There are two types of algorithms to detect such behaviors, dense-subgraph-based and vector-space-based. As examples, k-means is able to find groups of users who use similar resources like IP, email, etc., while CopyCatch [3] is a more recent graph-based algorithm focusing on the user relationships.

CopyCatch [3]. In a graph with nodes representing users and the edges representing relationships, we want to find a big enough set that consists of fans who follow or like another group of idols. In addition, we want to find fans both following the same idol and sharing something, like the operating IP addresses. Starting from a random seed of users and updating the idols and fans alternatively, the algorithm can find many fraudulent groups. FraudVis helps verify and compare them.

2.3 Requirements for Fraud Visualization

In this work, we collaborate closely with both algorithms and domain experts in a start-up company focusing on fraud detection services. We collect their requirements for visualization and summarize their most demanding questions into four tasks:

Feature Selection. The quality of feature selection has a great impact on the algorithm performance, therefore we need to highlight the feature selection process to answer the question of "what are the most important features" for domain experts. Using the visual comparison between the feature distributions of legit and fraudsters, we can answer "what are the distributions of the important features", and "what is the difference among different user groups" for algorithm experts.

Temporal Analysis. We need to visualize temporal behaviors in different time spans within user groups to answer "what did they do as a fraud group" for domain experts, and highlight the patterns in the temporal activities to answer "do users in the same group share the same pattern" for algorithm experts. Meanwhile, we also need to analyze the temporal behavior differences among user groups.

Correlation Analysis. We need to obtain more insights on the correlation inside each user group to answer the question of "do users in the same group have similarities" for domain experts. With the assumption in Sec. 2 that a fraud group may be generated in the same way, we can answer the question "will members in one group build a characterized network" for algorithm experts.

Individual Analysis. We also need to explore the details on individual activities and user profiles to answer "is one user fraudulent or not" for domain experts. It is also necessary for algorithm experts to double check "did I get the wrong label for this user".

3 Related Work

Fraud detection is a special application of user clickstream analysis. Earlier researchers take lots of efforts to understand users' habits with clickstream data [16, 18, 23] using methods like Markov chains [2, 17] and clustering [31, 32] to capture the common behaviors. More advanced systems capture the context [35] or correlate both the temporal and spatial patterns [10]. Clickstream analysis has greatly helped people predicting users' intents [20] and make recommendations [37]. Visualization has helped greatly in user behavior study in different fields, such as education [6], medical services [19]. These visualizations help experts to better understand the abnormal user behaviors.

Different from abnormal behaviors from real users, we focus on frauds that are crafted to avoid detection. Popular algorithms detect the unusual grouping behaviors with two types of unsupervised learning methods. One type of approaches, such as CatchSync [13], LockInfer [14] and fBox [24], all detect dense subgraphs in the high dimensional feature space. Other types of approaches combine traditional clustering with clever feature engineering [5, 29]. There are also graphical-model-based learning approaches [33, 34, 36].

Visualization is even more crucial for fraud detection, as the fraud patterns are not always intuitive. People have previously proposed many fraud visualization systems such as EVA [15], Network Explorer [9] and so on. EVA visualizes the anomaly transactions of

a bank, and NE is a system for visualizing frauds in health care. Specifically, EVA mainly visualizes how a score system works and the raw data of bank transactions. EVA mentions that data mining techniques and visuali analytics techniques are commonly used but not supported by Visual Analytics techniques yet, which motivates the FraudVis design.

4 FRAUDVIS

FraudVis supports two kinds of workflows: 1) a *drill-down workflow* allowing users to navigate through the different FraudVis views, and 2) a customizable *dashboard* allowing human reviewers to take a high-level overview of the current frauds in the system. The bottom and the top part of Fig. 2 show these two workflows respectively. In this section we introduce the visual design of each view.

(a) Group Index. We represent each fraud group as a circle in a bubble view and use the radius of the bubbles to show the size of these groups. Clicking on a group leads user to the next stage.

(b) Group Data Inspector. It is customary for algorithm experts to start data exploration with the raw data [26], so we put a tabular view in the second step. We supplement the table by encoding the "importance" of different columns in different title colors. As we discussed, the more consistent a feature is, the more important it is. Mathematically, we calculate the Shannon entropy for each feature. We sort the columns in increasing entropy order and color the low entropy columns darker to attract reviewers' attention.

(c) Group Activity View. It is important to understand the aggregate behavior of a group over a time period. We create the activity view by integrating a pie-timeline chart, a flow chart and a bar chart. We use the pies in the pie-timeline chart to show the percentage of different activities (i.e., following a user or sending a gift), at different time periods (e.g., per day). We encode the event type as the color of the flow lines and the number of activities with the line thickness. When the user selects a specific pie, we display a larger version of the pie above the timeline, providing more details and highlighting the event type compositions. To better highlight the change of activities over time, we also have a bar chart summarizing the activity counts in each day.

(d) User Interaction Graph. In many social network applications, user interactions (e.g., who follows whom) are often crucial indicators of frauds. We illustrate these interactions based on the force chart. The nodes represent users, and the edges represent interaction events between users. We always color the source users who start this relationship in yellow, and the target users in blue. We use the color encoding instead of edge arrows to highlight the number of sources and targets. We generate one edge per event (i.e., a log entry), and use the edge color to encode the most important dimension contributing to the clustering result, for example, the source IP address. We see more consistent edge colors in fraud groups, indicating more obvious grouping behavior on certain features.

To inspect/compare different users and events, we have a side panel in the view. The users can choose nodes/edges from the graph to display detailed information in the side panel, and vice versa.

(e) Feature Selection View. Although many views above already provide insights about feature selection, for those who want to dig deeper, we summarize feature distributions of a single group in this view. As there are potentially many features, we only choose the top 10 features based on an anomalous score and plot them in that order. By default, we use the KL-divergence between a feature's distribution and the overall distribution as the score.

We use bar chart to show the differences in distribution. For each feature, we plot a greyed-out bar chart in cyan to show the distribution of all data (including fraud and non-fraud) and layer the distribution of the fraud group on top. To make the difference more distinguishable, we make the top layer yellow and flashing.

(f) Inter-group Comparison. We want to provide an intuitive overview of how good the algorithm works on different groups: i.e., whether the group is a dense cluster - the denser a cluster is, the more confident we are that it is fraud. To project the high-dimensional data onto the 2D display, we adopt the widely-used t-SNE [30], and use the KL-divergence between two users as the distance metric.

To better illustrate a group, as references, we also plot four other fraud groups with the most similar size, as well as a random sample of non-fraud data points onto the same figure. These references provide user with a visual scale that how "concentrated" the cluster is. Obviously the legit users scatter around the figure, while different fraud groups have different concentration.

(g) Individual Analysis. Human experts want to focus on a single user from time to time. In this view, we try to show all the details about a single user: a timeline view to illustrate all his activities. Also, we train a decision tree using the algorithm output as the ground truth and highlight the user's decision path on the tree. Although the decision tree is not how we perform the detection, some human experts still find it insightful on explaining certain results.

Dashboard. For the dashboard, in addition to a customizable page where users can choose which views to display on the same page, we have a timeline-inter-group view (*h* in Fig. 2). This view is similar to the inter-group comparison view discussed above, but adds a timeline, allowing users to select time spans of their interests. We evaluate the t-SNE parameters at the first time span and use it for all time spans. That is, if all features of a fraud event are the same, then the point will not move. Using the timeline, users can discover the recurring patterns of different fraud groups.

5 CASE STUDIES

We present two real-world case studies, a Youtube-like video website with 300k users, three million logs in two weeks, and an Amazon-like e-commerce site with 30k logs in four days. The video logs focus on user interactions including follow, gift, like and comment. There is also non-personally-identifiable account information such as gender and email types (i.e., public or company email). We believe it is a representative social media application.

In contrast, for the e-commerce site, the user only cares about detecting fake account registrations. We only use logs from the registration page for four days. Each log entry describes a new user registration, such as the phone number, IP address, timestamp, device and the time spent on the registration page.

In both case studies, we take the detection results from algorithm experts, visualize them in FraudVis, and ask the domain experts to review them. Here we summarize the key findings.

5.1 Case Study 1: Social Video Website

The algorithm experts choose the CopyCatch algorithm [3] for fraud detection. Fig. 1 shows the FraudVis visualization of different views for a single fraud group. Our key observations are:

Silence and burst. Using the group activity view (*c* in Fig. 1), we can clearly observe that the fraudsters in the group have no activities for some time, before issuing many "follow" actions all on a single day. After that they go back to sleep again for two days, and then continue with follow or like activities. A domain expert can be quite sure these "silence and burst" behaviors indicate some click farms, and the group activity view clearly reveals the behavior.

Correlation Property For Intra-group. A domain expert can quickly tell what goes wrong from the user interaction graph (*d* in Fig. 1). There are two "weird" things of this group: 1) there are only four sources (yellow) nodes, but a moderate number of targets (blue) nodes; and 2) The edges in the graph encode different IP addresses. There are only three edge colors, meaning that all these actions are done from three IP addresses. Examining these three addresses (mouse over), we find them even within the same subnet. Again, we are confident these activities are all from a click farm.

Feature Selection For Intra-group. To see which feature looks the most diverse from the overall distribution (*e* in Fig. 1), we can look

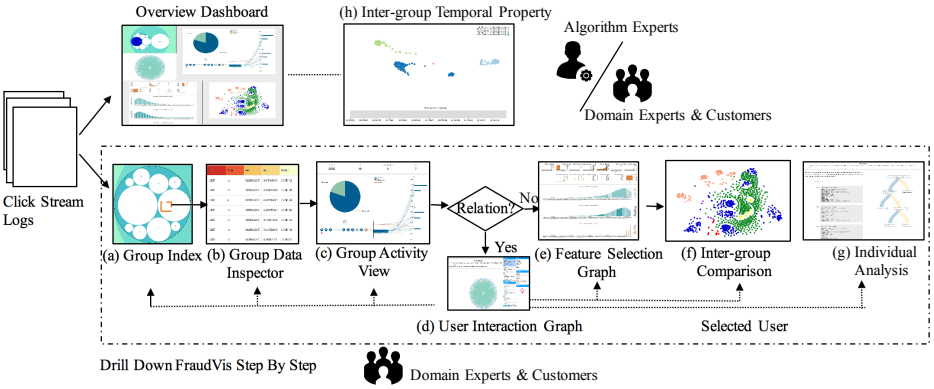
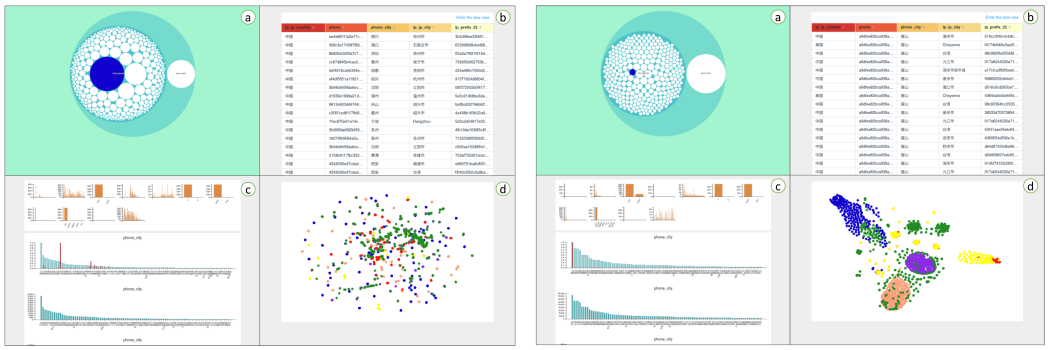


Figure 2: The workflow of FraudVis. We offer two pipelines for different target users, an overview for experts and a set of more detailed views for domain experts and customers to drill down FraudVis with friendly instructions.



(a) stock k-means (b) improved

Figure 3: Result comparison between stock k-means and improvements in Case Study 2.

at the feature selection view. FraudVis automatically chooses the *time shot* feature, i.e., the activity count at different time periods. We can see that the distribution of a legit user group exhibits a steady number of activities over time, however, again, the fraud group exhibits "silence and burst" behavior.

Inter-group Analysis. Inter-groups comparisons(*f* in Fig. 1) indicate the clustering quality. We can see the legit (green) users scatter around the graph sparsely, while fraud groups cluster more tightly together, indicating that the nodes are indeed similar in certain dimensions.

The timeline-inter-group view (*h* in Fig. 2) is also insightful. We can see that some fraud groups become active at similar times too. For example, on the morning of August 12th, members of 50% groups had some activities. Co-occurrence of different groups might indicate that they are under single control.

5.2 Case Study 2: E-commerce Website

Limited by space, we only present a comparison between two detection algorithms for this case study: the stock k-means and an improved version. Taking a randomly chosen fraud group as an example, the feature selection view shows that the area code of the phone number is the most important feature where the fraud users concentrate on.

Comparing k-means and improved results (Fig. 3(a) and Fig. 3(b)), we clearly see three pieces of evidence showing that the k-means results are worse: 1) the distribution of the most important feature hardly differs from the overall distribution (*c* in Fig. 3(a)); 2) In the inter-group comparison view, members from different groups are mixed together and hard to separate; and 3) there is a large fraud group containing many users that are not similar (*d* in Fig. 3(a)). In comparison, Fig. 3(b) shows the results from the improved version, whose grouping results look much more consistent.

6 Conclusion

Existing researchers have come up with hundreds of algorithms to detect suspicious fraudsters from operation logs. Companies also define rules, develop algorithms, or hire people to discover online frauds. There are two main pain points for both academic research and industrial adoption: 1) how to explain the fraud behavior to domain users with little technology background? 2) how to test the result of various fraud detection algorithms and discover the fundamental features contributing the most to the detection?

Working closely with experts in the fraud detection area, we have designed FraudVis to tackle these two problems. The system is achieved by synthesizing the knowledge learned from the underlying mechanism that generates the fraudulent accounts at the underground black market. Our work provides a fresh view and a working system to display high-dimensional fraud behaviors and visually interpret the result of unsupervised fraud detection algorithms.

For the future work, we think the design of FraudVis is not fine-grained enough. For two target users, we try to care about both of them. However, for algorithm experts, they want to know more about how the feature selection process works, and even need to interact with feature selection process timely. Thus, we may need to customize a more fine-grained design for algorithm experts.

Acknowledgments

This research is supported in part by the National Natural Science Foundation of China (NSFC) grant 61532001, Tsinghua Initiative Research Program Grant 20151080475, MOE Online Education Research Center (Quantong Fund) grant 2017ZD203, and gift funds from Huawei and Ant Financial.

REFERENCES

[1] L. Akoglu, R. Chandy, and C. Faloutsos. Opinion fraud detection in online reviews by network effects. 2013.

[2] F. Benevenuto, T. Rodrigues, M. Cha, and V. Almeida. Characterizing user behavior in online social networks. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*, IMC '09, pp. 49–62. ACM, New York, NY, USA, 2009. doi: 10.1145/1644893.1644900

[3] A. Beutel, W. Xu, V. Guruswami, C. Palow, and C. Faloutsos. Copycatch: Stopping group attacks by spotting lockstep behavior in social networks. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pp. 119–130. ACM, New York, NY, USA, 2013. doi: 10.1145/2488388.2488400

[4] R. J. Bolton and D. J. Hand. Statistical fraud detection: A review. *Statistical Science*, 17(3):235–249, 2002.

[5] Q. Cao, C. Palow, C. Palow, and C. Palow. Uncovering large groups of active malicious accounts in online social networks. In *ACM Sigsac Conference on Computer and Communications Security*, pp. 477–488, 2014.

[6] Y. Chen, Q. Chen, M. Zhao, S. Boyer, K. Veeramachaneni, and H. Qu. Dropoutseer: Visualizing learning patterns in massive open online courses for dropout reasoning and prediction. In *2016 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 111–120, Oct 2016. doi: 10.1109/VAST.2016.7883517

[7] B. Davis and W. Conwell. Methods and systems to help detect identity fraud, Dec. 20 2006. US Patent App. 11/613,891.

[8] F. H. Glancy and S. B. Yadav. A computational model for financial reporting fraud detection. *Decision Support Systems*, 50(3):595–601, 2011.

[9] J. A. Guerra-Gomez, A. Wilson, J. Liu, D. Davies, P. Jarvis, and E. Bier. Network explorer: Design, implementation, and real world deployment of a large network visualization tool. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '16, pp. 108–111. ACM, New York, NY, USA, 2016. doi: 10.1145/2909132.2909281

[10] X. Han, L. Wang, S. Xu, G. Liu, and D. Zhao. Linking social network accounts by modeling user spatiotemporal habits. In *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pp. 19–24, July 2017. doi: 10.1109/ISI.2017.8004868

[11] 2015 Internet Crime Report. `https://pdf.ic3.gov/2015_IC3Report.pdf`.

[12] M. Jiang, P. Cui, A. Beutel, C. Faloutsos, and S. Yang. Catchsync: Catching synchronized behavior in large directed graphs. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pp. 941–950. ACM, New York, NY, USA, 2014. doi: 10.1145/2623330.2623632

[13] M. Jiang, P. Cui, A. Beutel, C. Faloutsos, and S. Yang. Catchsync: catching synchronized behavior in large directed graphs. pp. 941–950, 2014.

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

[15] R. A. Leite, T. Gschwandtner, S. Miksch, S. Kriglstein, M. Pohl, E. Gstrein, and J. Kuntner. Eva: Visual analytics to identify fraudulent events. *IEEE Transactions on Visualization and Computer Graphics*, PP(99):1–1, 2017. doi: 10.1109/TVCG.2017.2744758

[16] Z. Liu, Y. Wang, M. Dontcheva, M. Hoffman, S. Walker, and A. Wilson. Patterns and sequences: Interactive exploration of clickstreams to understand common visitor paths. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):321–330, Jan 2017. doi: 10.1109/TVCG.2016.2598797

[17] L. Lu, M. Dunham, and Y. Meng. Mining significant usage patterns from clickstream data. In *Proceedings of the 7th International Conference on Knowledge Discovery on the Web: Advances in Web Mining and Web Usage Analysis*, WebKDD'05, pp. 1–17. Springer-Verlag, Berlin, Heidelberg, 2006. doi: 10.1007/11891321_1

[18] A. L. Montgomery, S. Li, K. Srinivasan, and J. C. Liechty. Modeling online browsing and path analysis using clickstream data. *Marketing Science*, 23(4):579–595, Sept. 2004. doi: 10.1287/mksc.1040.0073

[19] J. H. Park, S. Nadeem, S. Mirhosseini, and A. Kaufman. C2a: Crowd consensus analytics for virtual colonoscopy. In *2016 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 21–30, Oct 2016. doi: 10.1109/VAST.2016.7883508

[20] J. Y. Park, N. O'Hare, R. Schifanella, A. Jaimes, and C.-W. Chung. A large-scale study of user image search behavior on the web. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pp. 985–994. ACM, New York, NY, USA, 2015. doi: 10.1145/2702123.2702527

[21] B. A. Prakash, M. Seshadri, A. Sridharan, S. Machiraju, and C. Faloutsos. Eigenspokes: Surprising patterns and scalable community chipping in large graphs. In *2009 IEEE International Conference on Data Mining Workshops*, pp. 290–295, Dec 2009. doi: 10.1109/ICDMW.2009.103

[22] M. Ruiz-Sanchez. Methods and apparatus for protecting against credit card fraud, check fraud, and identity theft, Apr. 19 2002. US Patent App. 10/125,645.

[23] S. Senecal, P. J. Kalczynski, and J. Nantel. Consumers' decision-making process and their online shopping behavior: a clickstream analysis. *Journal of Business Research*, 58(11):1599–1608, 2005.

[24] N. Shah, A. Beutel, B. Gallagher, and C. Faloutsos. Spotting suspicious link behavior with fbox: An adversarial perspective. In *IEEE International Conference on Data Mining*, pp. 959–964, 2014.

[25] G. Stringhini, C. Kruegel, and G. Vigna. Detecting spammers on social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference*, ACSAC '10, pp. 1–9. ACM, New York, NY, USA, 2010. doi: 10.1145/1920261.1920263

[26] D. R. Thomas. A general inductive approach for analyzing qualitative evaluation data. *American Journal of Evaluation*, 27(2):237–246, 2006. doi: 10.1177/1098214005283748

[27] K. Thomas, D. McCoy, C. Grier, A. Kolcz, and V. Paxson. Trafficking fraudulent accounts: The role of the underground market in twitter spam and abuse. In *USENIX Security Symposium*, pp. 195–210, 2013.

[28] K. Thomas, D. Yuxing, H. David, W. Elie, B. C. Grier, T. J. Holt, C. Kruegel, D. McCoy, S. Savage, and G. Vigna. Framing dependencies introduced by underground commoditization. In *In Proceedings (online) of the Workshop on Economics of Information Security (WEIS*. Citeseer, 2015.

[29] T. Tian, T. Zhang, T. Zhang, T. Zhang, and T. Zhang. Crowd fraud detection in internet advertising. In *International Conference on World Wide Web*, pp. 1100–1110, 2015.

[30] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[31] G. Wang, T. Konolige, C. Wilson, X. Wang, H. Zheng, and B. Y. Zhao. You are how you click: Clickstream analysis for sybil detection. In *Proceedings of the 22Nd USENIX Conference on Security*, SEC'13, pp. 241–256. USENIX Association, Berkeley, CA, USA, 2013.

[32] G. Wang, X. Zhang, S. Tang, H. Zheng, and B. Y. Zhao. Unsupervised clickstream clustering for user behavior analysis. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pp. 225–236. ACM, New York, NY, USA, 2016. doi: 10.1145/2858036.2858107

[33] L. Xiong, B. Pczos, and J. Schneider. Group anomaly detection using flexible genre models. *Advances in Neural Information Processing Systems*, pp. 1071–1079, 2012.

[34] L. Xiong, B. Pczos, J. G. Schneider, A. Connolly, and J. Vanderplas. Hierarchical probabilistic models for group anomaly detection. *Journal of Machine Learning Research*, 15:789–797, 2011.

[35] H. Yin, B. Cui, L. Chen, Z. Hu, and Z. Huang. A temporal context-aware model for user behavior modeling in social media systems. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, SIGMOD '14, pp. 1543–1554. ACM, New York, NY, USA, 2014. doi: 10.1145/2588555.2593685

[36] R. Yu, X. He, and Y. Liu. Glad:group anomaly detection in social media analysis. *Acm Transactions on Knowledge Discovery from Data*, 10(2):1–22, 2015.

[37] T. Zhao, M. Hu, R. Rahimi, and I. King. It's about time! modeling customer behaviors as the secretary problem in daily deal websites. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 3670–3679, May 2017. doi: 10.1109/IJCNN.2017.7966318