Discovery of Rating Fraud with Real-Time Streaming Visual Analytics

Kodzo Webga and Aidong Lu

Abstract—The rating fraud in online e-commerce stores targets at receiving large revenues through boosting the popularity of selected items with fake ratings. The challenges of detecting rating frauds come from discovering small scale abnormal activities in a large amount of data and detecting frauds in a time-critical manner from online rating streams. This paper presents a real-time visual analytics system that consists of two essential components: a server for automatically handling data streams and a visual analytics interface for performing interactive analysis. Based on the features of rating frauds, we present a detection solution which balances computationally expensive algorithms and interactive analysis between the server and analysts. Specifically, our detection system filters data through performing an initial suspicion level detection on the server, and analysts can combine different statistical analysis of the user / item matrix through a co-mapped singular value decomposition (SVD) diagram, re-ordered matrix representation, and the temporal view. We demonstrate our approach with case studies of different fraud scenarios and show that rating frauds can be effectively detected.

Index Terms—Rating fraud, fraud detection, security visualization, streaming visualization

1 Introduction

Online e-commerce stores such as Amazon and eBay are widely available and have formed an important part of our everyday lives. Nowadays almost everything can be purchased online, such as movies and books. The online stores often accept ratings from users and use them to recommend related items. As higher ranks on the leaderboard generally lead to a larger number of downloads and yield more revenue, it has become more and more frequent that some merchants inflate their sale records or post phony product ratings [30]. This is usually implemented by generating a variety of fake users to boost the sale records, ratings or reviews in a very short time. Related to the rating frauds, similar frauds have been deployed in social networks with different formats, including ranking fraud, web ranking spam and online review spam [30, 24, 26].

The challenges of detecting rating frauds come from the difficulties of identifying suspicious activities from the perspectives of user, item, and time duration separately. It often requires a detection mechanism to assist analysts to explore abnormal patterns by combining the statistical features from all relevant perspectives. Also, the product ratings in online stores usually come as individual records in real-time, while fake ratings are hidden in a large amount of rating records purposely. It requires an efficient mechanism to handle the rating streams and filter the data for raising situational awareness of suspicious activities promptly.

Most existing visual analytics approaches are for time-varying datasets and are more concentrated on the history of rating records. While they may be applicable to explore historical patterns, they are not suitable for handling streaming data and performing time-critical detection tasks. Current streaming visualization approaches often concentrate on visualizing temporal trends of data, which may be hard for analysts to identify subtle traces of frauds in a large amount of rating records. In short, a successful detection approach of the rating fraud requires an effective mechanism to process and filter real-time data and a powerful set of analytics tools to perform interactive detection tasks thoroughly.

In this paper, we present a customized streaming system to address the challenges of rating fraud detections. Our streaming system contains two essential components: a server for handling data streams and a visual analytics interface for analysts to perform interactive analysis. We balance the workload between the two components by having the server to process, filter, and store rating records of selected time windows and perform computationally expensive algorithms by taking advantage of the computation powers. The visual analytics interface is light on the storage and computation by only keeping relevant data records and processed results, while it provides a set of tools for analysts to perform a detailed analysis. Analysts are alarmed only when suspicious activities are found on the server and they can request history data records and computation results through the interface directly, thus the visual analytics side can be performed on most platforms without strong computation capabilities.

We further summarize the rating streams as $M \times N$ matrices with M users and N items. The problem of rating fraud can be formalized as finding abnormal patterns from $M \times N \times T$ rating records in a time-critical manner. To explore the statistical features of the rating matrices effectively, we adopt Singular Value Decomposition (SVD) which is a widely used dimension reduction algorithm. We design the interface of visual analytics with three different types of visual patterns: a co-mapped SVD diagram for visualizing the correlations among the separate high-dimensional spaces in the SVD results; a matrix representation for visualizing rating records with interactive analysis methods, including SVD and interaction supported re-ordering approaches; and a time view for visualizing the changes of ratings and supporting free selection of time durations to explore. We demonstrate the performance of our streaming platform with results and case studies from different simulated and real fraud scenarios.

The remainder of the paper is organized as follows. We first summarize the related work on online e-commerce fraud detection, streaming visualization, and SVD-based approaches in Section 2. Section 3 provides an overview and design rationale of our streaming system. Sections 4 and 5 describe the technical details of the server and visual analytics interface respectively. Section 6 provides case studies and discussions of our approach. Finally, we conclude this work and describe future work in Section 7.

2 Related Work

This section briefly summarizes three main research topics related to our contributions in this paper: online fraud detection, streaming visualization, and related SVD visualization approaches.

2.1 Online Fraud Detection

Related to the rating fraud, the field of data mining has developed a number of approaches to detecting various online fraud activities, in-

- *Kodzo Webga is with University of North Carolina at Charlotte. E-mail: kwegba1@uncc.edu.*
- *Aidong Lu is with University of North Carolina at Charlotte. E-mail: aidong.lu@uncc.edu.*

978-1-4673-7599-3/15/$31.00

cluding web ranking spams [24], online review spams [26], and ranking frauds in mobile App recommendation [30].

To the best of our knowledge, the visual analytics approaches to detecting the rating fraud are rare, although they are similar to some existed network security problems [23, 29, 10, 8, 25]. For example, network attacks often appear in real-time and are carried out in a time duration, which contains similar data streams as the rating fraud. Many network security applications also require time-critical detections, where an effective streaming system is necessary as well. The statistical approaches for identifying network patterns, when using the matrix representation, can be shared between detecting network attacks and rating frauds. However, the data format of online e-commerce fraud is a non-square matrix between users and items, which is different from the majority network security applications.

2.2 Streaming Visualization

Due to the space limit, we only concentrate on the most recent work on streaming visualization instead of related time-varying visualization, which does not necessarily require time-critical solutions. Streaming visualization generally provides a real-time visualization that is updated with data streams automatically, such as the visual sedimentation approach using different sedimentation effects to visualize the flow of real-time data [13]. Fischer et al. [9] developed a real-time visual analytic system using distributed processing and Spark framework, which analyzed streams with stream slices to enhance situational awareness. Chin et al. [6] studied the effectiveness of dynamic trees and tree-maps, spiral time lines, and graphs for visualizing data stream based on the structure of the dataset. Bach et al. [4] visualized dynamic networks with a matrix cube, which represented each time step as a matrix. Li and Baciu [16] designed a web framework for dynamic visualization of large streaming data by serializing the original data into multiple streams to be contained on current hardware. Yang et al. [27] demonstrated a stream system which applied innovative multi-query strategies to compute popular patterns for visual analytics. Also, a set of taxonomy for dynamic data analysis has been proposed for real-time network dataset visualization [3, 7].

2.3 Related SVD Visualization Approaches

Singular Value Decomposition (SVD) is a widely used dimension reduction algorithm. It decomposes a matrix into a sequence of combined rows and columns features. While SVD algorithms have been popular in matrix factorization and recommendation algorithms, SVD spaces are not convenient for user interaction and needs to be modified for visual analytics. For example, Lu et al. [17] proposed a similarity-based re-ordering method for parallel coordinates according to the combination of Nonlinear Correlation Coefficient and SVD. Hou et al. [11] constructed a 2D visual representations by mapping user queries and text information retrieval results using SVD. Quille et al. [21] used a SVD-based analytical process to identify communities and outstanding elements in the DBLP dataset. Luo et al. [18] used hyperbolic and multi-modal view to visualize the recommendation list.

Related to product rating and recommendation algorithms, most approaches are from data mining and machine learning fields. For example, Kermarrec et al. [15] used SVD-like matrix factorization and PCA for global mapping of movie ratings from high dimensions to two dimensional space. Similar to the existing approaches, our SVD diagram also modifies the original SVD spaces for interactive visualization. In particular, we propose a co-mapped diagram for exploring the interactions among the related SVD dimensions.

3 Overview

As illustrated in figure 1, our system contains an automatic processing component on the server side and an interactive visualization interface on the user end. To handle streaming data effectively, the server component handles data streams and heavy computations including the SVD, while the visualization side serves as an interface for analysts to explore selected time windows with a set of visual analytics tools.

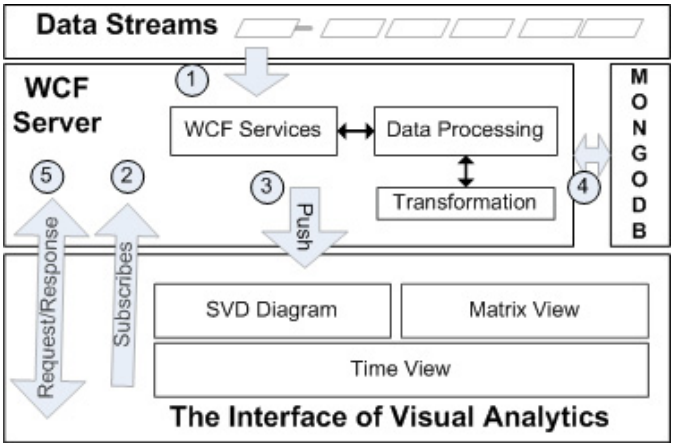


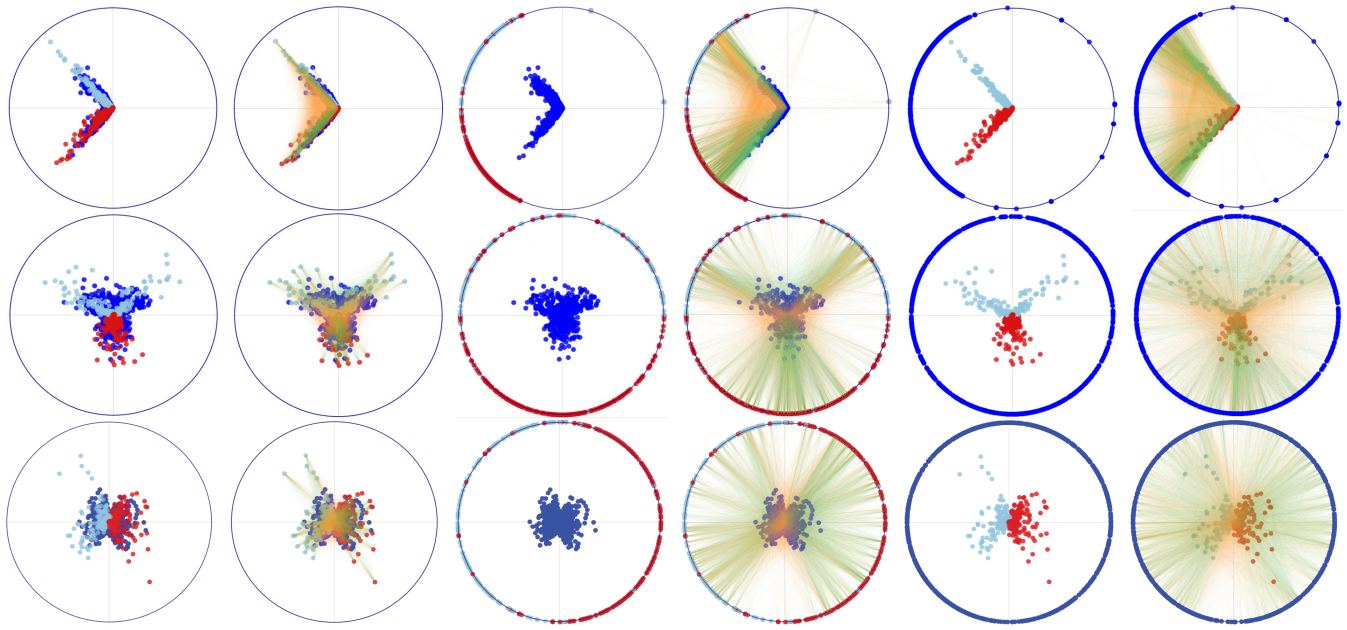
Fig. 1: The architecture of the streaming system. We balance the workload between the server and the interface of visual analytics. The server handles all computationally expensive operations and the interface provides a rich set of tools for visual analytics on selected time windows.

For detecting the rating fraud, the server filters the data streams with initial processing and assessment of suspicion level. According to different suspicion levels, the server determines a need to store or discard the data trunks, and to increase or decrease the suspicion level threshold on the interface of visual analytics. While analysts can interact with the data streams in real-time using the interface, they are required to perform further analysis only when alerted by the server. Specifically, the server component performs the following important tasks.

- Taking the stream feeds and generating data trunks to send to the matrix view on visualization interface based on a selected size of time window.
- Collecting and processing data. The server combines all the records within a time window and builds an accumulated matrix. With the accumulated matrix, the server first performs basic operations, including evaluating the average ratings for each user and item. This information is sent to the time view on the visualization interface to update in real-time.
- Performing a basic suspicious activities detection through measuring two differences from adjacent time windows for each item: average ratings and the numbers of rating activities.
- Sending an alert to the visualization interface requesting further investigation from the analysts, once the suspicion level is raised above a threshold. The server computes the SVD for the cumulated time window and sends the result to the SVD view on the visualization interface.
- Determining if the data from the current time window should be stored for further analysis based on the suspicion level.
- Taking the requests of users from the visualization interface and sending the required data trunks back.
- Performing SVD algorithm on the accumulated matrix of the required data trunks and sending the results back.

On the interface of visual analytics, analysts can perform interactive analysis with patterns from three different domains.

- SVD spaces for exploring combined features of users and items.
- Matrix view for visualizing the accumulated rating records at different orderings.
- Time view for observing the changes of items on the top rating list and selecting time ranges for investigation.



(a) SVD spaces of both U and V | (b) (a) with edges | (c) Co-mapped SVD with items outside | (d) (c) with edges | (e) Co-mapped SVD with users outside | (f) (e) with edges

Fig. 2: Examples of co-mapped SVD diagrams. Each row shows different mappings of one dataset. Comparing the co-mapped SVD diagrams in columns (c) - (f) with the direct SVD projections in columns (a) and (b), the co-mapped SVD diagrams avoid the misinterpretation of mapping independent U and V spaces directly, reveal edges between users and items with less clutters, and preserve the grouping information on the circle layout. For example, the two outlier movies shown in (d) of the first row are hidden in (a) and (b) and the connections between groups of users and items are generally better revealed in (c)-(f) than the original projections in (a) and (b).

4 User Interface for Visual Analytics

This section describes our user interface for visual analytics. We have selected three types of domains to visualize rating streams and detect the rating fraud. Among which, a time view is often a must for observing temporal changes of data streams. Based on the properties of a non-square matrix of rating records, we design a co-mapped SVD diagram for studying relationships between users and items. In addition, we include a matrix view for quick updates of the rating records in real time and interactive analysis of accumulated rating records.

4.1 Co-Mapped SVD Diagram

The rating streams can be summarized as an $A = M \times N$ non-square matrix. To study various data features between the M users and N items, we have explored several dimension reduction algorithms that are adopted in the data mining field. Among which, SVD is a popular matrix factorization algorithm and it has been widely used in collaborative filtering for recommendation approaches [14]. The SVD decomposition of A takes the form

$$A = USV^{\mathrm{T}} \tag{1}$$

where U and V are orthogonal matrices, U is $M \times M$ matrix, and V is $N \times N$ matrix. S is the diagonal matrix with the same dimensions as the original matrix A. The locations of items in the matrix U represent their relationships in A with close distances for strong similarities; the same as the locations of users in the matrix V.

Previous approaches to visualize SVD are mainly node-link diagrams from user-selected dimensions. However, one main challenge of visualizing SVD spaces is that two high-dimensional spaces, U and V, represent the latent features of the rows and columns in spaces; while the S space presents the significance of SVD dimensions. As shown in Figure 2, we layout the U and V spaces in different manners - one space in the original projection and the other space in the outside circular layout. We choose to not overlay the spaces of U and V to avoid misinterpretation of the relationships between users and items. The co-mapped SVD layout still allows visual exploration of the same SVD dimensions from U and V spaces jointly, as they correspond to the same combination of data features.

The rationale behind the circular layout is that similar to eigenspaces [12], the quasi-orthogonal patterns [28] are shown in the SVD spaces. By projecting nodes from the SVD space to the circular layout, it preserves the clustering information and transforms the line patterns to clustered groups. Also, the nodes with the central role in a cluster are preserved in the circular layout. On the contrary, random nodes may be projected on any random locations on the circular layout. For example, the items of two categories are colored differently in Figure 2 and their grouping patterns from (a) are preserved in (c) by locating at different portions of the outside circular layout. The users in the original SVD space shown in (a) are generally accumulated near the center because of their strong similarities; they are scattered on the circular layout in (e) according to their preferences over the rated items.

Specifically, we first sort all the SVD dimensions according to S values, with the highest S values indicating the most important dimension. Analysts can choose any two SVD dimensions and the mapping type of the projection / circular layout on the SVD diagram. The projection is achieved according to the angle of a node in the original SVD subspace Θ_{svd} and the distance to the Origin of SVD space $distance_{svd}$ as follows:

$$\Theta_{cl} = \Theta_{svd} + rand() \times 2\pi/(1 + distance_{svd}) \tag{2}$$

where Θ_{cl} is the projected angle and $rand()$ generates a random number between 0 and 1. The nodes with larger distance are projected closer to the original angle, and they remain at the center of the clustered group on the circle layout. These nodes are often the users or items with a large amount of ratings. They are also often the targets or results of the rating frauds.

For dimension reduction algorithms, other approaches are possible choices as well, such as eigen-decomposition. However, for non-square matrices of the ranking records, we need to fill a large number of non-existing ratings with zeros in the adjacency matrix to perform

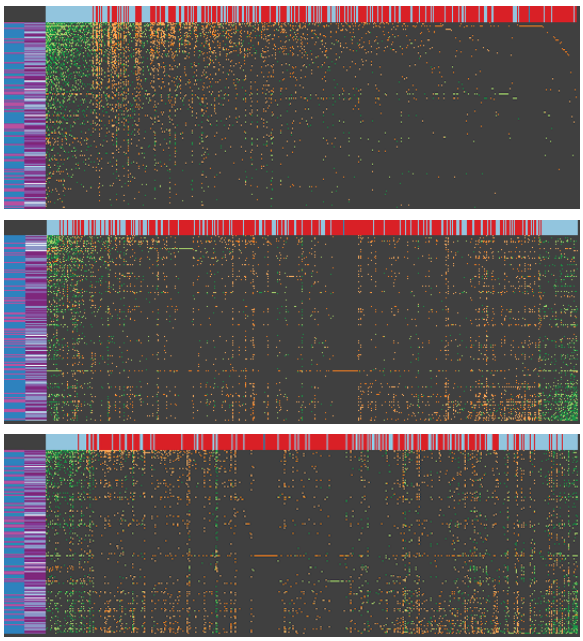


Fig. 3: Matrix representation. The same accumulated user and item matrix reordered based on different SVD dimensions (first, second, and the fifth) for both row and column. Different orderings reveal clearly different matrix patterns and can be combined with the SVD diagram for interactive analysis.

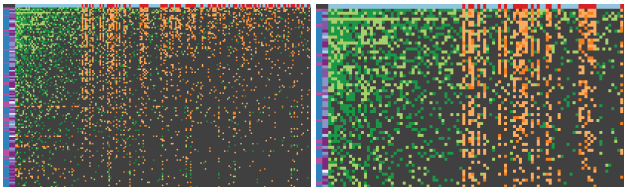


Fig. 4: Matrix scaling effect - 5 pixel per cell and 10 pixel per cell respectively.

eigen-decomposition. Also, SVD is selected as it is widely used in collaborative filtering algorithms for recommendation - a popular application of rating datasets. Exploring SVD dimensions may allow us to combine both collaborative filtering algorithms and visual analytics approaches in the future.

4.2 Matrix Representation

The matrix representation visualizes the accumulated rating matrix along with additional attributes of users and items. The purpose of the matrix representation are two folds: one is to provide an efficient way to update the stream feeds; the other is to reveal group patterns for items of similar rating history and users of similar behaviors, including fake users.

As shown in Figures 3, 4 and 5, we provide three types of interactions to enable visual analytics with the matrix representation.

- Scaling - allowing analysts to focus on a portion of the matrix, as the accumulated matrix expands significantly with time.
- Highlighting - allowing analysts to highlight one node or a group of nodes by making selection on the SVD diagram and moving selections to the front of the rows for users or columns for items.
- Reordering - allowing a joint exploration with the SVD diagram by ordering the row or column with node sequences on specified SVD dimensions.

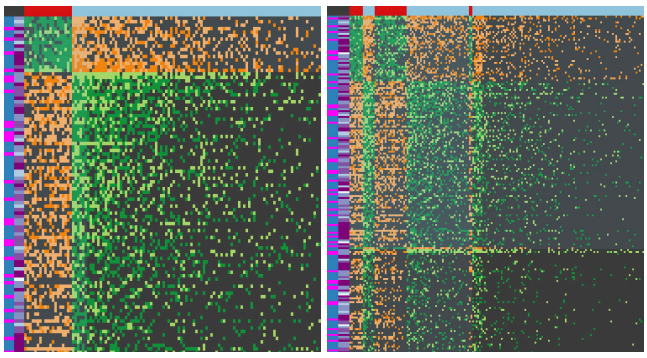


Fig. 5: Matrix interactively reordered by grouping nodes clustered in the SVD diagram to reveal potential rating patterns of groups, such as the two groups with opposite tastes on the left matrix. The lighted shadow on the right matrix indicates the selection of groups.

4.3 Time View

For detecting the rating fraud, we use the time view to visualize important changes of the rating records. The suspicion level of each time window is also shown on the time view. To update the information from data streams efficiently, we visualize the top list of rating changes, which often captures the start of rating fraud during the interactive investigation. The top rating list of each time window is shown as a vertical bulletin board and the time view is automatically filled and updated with data streams. To highlight any suspicious activities, we also adjust the transparency values of items on the bulletin board, which helps to identify the time periods of rating frauds.

As the time view is for users to visualize the changes of ratings, we assign colors for items on the time view with a fast hashing algorithm. The hashing algorithm maps the unique item's ID code to a fixed index in a pre-selected color table with 20 different colors (e.g. the 20 color categories of D3 [1]). While there are many options for achieving the color mapping, we choose the fast hashing as it is required for every update of the time window.

$$index = ID \bmod size\_of\_color\_table \tag{3}$$

As human vision cannot perceive the differences of all colors, it is inevitable that different items may have the same perceived color. The pre-selected color table contains a set of qualitative colors that users can distinguish well and only has a small amount of overlap randomly on our top rating list.

The key interaction provided on the time view is for users to select a time range by combining multiple time windows. For example, a duration with time windows of both normal and high suspicion levels may capture a fraud activity performed by several groups of users jointly. The visualization interface sends the selected time range to the server, requests SVD to be performed on all the saved data trunks, and updates the SVD diagram with the received results from the server.

5 Streaming System Design and Implementation

In this section, we describe our streaming system for receiving streaming datasets, processing server operations, and sending data to the visualization interface. The system on server side has two groups of functionalities: one for stream data services and the other for computations and handling requests from analysts.

5.1 Data Streaming Services

In order to satisfy the key design considerations, we have chosen an emerging technology that is widely used in industries, Windows Communication Foundation (WCF), which is flexible to develop distributed, scalable, and cross-platform services. WCF helps to build services that can be configured to use different transfer protocols without any extra changes to the core of the services by just adding endpoints with the appropriate protocol. For example, the same service

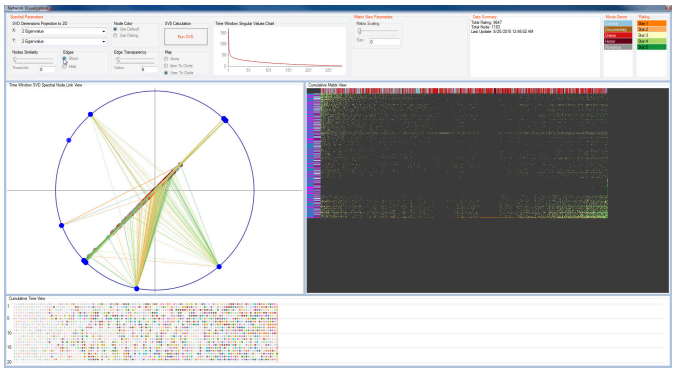


Fig. 6: User interface for visual analytics. There is a parameter bar on the top of the interface in additional the SVD diagram (middle left), matrix (middle right), and the time view on the bottom. The parameter bar includes the functions to choose SVD dimensions and adjust visualization options for the co-mapped SVD diagram, the curve of S values for guiding the selection of SVD dimensions, and color attributes of item categories and ratings for both the co-mapped SVD diagram and the matrix representation. During the procedure of interactive detection, analysts can often start with any outlier patterns on the SVD diagram, make hypothesis through adjusting the visualization options of the SVD diagram and the matrix view, until finally reach the conclusion by revealing distinctive patterns. The time view shows the top 20 items with the largest changes of average ratings for the recent time windows. Once a suspicious activity if detected, the time view attracts the attention of analysts by highlighting the time window with a vertical orange line. This example has the highlight line on the last time window, indicating that the snapshot is captured when the first suspicion level is raised.

can be called using .Net Remoting and ASMX web service technology [19]. For data storage, we use MongoDB [2] because of its agility, scalability, and performance. MongoDB is document-oriented database that makes it easy to store data of any structure and modify the schema dynamically.

For efficiency, we use the subscriber-publisher pattern to implement the data exchange between the visualization and the server. When the stream data arrives, any active filtering is applied and the desired data is queued into a stream pipeline. Based on the time window, the data is processed and sent to the visualization side as an adjacency matrix and a list of average ratings of users and items. Because it is a time consuming process, SVD is computed only if suspicious activities are found or upon the request of analysts.

Specifically, the data flows on the server are as follows. At the start, the client - visualization interface - subscribes to the data stream service by sending its session ID to the server. The data stream reader enqueues each record from the stream to the data stream queue. The data push service then de-queues the data and splits it into chunks with the maximum customized package size allowed and sends it to the visualization space using WSDualHttpBinding protocol. This guaranties the in-order deliverability and resilience [20]. Any request/response that involves data transmission between the server and the client is done using the NetTcpBinding protocol.

During the transfers of data, the last chunk is tagged to signal the end of the transfer for the client. On the visualization side, we customize the reading of the pushed data chunks to check for the tagged signal. This way, the visual analytics interface can accurately manage the updates of data streams. Analysts can also request for history rating records by subscribing to the raw data.

5.2 Data Processing and Computation

An important function of the server is automatic filtering. We apply a simple yet effective assessment of suspicion level detection by comparing the average rating and historical rating records of items, as most

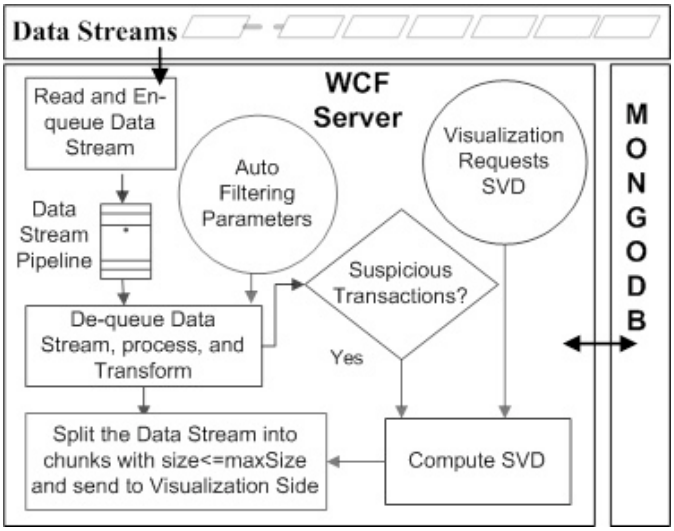


Fig. 7: The data flow on the streaming server.

fraud activities tend to involve extreme high or low ratings. Once the changes of average ratings or the numbers of activities of any item reaches a user-assigned threshold, the server marks the suspicion level and applies corresponding filtering mechanisms automatically. Data related to suspected users and items in suspicious activities are given special priority. Historical records beyond certain time ranges under the threshold can be filtered out to save storage. The automatic filtering function can also be turned off to store all the rating records.

The server automatically saves the reduced and transformed data in the matrix format into the MongoDb database when suspicious activities are detected. A copy of the processed data is also kept in memory for some time before it is progressively discarded. When the visualization interface requests data, such as SVD result of a given period of time, the server first checks the availability in memory. This technique can save time for server from repeatly accessing the external storage even if the requested data is not available. Based on the time frame of requested data, the server merges all available data records, computes the SVD, and sends the results back to the visualization space.

For requests involved of long computation time durations, the server sends an acknowledgment and frees the visualization interface from waiting for an instant response. Once the server finishes the computations, it sends the results back to the visualization interface. This technique ensures that the visualization interface is responsive for smooth user interaction and visual analytics.

6 Results and Case Studies

This section describes example results, case studies, and quantitative results. Two following datasets are used in our experiments.

- The MovieLens100K dataset [22] contains 100K ratings from 1 to 5 and 1682 movies from different categories rated by 943 users. Each user has rated at least 20 movies during the data period. The ratings are from September 19th, 1997 through April 22nd, 1998.

- The Amazon video game reviews dataset has 228,570 users and 21,025 products, and 463,669 ratings. The rating are based on 1 to 5 scales. In this study, we have removed unknown users and kept 364,541 ratings.

The rating frauds are added to the first dataset by simulating real rating fraud scenarios that change in scales, rating behaviors, and time durations. The second dataset is slighted downsized for a real-life case study.

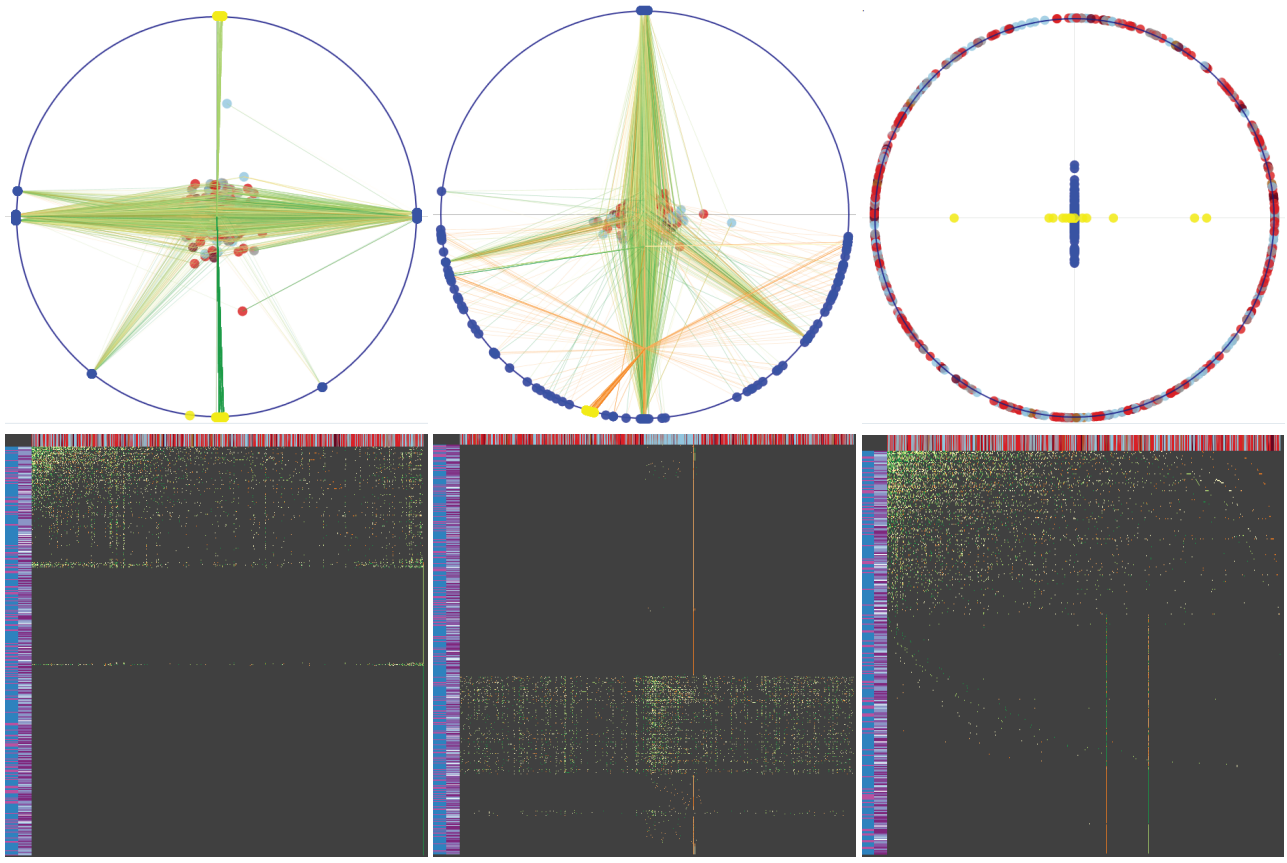


Fig. 8: Example results of fraud detection. The top row shows the co-mapped SVD diagrams and the second row shows the reordered matrices. The suspicious users are marked in yellow color in the SVD diagrams. The case 1 and case 2 project users on the outside circular layout, which reveals substantial edge patterns - green clusters in case 1 and red clusters in case 2 - that are related to the rating frauds. The case 3 projects items on the circular layout, which demonstrates an orthogonal pattern of normal users in blue and suspicious users in yellow. By applying the SVD-based recording strategy on the matrix view, we can also successfully reveal the group patterns of rating fraud for all the three cases.

6.1 Example Results

We simulate different scenarios of rating frauds and perform interactive detection with our system. The following describe three different cases and the resulted patterns from our system.

Case 1: Fake users rate one target movie high and others very low.

As the results shown on the first column of Figure 8, the matrix is reordered based on the third dimension of SVD to single out the target movie and fake users. The SVD diagram maps users on the circular layout and reveals the fraudulent item.

Case 2: Fake users rate one target movie very low while other movies randomly.

As the results shown on the second column of Figure 8, we can observe similar patterns as in Case 1. First, reordering the matrix based on the first dimension singles out the target movie and related users. Second, mapping the users on the circular layout of SVD diagram in the first and second dimensions also shows the target movie and relevant pattern.

Case 3: Groups of users with opposite ratings: one group rates one movie very high and another movie very low while the opposite group rates the first movie very low and the second movie very high.

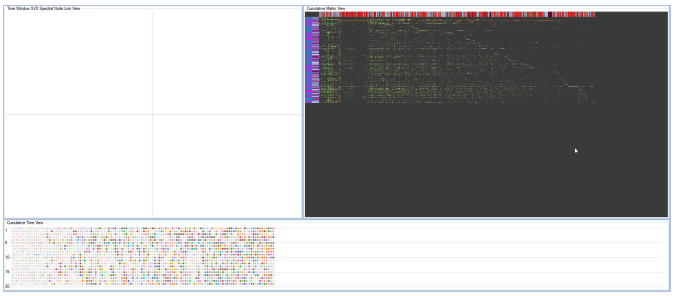
As the results shown on the third column of Figure 8, we can observe patterns suggesting a competition between two groups of users in their rating behaviors. First, reordering the matrix based on the second dimension singles out the target movies and related users. Second, projecting the data on first and second dimensions and mapping the users on the circular layout show the target movies and relevant patterns.

6.2 Case Study 1: MovieLens100k for rating fraud detection

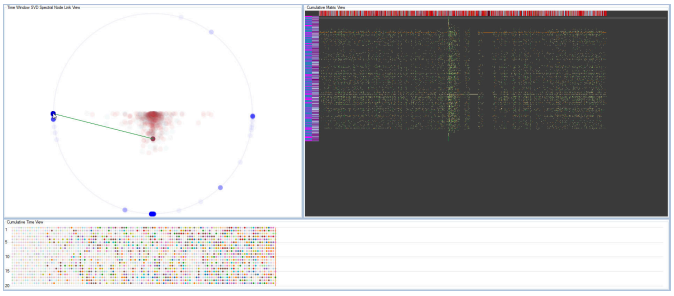
In this case study, we filtered out the unpopular movie genres and focused on the most popular ones with 40,000 ratings. We then purposely added 400 fake users who rated a particular movie very high (4 to 5) within a short time range but in a sporadic manner. A portion of fake users gave low ratings to other movies randomly.

We run our system by first starting the stream simulator. Once it is up, we start our WCF service then launch the visualization interface. The first row of Figure 9 shows the interface when there is no alert from the server. At this stage, analysts can still request SVD results from the command panel if they find any interesting patterns from the matrix view or the time view for further investigation.

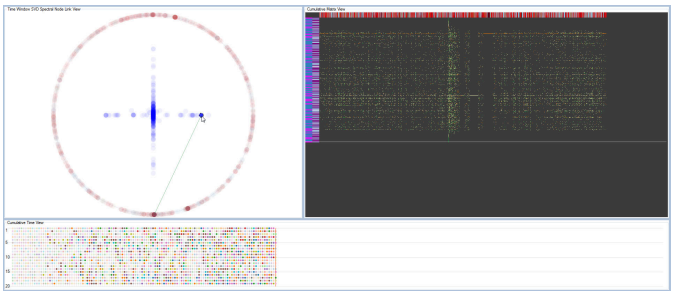
The second row of Figure 9 shows the beginning of a possible fraud signal by the server. This is notifiable on the time view with an orange background for the bulletin list of the new time windows. The SVD diagram is also updated and the matrix is automatically reordered based on the first dimension of SVD. In addition, the cumulative counting of total nodes (users and movies) and the total number of rating records are shown on the visualization interface. The top 20 movies with the highest ratings are also shown on the time view. These functions help the analysts to monitor the stream data. When we observe the alert from the server, we start to explore the feature dimensions that contain less clusters on the co-mapped SVD diagrams. We choose S values that are around both ends of the S curves. With a few attempts, we identify the dimensions that cluster majority users as several groups. By selecting groups and setting the similarity rendering threshold to 100 percent, users with similar behaviors are singled out and high-



Before suspicious activity is found



Interactive exploration with the interface



Suspicious users are identified.

Fig. 9: Case study 1 - The three snapshots capture important time windows before any suspicious activity is found, interactive exploration with the interface, and the users suspected in fraudulent activities are identified. During the second phase, by reducing the number of clusters for users and movies and reordering the matrix view, two groups of users are revealed.

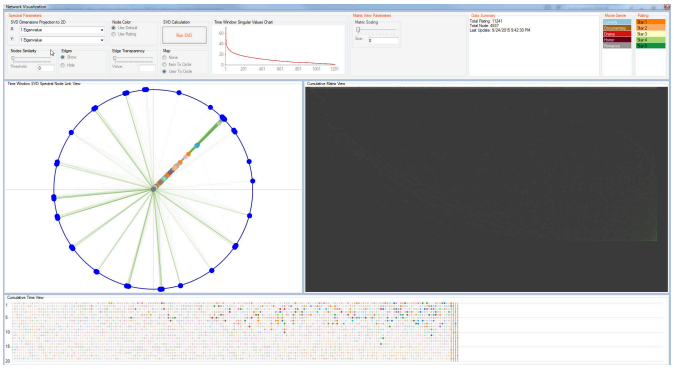
lighted in the matrix view. At the end, we return to the first feature dimension and are able to identify the target movie as well, as shown on the third row of Figure 9.

6.3 Case Study 2: Amazon reviews of video game for rating fraud detection

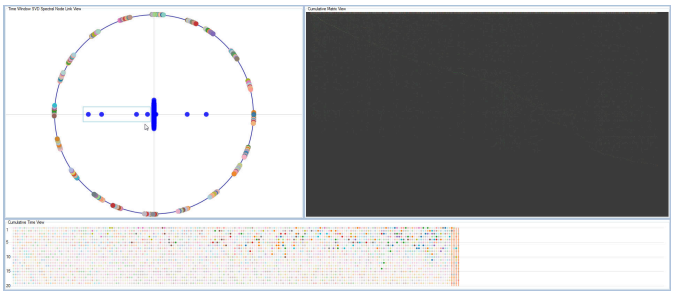
In this study, we use the Amazon reviews of video game dataset without adding any fraud activities and the goal is to test if our system can detect any rating fraud from a real-life dataset. We set the node colors of the video games in SVD diagram to be the same as the time view, as there is no information of game category. We run the streaming simulation using a randomly generated stream feed which speed is between five seconds and one minute. We then identify an appropriate time window to assess the suspicion level on server, around 20 ratings per time window.

At the beginning, the matrix view is very sparse and we cannot observe any significant patterns. On the time view, we observed that before the alert from the server (the first row of Figure 10), there were several video games rated by more than 5 users that created more than 0.5 changes in the average rating from the previous average rating records. This can be observed by the visible nodes on the time view, as the games with higher rating changes and larger rating records are more visible than others.

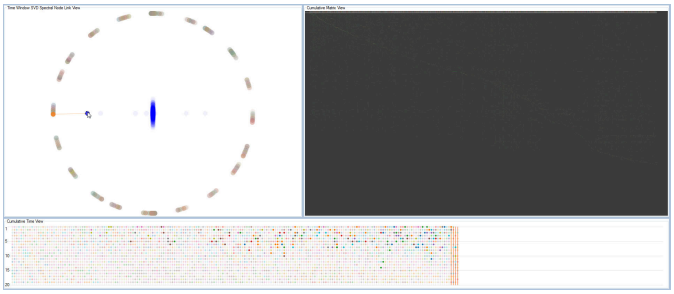
We started the investigation by interacting first with the SVD diagram. We explored different dimensions from the SVD features space to select dimensions with less clusters between the users and the video games. The second row of Figure 10 shows the result after we selected suspicious users from the SVD diagram and reordered them in the matrix view. We can further observe suspicious video games on the top of the matrix view and visualize their rating changes prior to the alert on the time view window. The third row of Figure 10 shows the video games rated in short time range negatively (ratings between 1 to 2) by users who did not rate any other video games. Although analysts need to draw the final conclusion if this is a real rating fraud, this case study demonstrates that our system helps to identify such suspicious activities in real-life scenarios.



Suspicious alert sent from the server



Interactions with SVD diagram and matrix view



Suspicious users are identified

Fig. 10: Case study with a real-life dataset - The three snapshots capture important time windows when suspicious activity is found on the server, interactive exploration with the interface, and the suspicious users and the video games are identified. During the second phase, analysts revealed more suspicious nodes on the top of the matrix by selecting nodes in different groups and reordering the matrix view according to the selection. During the last phase, the suspicious users from the SVD diagram are identified and highlighted in the matrix view.

6.4 Quantitative Results

The overall performance of our system is tested. We run both WCF server and the visual analytic interface on a Dell Optiplex 980 machine with Intel Core i7 at 2.93 GHz processor. It has Windows 7 Enterprise

64-bit as an operation system,12.0 GB of RAM installed, 4 physical Cores and 8 Logical Processors. Although it is hard to evaluate every single algorithm in the system in term of time complexity, we have evaluated the key components of the system and ensured interactive performance.

We generate a streaming simulator to feed the dataset to our system at different ranges of data speed (three seconds to one minute) and amount (1,000 to 20,000 ratings). The maximum delay from the time that the server reads the stream data to the time that the visual analytic interface receives the data is 659 milliseconds. On the server side, the SVD decomposition is the most computationally extensive operation. It takes around 2.5 seconds for a matrix with 2,500 nodes and 100,000 ratings and 3 seconds for a matrix of 6,000 nodes and 15,000 ratings. We separate SVD from streaming services and only perform it when suspicious activities are found or requested by analysts.

The visual analytic interface also handles the user interactions smoothly. For a stream data feed with size from 1000 to 20,000 ratings, it only takes 70 milliseconds for the time view and 500 milliseconds for the matrix view on average to update.

The rendering of the edges on the SVD diagram is the only time-consuming component, as the rendering time increases with the number of rating records, around 8 seconds for 100,000 ratings. However, most time windows do not contain such a large amount of rating records and analysts can switch between nodes and edges interactively. Edges can also be filtered based on the rating values, rating records of items, and user selections for interactive performance.

7 Conclusion and Future Work

In this work, we present a streaming visual analytics system to detect the rating fraud for online e-commerce networks. The platform contains two essential components for handling data streams, a customized WCF server for receiving and processing data streams and a visual analytics interface for performing interactive analysis. With the goal of detecting rating frauds, the two components are designed from different aspects of identifying suspicious time windows, users, items, and statistical patterns in different domains. The system provides an effective rating fraud detection approach through balancing the workload of computationally expensive algorithms and interactive analysis of abnormal activities.

In the future, we could integrate streaming SVD approaches such as [5] to pre-compute information from selected users and items so that they are ready for interactive analysis. We plan to expand our streaming platform to take online streams and develop heterogeneous multi-processing solutions to handle data streams of various types. We also plan to extend our platform to handle a set of frauds existed in online commercial networks, including the web ranking spams and online review spams.

Acknowledgments

We wish to thank Yuemeng Li for pointing us to SVD approaches and the reviewers for their valuable comments. This work was supported in part by U.S. National Science Foundation (CCF-1047621).

References

[1] http://d3js.org/.

[2] http://www.mongodb.com/.

[3] B. Bach, E. Pietriga, and J.-D. Fekete. Graphdiaries: Animated transitions and temporal navigation for dynamic networks. *Visualization and Computer Graphics, IEEE Transactions on*, 20(5):740–754, May 2014.

[4] B. Bach, E. Pietriga, and J.-D. Fekete. Visualizing dynamic networks with matrix cubes. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 877–886. ACM, 2014.

[5] M. Brand. Fast low-rank modifications of the thin singular value decomposition. *Linear Algebra and its Applications*, 415(1):20–30, 2006. Special Issue on Large Scale Linear and Nonlinear Eigenvalue Problems.

[6] G. Chin, M. Singhal, G. Nakamura, V. Gurumoorthi, and N. Freeman-Cadoret. Visual analysis of dynamic data streams. *Information Visualization*, 8(3):212–229, 2009.

[7] J. Cottam, A. Lumsdaine, and C. Weaver. Watch this: A taxonomy for dynamic data visualization. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pages 193–202, Oct 2012.

[8] T. K. Dang and T. T. Dang. A survey on security visualization techniques for web information systems. *International Journal of Web Information Systems*, 9(1):6–31, 2013.

[9] F. Fischer and D. A. Keim. Nstreamaware: real-time visual analytics for data streams to enhance situational awareness. In *Proceedings of the Eleventh Workshop on Visualization for Cyber Security*, pages 65–72. ACM, 2014.

[10] L. Harrison and A. Lu. The future of security visualization: Lessons from network visualization. *Network, IEEE*, 26(6):6–11, November 2012.

[11] J. Hou, Y. Zhang, J. Cao, and W. Lai. Visual support for text information retrieval based on matrix's singular value decomposition. In *Web Information Systems Engineering, 2000. Proceedings of the First International Conference on*, volume 1, pages 344–351 vol.1, 2000.

[12] X. Hu, A. Lu, and X. Wu. Spectrum-based network visualization for topology analysis. *Computer Graphics and Applications, IEEE*, 33(1):58–68, Jan 2013.

[13] S. Huron, R. Vuillemot, and J.-D. Fekete. Visual sedimentation. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2446–2455, 2013.

[14] D. Kalman. A singularly valuable decomposition: the svd of a matrix. *The college mathematics journal*, 27(1):2–23, 1996.

[15] A.-M. Kermarrec and A. Moin. Data Visualization Via Collaborative Filtering. Research report, Inria, Feb. 2012.

[16] C. Li and G. Baciu. Valid: A web framework for visual analytics of large streaming data. In *Proceedings of the 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*, TRUSTCOM '14, pages 686–692, 2014.

[17] L. F. Lu, M. L. Huang, and T.-H. Huang. A new axes re-ordering method in parallel coordinates visualization. In *Machine Learning and Applications (ICMLA), 2012 11th International Conference on*, volume 2, pages 252–257, Dec 2012.

[18] H. Luo, J. Fan, D. A. Keim, and S. Satoh. Personalized news video recommendation. In *Advances in Multimedia Modeling*, pages 459–471. Springer, 2009.

[19] A. Mackey. Windows communication foundation. In *Introducing. NET 4.0*, pages 159–173. Springer, 2010.

[20] N. Pathak. *Pro WCF 4: Practical Microsoft SOA Implementation*. Apress, 2011.

[21] R. V. E. Quille, C. Traina, Jr., and J. F. Rodrigues, Jr. Spectral analysis and text processing over the computer science literature: Patterns and discoveries. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, SAC '14, pages 653–657, 2014.

[22] G. Research. Movielens100k: Movie rating dataset.

[23] H. Shiravi, A. Shiravi, and A. Ghorbani. A survey of visualization systems for network security. *Visualization and Computer Graphics, IEEE Transactions on*, 18(8):1313–1329, Aug 2012.

[24] N. Spirin and J. Han. Survey on web spam detection: Principles and algorithms. *SIGKDD Explor. Newsl.*, 13(2):50–64, may 2012.

[25] D. Staheli, T. Yu, R. J. Crouser, S. Damodaran, K. Nam, D. O'Gwynn, S. McKenna, and L. Harrison. Visualization evaluation for cyber security: Trends and future directions. In *Proceedings of the Eleventh Workshop on Visualization for Cyber Security*, VizSec '14, pages 49–56, 2014.

[26] S. Xie, G. Wang, S. Lin, and P. S. Yu. Review spam detection via temporal pattern discovery. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 823–831, 2012.

[27] D. Yang, Z. Guo, Z. Xie, E. A. Rundensteiner, and M. O. Ward. Interactive visual exploration of neighbor-based patterns in data streams. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, SIGMOD '10, pages 1151–1154, 2010.

[28] X. Ying and X. Wu. Graph generation with prescribed feature constraints. In *In Proc. of the 9th SIAM Conference on Data Mining*, 2009.

[29] Y. Zhang, Y. Xiao, M. Chen, J. Zhang, and H. Deng. A survey of security visualization for computer network logs. *Sec. and Commun. Netw.*, 5(4):404–421, Apr. 2012.

[30] H. Zhu, H. Xiong, Y. Ge, and E. Chen. Discovery of ranking fraud for mobile apps. *Knowledge and Data Engineering, IEEE Transactions on*, 27(1):74–87, Jan 2015.