Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

# RCAnalyzer: Visual Analytics of Rare Categories in Dynamic Networks*

Jiacheng Pan†[1], Dongming Han[1], Fangzhou Guo[1],
Dawei Zhou[2], Nan Cao[3], Jingrui He[2], Mingliang Xu[4 5], Wei Chen†‡[1]

[1]*State Key Lab of CAD & CG, Zhejiang University, Hangzhou, Zhejiang, P.R. China, 310058*

[2]*Department of Computer Science and Engineering, Arizona State University, Arizona*

[3]*Tong Ji Intelligent Big Data Visualisation Lab (iDVx Lab), TongJi University, Shanghai, P.R. China*

[4]*The School of Information Engineering, Zhengzhou University, Zhengzhou, P.R. China, 450000*

[5]*Henan Institute of Advanced Technology, Zhengzhou University*

†E-mail: panjiacheng@zju.edu.cn; chenvis@zju.edu.cn

Received mmm. dd, 2016; Revision accepted mmm. dd, 2016; Crosschecked mmm. dd, 2017

**Abstract:** A dynamic network refers to a graph structure whose nodes and/or links will dynamically change over time. Existing visualization and analysis techniques mainly focus on summarizing and revealing the primary evolution patterns of the network structure. Little work focuses on detecting anomalous changing patterns in a dynamic network, the rare occurrence of which could damage the development of the entire structure. In this paper, we introduce the first visual analysis system RCAnalyzer designed for detecting rare changes of sub-structures in a dynamic network. The proposed system employs a rare category detection algorithm to identify anomalous changing structures and visualize them in context to help oracles examine the analysis results and label the data. In particular, a novel visualization is introduced, which represents the snapshots of a dynamic network in a series of connected triangular matrices. Hierarchical clustering and optimal tree cut are performed on each matrix to illustrate the detected rare change of nodes and links in the context of their surrounding structures. We evaluate our technique via a case study and a user study. The evaluation results verified the effectiveness of our system.

**Key words:** Rare Category Detection; Dynamic Networks; Visual Analytics
https://doi.org/10.1631/FITEE.1000000 **CLC number:** TP

## 1 Introduction

In many cases, relations among objects can be modeled as time-evolving networks, such as the collaborations among researchers, transactions among traders, and communications in social networks. These relations reflect how individuals act in a network over time and reflect the goals of their activities (Jovanovic et al., 2015). Most individuals in a network behave normally, while a minority may act differently from the others, indicating anomalous situations. Anomalies could be positive, such as superstars in a collaboration network and recipients or benefactors in a financial network, or negative enough to damage the development of the entire graph, such as frauds in a trading network and criminals or spies in a communication network. In either case, finding these anomalous changing behaviors of network structures is valuable.

Most of the existing anomaly detection algorithms are automatic, and do not take human insights into account. In contrast, active learning is a special case of machine learning that improves auto-

‡ Corresponding author

* This research is supported by National Natural Science Foundation of China (U1866602,61772456)

ORCID: Jia-cheng PAN, https://orcid.org/0000-0002-8676-9990

matic algorithms' performance with human knowledge. Following an active learning procedure, many rare category detection (RCD) methods are thus developed following (He and Carbonell, 2009, 2008; Huang et al., 2013, 2011; Pelleg and Moore, 2005), i.e., candidates that are most likely to represent rare categories are detected and shown to be labeled by users. Rare category detection methods are one set of anomaly detection algorithms which recognize abnormal individuals as rare categories because their number is usually very small. Once labeled, the algorithm will propagate the label to the nearby instances which are similar to the labeled one in a feature space. Those representative candidates are usually centers of rare categories. This procedure has one major limitation, i.e., it is still difficult for users to make a correct judgment (i.e., whether or not the candidate represents a rare category) by only showing one single data instance to them with the entire context information missing. This is particularly difficult for detecting rare categories from a dynamic graph as both the temporal and structural information need to be considered while labeling a candidate. Therefore, visualization could be helpful in terms of supporting the interactive data exploration and providing a rich context representation.

However, challenges exist in designing such a visualization system to support the process of rare category detection in a dynamic network. First, although capturing the temporal dynamics of a changing structure itself is a problem that has been extensively studied (Beck et al., 2014), none of the existing techniques is developed to support the visualization of rare categories. Second, capturing the changing structures of rare categories in the context of a big dynamic graph is challenging as the rare categories are usually very small and their evolutions could be very likely to be ignored. Third, to better support the decision-making process, the visualization should be able to differentiate different structures in detail, and this is not easy to achieve.

To address the above challenges, in this paper, we propose a novel visualization system called RCAnalyzer. RCAnalyzer represents a large dynamic network in the form of a series of connected triangular matrices with each matrix representing a snapshot (Fig. 1). A hierarchical clustering algorithm and a tree cut algorithm are developed to produce an adaptive focus+context view that aggregates the graph structure into a hierarchy so that a large graph can be fully displayed while showing the detailed structures of potential rare categories. The proposed matrix based visualization facilitates an in-context visual comparison of substructures in a dynamic graph, thus helping with rare category detection. In particular, this paper has the following contributions:

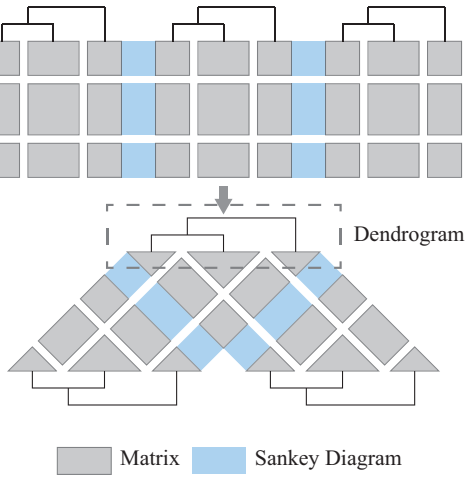


**Fig. 1 The basic design of the matrices view is a combination of matrix, Sankey diagram and dendrogram. Compared to a square matrix, triangles are more space efficient.**

- A novel tree cut algorithm that produces a multi-focus view to illustrate the substructure details of multiple rare categories in the context of a big dynamic graph.
- A novel dynamic network visualization design in the form of a series of connected triangular matrices that highlights the detected rare categories in both the temporal and topological context, facilitating the substructure comparison.
- An integrated visual analysis system that supports the detection of rare categories and facilitates rare category labeling.

The paper is organized as follows. Related work is discussed in section 2. The BIRD algorithm and analytical tasks are introduced in section 3. In section 4 we introduce the design of our system. System evaluations are introduced in section 5. We discuss our work in section 6 and conclude the paper in section 7.

## 2 Related work

### 2.1 Dynamic network anomaly detection

Anomaly detection in dynamic networks refers to the detection of anomalous nodes, edges, subgraphs, and time-evolving changes. Several existing surveys have reviewed the most popular anomaly detection methods used in dynamic networks (Bhuyan et al., 2013; Ranshous et al., 2015). Ranshous et al. categorized the existing methods into 5 types (Ranshous et al., 2015): community-based, compression-based, decomposition-based, distance-based, and probabilistic-model-based. For example, based on compression based methods, a graph stream can be divided into multiple segmentations using the minimum description length (MDL) principle. Anomaly changes can be then detected at the time points when a new segment begins (Sun et al., 2007). Probabilistic-model-based methods usually construct a "normal" model and use it to detect anomalies that deviate from the "normal" model. For example, when the number of communications deviates from the expected number generated by conjugate Bayesian models, the time point would be considered as an anomaly (Heard et al., 2010).

As we mentioned in Section 1, these anomaly detection works do not capture user's intention. In contrast, rare category detection refers to a series of active learning methods which incorporate human knowledge. Many RCD methods requires prior information to detect the minority classes (Pelleg and Moore, 2005; He and Carbonell, 2008; He et al., 2008, 2010; Zhou et al., 2015a, 2017, 2015b). However, many data sets don't have any prior information. To avoid this limitation, Huang et al. (Huang et al., 2011, 2013), He et al. (He and Carbonell, 2009) presented a series of prior-free methods. Compactness-assumption-based methods (He and Carbonell, 2008; He et al., 2008; Zhou et al., 2015b, 2017) assume that the distribution of the major categories is smooth and compact and compactness-isolation-assumption-based methods (Huang et al., 2013; Vatturi and Wong, 2008) require the rare categories to be isolated from the major category. Lin et al. present RCLens (Lin et al., 2017), a visual analytics system supporting user-guided rare category exploration and identification. RCLens is able to support users identify rare categories in a high dimensional dataset. However, it is not designed for rare category identification in dynamic networks.

### 2.2 Visualization of anomaly

Many visualization techniques have been developed to help the detection and analysis of anomalies (Haberkorn et al., 2014; Liu et al., 2017; Chandola et al., 2009; Zhang et al., 2017). Dimension reduction methods, such as principal component analysis(PCA) (Jolliffe and Ian, 1986), and multidimensional visualization techniques, such as parallel coordinate plots (Inselberg, 2009) and DICON (Cao et al., 2011), are commonly used to visualize the data distribution and show outliers with abnormal distribution. In ViDX (Xu et al., 2017), an extended Marey's graph is used to show outliers in the manufacturing procedure. Anomalies in network traffic data (Corchado and Herrero, 2011; Tsai et al., 2009; Teoh et al., 2002) and social media data (Thom et al., 2012; Zhao et al., 2014; Cao et al., 2016) have also drawn a lot of attention. Fluxflow (Zhao et al., 2014) detects the diffusion of anomalous information in social media and TargetVue (Cao et al., 2016) uses glyph-based designs to show the anomalous behaviors in online communication systems based on an unsupervised learning model. Wang et al. (Wang et al., 2013) presented SentiView to visualize the sentiment in internet topics and enables analysts to monitor abnormal events on the internet. Fan et al. (Fan et al.) presented an interactive visual analytics approach which combines active learning and visual interaction to detect anomalies.

Compared to the existing methods, our method focuses on detecting the rare categories in dynamic networks based on RCDs. To the best of our knowledge, there isn't an existing visualization system that supports labeling users in analyzing and labelling anomalies based on RCDs. Moreover, we developed a series of interactions which enable users to compare rare categories within entire dynamic networks.

### 2.3 Visualization of dynamic networks

Visualization of dynamic networks has had a lot of study over the years. A fine survey by Beck et al. (Beck et al., 2014) has reported the state of art of dynamic network visualization. Beck et al. classify the visualization techniques of dynamic networks into animated diagrams (Bach et al., 2013; Yee et al., 2001) and timelines of a series of static

charts, such as node-link diagrams or adjacency matrices. Timelines with matrix-based and flow-based representation methods are most relevant to our work. Archambault et al. (Archambault et al., 2011) found that small multiple-based techniques have better performance than animation-based techniques.

Matrix-based techniques can be classified into two categories. The first category embeds a timeline into each cell of the matrix. Gestaltlines (Brandes and Nick, 2011), fingerprint glyphs (Oelke et al., 2013), and the horizon graph (Burch et al., 2013) are used to show the evolution of dyadic relations in a matrix. However, this category of methods often does not fit well with large data sets. The second category lays a sequence of adjacency matrices in a certain order (Bach et al., 2015, 2014; Zhao et al., 2015). Van den Elzen et al. (Elzen et al., 2015) reduce the matrices into points and lay the points by production methods. NodeTrix (Henry et al., 2007) and Dendrogramix (Blanch et al., 2015) both visualize a static graph by combining several visualization representation. However, they are not designed for visualizing dynamic networks and thus cannot show the change of networks properly.

Flow-based techniques use flow metaphors to represent the evolution of communities in networks (Vehlow et al., 2015; Hlawatsch et al., 2014). Sankey diagram (Riehmann et al., 2005) and ThemeRiver (Havre et al., 2000) are the most common methods used. For example, Vehlow et al. (Vehlow et al., 2015) use Sankey diagrams to show the changes of community structures. Flow-based techniques aggregate networks by group information, and thus often lack details of the local areas of the network.

In this paper, we combine adjacency matrices, Sankey diagrams, and tree structures based on a multi-focus tree cut algorithm and visualize focused areas with fine-grained detail and unfocused areas with coarse-grained detail within a sequence of matrices.

## 3 Overview

Rare category detection (RCD) algorithms aim to find an initial example of rare classes in the data (Pelleg and Moore, 2005). To best of our knowledge, Batch-update Incremental RCD (BIRD) (Zhou et al., 2015b) is the first (and the only) work designed for detecting rare categories in dynamic networks. It takes snapshots of dynamic network topology at two different time steps as input and iteratively detects *rare category candidates*, which potentially belong to a rare category. In this section, we first introduce related concepts of BIRD, and then introduce the analytical tasks users should complete based on RC-Analyzer to detect rare categories in dynamic networks.

### 3.1 Batch-update incremental RCD (BIRD)

Here, we review the key ideas of the incremental rare category detection algorithm - BIRD (Zhou et al., 2017, 2015b), which pave the way for our forthcoming introduction of the rare category visual analytic system.

The Batch-update incremental Rare Category Detection (BIRD) algorithm aims to detect rare categories in dynamic networks. According to BIRD, a pair of nodes is closely connected if their transition probability is high. Therefore, the BIRD algorithm believes the transition probability of nodes in one rare category should have a lower bound and the transition probability of nodes in different rare categories should have an upper bound (He et al., 2008). Therefore, a rare category is a group of connected nodes that possess the following two features: (1) These nodes form a compact structure, which means they are closely connected. The transition probabilities among these nodes are relatively high and larger than the lower bound. (2) The compact structure should have a clear border. The transition probabilities among the nodes in this structure (rare category) and the other rare categories are relatively low and smaller than the upper bound. There are two visual examples showing these two features intuitively in Fig. 2.

BIRD is an iterative algorithm. In each iteration, it detects a node whose neighborhood density changes significantly between two given adjacent time steps in a dynamic network. This node is potentially a representative node of a rare category.

Similar to the existing graph-based RCD algorithms (He and Carbonell, 2008; He et al., 2008; Zhou et al., 2015a), the BIRD algorithm can be mainly separated into the following two parts:

1. Compute the global similarity matrix $A$,

$$A = (I - \alpha W)^{-1} \tag{1}$$

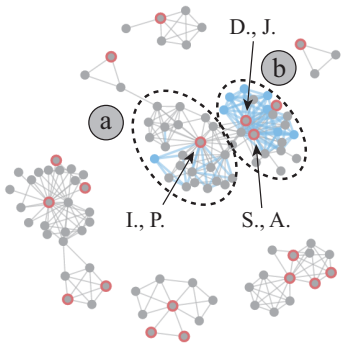


**Fig. 2 The compact neighborhood structures of D., J. and S., A. (A) and I., P. (B).**

where $I$ is an identity matrix, $W$ denotes the transition probability matrix of the given graph $G$, and $\alpha$ is a positive discounting constant in the range of $(0, 1)$. Note that the global similarity matrix $A$ helps sharpen the changes of the local density near the boundaries of each class. This considerably reduces the workload of identifying rare categories in the query process.

2. Update the query score iteratively based on the labeling information from users and return the example with the largest query score to users for inspection. In general, the query process selects the examples from regions where local density changes the most, and thus the queried examples tend to have a high probability of hitting the regions of rare categories.

Before algorithm BIRD (Zhou et al., 2017, 2015b), previous studies (Pelleg and Moore, 2005; He and Carbonell, 2008; He et al., 2008, 2010) were all built for static graphs. For this reason, BIRD extends the problem to the dynamic setting and efficiently updates the RCD model by using the local changes to avoid reconstructing it from scratch. To be specific, the BIRD algorithm (1) efficiently updates the global similarity matrix $A^{(t)}$ at each time step $t$ based on the global similarity matrix $A^{(t-1)}$ at previous time step $t-1$ and the updated edges in current time step $t$; (2) locally updates the query scores of the examples which may be infected by the changes in current time step $t$.

The original BIRD algorithm outputs the rare category candidate with the highest query score and waits for users to label the candidate. The query process might repeat many times. Thus, we slightly modify the BIRD algorithm by making the algorithm output candidates with top $k$ query scores, where $k$ is a manually set parameter.

The workflow of analyzing rare categories in dynamic networks with BIRD contains three stages. First, users set parameters and select two adjacent snapshots to initialize BIRD. Second, users analyze and identify rare categories based on the candidates detected by BIRD. Third, users label the candidates. The label result is returned to BIRD. When users think that all rare categories between the two snapshots are found, they can select other time steps and repeat the workflow to analyze other rare categories.

### 3.2 Analytical tasks

According to the analysis workflow, we summarize what analytical tasks should be completed by users based on these data as follows:

T1 Set parameters to initialize BIRD. Users need to set a series of parameters before BIRD can detect rare category candidates. The most important parameters are the starting time step and the ending time step, which determine $G_t$ used for initialization of BIRD.

T2 Identify new rare categories from the examples detected by BIRD. After BIRD is initialized, it will iteratively output detected rare category candidates. Users first identify candidates that truly belong to rare categories by analyzing their neighborhood structure. Then users compare the detected rare category with labeled rare categories to determine whether it is a new rare category.

T3 Label the examples based on analysis results. After analyzing rare category candidates, users label each candidate by a specific number. Labels are then returned to BIRD.

## 4 System design

In this section, we first introduce the design requirements of the RCAnalyzer for completing the analytical tasks, and then we introduce the design of the RCAnalyzer in detail.

### 4.1 Design requirements

We identify the following design requirements that the RCAnalyzer should fulfill based on the analytical tasks.

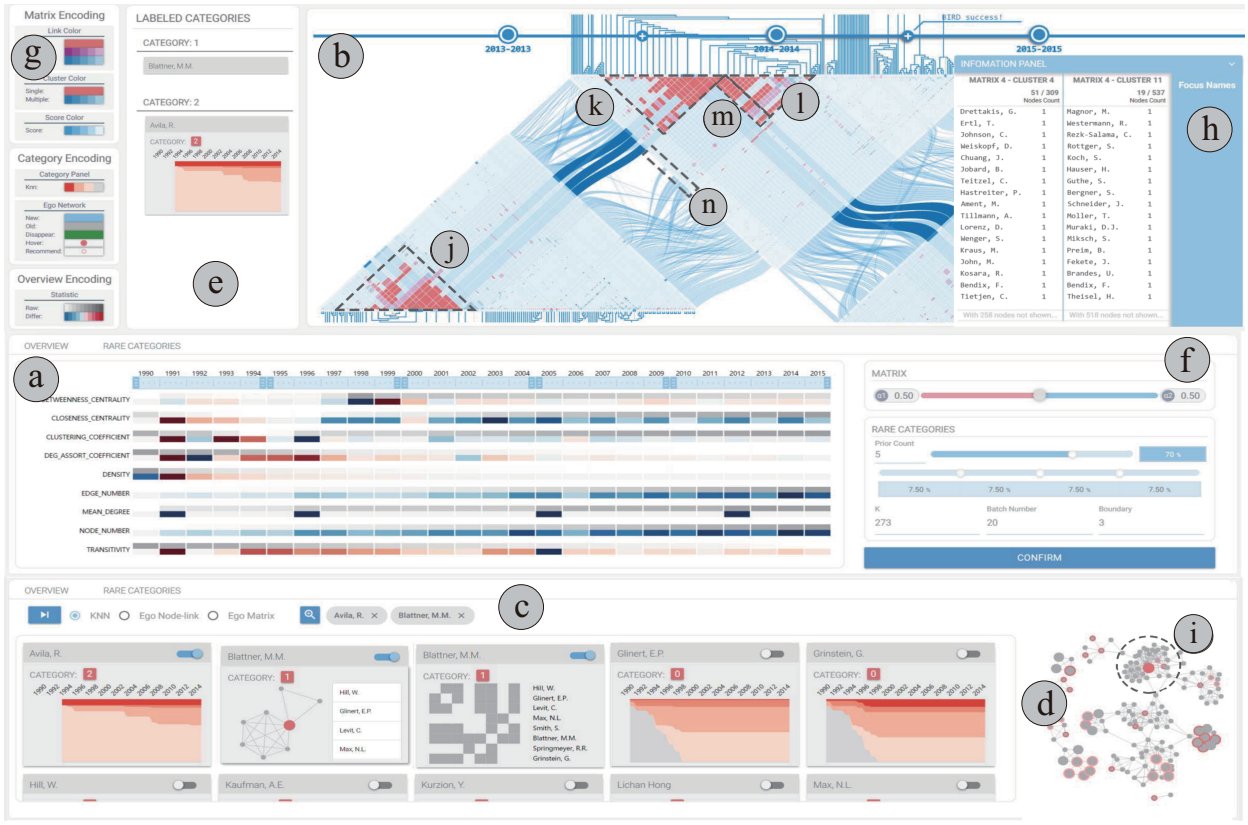


**Fig. 3 User interface of RCAnalyzer. (a) the timeline view; (b) the matrices view; (c)the instance view; (d) the sub-network view; (e) the label result view; (f) the parameter panel; (g) the encoding panel; and (h) the information panel. BIRD detects W. D., X. W., and H. L. between 2014 and 2015. (i) the compact neighborhood structures formed by them and their surrounding area in the sub-network view; (j) the small community constituted by them and their surrounding area in 2013; (k) the same area as (j) in 2014; (l) a dense structure appeared beside (k); (m) two nodes in (k) have a lot of connections to nodes in (l); (n) the Sankey diagram shows 8 nodes in (l) are nodes in 2014. (l) indicates the existence of a paper with lots of coauthors, which might be a result of multilateral cooperation. The abnormal change of the surrounding areas of W. D., X. W., and H. L. make them a rare category.**

For setting parameters to initialize BIRD (T1), we identify the following design requirements:

R1 **Provide an overview of dynamic networks.** Users need to first explore the entire dynamic networks and understand the overall change of dynamic networks. With an overview, users can decide on which time periods they would focus on.

To identify examples belonging to rare categories among all detected examples (T2), we identify the following design requirements:

R2 **Capture the changing structures of rare categories in the context of dynamic networks.** It is necessary to show the evolution of candidates in the background of the entire network. This helps users to identify the differences between the instance and the majority class.

R3 **Reveal the features of detected examples.** It is essential to show the features of the surrounding area of candidates to identify rare categories. The features include the ego network of the instance and the similar nodes detected by BIRD.

R4 **Reserve the context of labeled rare categories.** The system should remind users what kind of rare categories are detected and support the comparison between new candidates and labeled categories.

To label the examples based on analysis results (T3), we identify the following design requirements:

R5 **Enable users to set and reset the labels of candidates.** The system should enable users to label rare categories and change labels of rare categories when they make mistakes.

### 4.2 System pipeline

From the design requirements, we designed the user interface of the RCAnalyzer (see Fig. 3). It consists of a) a timeline view, which shows a high-level overview of dynamic networks (R1); b) the matrices view, which shows the aggregated adjacency matrix of dynamic networks at each time segment initially (R1) and shows the details of the neighborhood of multiple vertices after node selection (R2); c) the example view, which shows the feature of candidates (R3) and the query history of BIRD (R4); d) the label result view, which shows the historical label result and enables users to reset labels of labeled categories (R5 and R4).

Based on the analytical tasks, we design the architecture of the RCAnalyzer, as shown in Fig. 4.

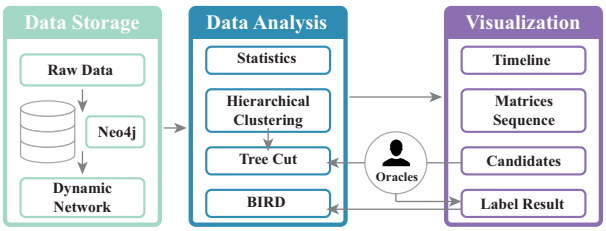


**Fig. 4 System pipeline.**

The RCAnalyzer consists of three major modules, a data storage module, a data analysis module, and a frontend visualization module. We use neo4j to store the dynamic networks in the data storage module. The data analysis module contains four components, consisting of BIRD, a hierarchical clustering algorithm, a tree cut algorithm and a bundle of statistics metrics. BIRD detects candidates of the rare categories iteratively. The hierarchical clustering algorithm extracts a tree structure from the network topology and the tree cut algorithm groups nodes to clusters based on the tree structure and the network topology. The statistics metrics measure the macro condition of the dynamic network.

The visualization module contains four major views: a timeline view, which shows the variation of network statistics and assist users select, merge, and filter time steps; a matrices view, which visualizes the network dynamics based on the tree cut result; an instance view, which displays the features of the rare category candidates detected by BIRD; a label result view, which reminds users what rare categories have been discovered.

### 4.3 The timeline view

The timeline view provides a highly abstracted overview of the dynamic network (R1). Metrics including betweenness centrality, closeness centrality, clustering coefficient, degree assort coefficient, density, edge number, node number, average degree, and transitivity are calculated to show the state of dynamic networks at each time stamp. The timeline view contains two parts, an interactive time axis, and a pixel map. The pixel map visualizes metrics, which helps users to find interesting snapshots of dynamic networks. The interactive time axis (see Fig. 3 (A)) enables users to select different snapshots (R1). After the time periods are submitted, the selected snapshots are extracted and merged accordingly. The data of merged snapshots are then visualized in the Matrices View to show the network data in detail.

**Design Considerations** We considered using three different visual designs in the timeline view to visualize the metrics: a line chart, a pixel map, and a glyph design. A line chart is intuitive to show time-varying data, while it lacks space efficiency. Using glyphs to show the metrics at each time stamp individually is space efficient while lacking intuitiveness. Thus, we choose to use a pixel map to show the metrics because a pixel map is more space efficient than line charts and more intuitive than a series of glyphs.

### 4.4 The matrices view

After time periods are selected in the timeline view, the data analysis module first aggregates snapshots of the dynamic network according to the selected time periods. The matrices view is designed for showing the dynamics of the network topology and the dynamics of selected rare category candidates. A hierarchical clustering algorithm (Newman and Girvan, 2004), which builds a dendrogram based on network topology, is applied on each aggregated snapshot to reduce the number of entries in each matrix because a large matrix can hardly be visualized in a limited space with satisfactory detail. Same clusters at different time stamps are linked together to show the dynamics of the network. However, users cannot really explore and compare the neighborhood of rare category candidates in aggregated matrices because of the lack of detail. Therefore, a multi-

focus tree cut algorithm is applied to each dendrogram to provide fine-grained detail of user-selected candidates and coarse-grained detail of other nodes. In this way, users are able to observe and compare the evolution pattern of rare category candidates (R2)

#### 4.4.1 Multi-focus tree cutting

When users are interested in one or more rare category candidates, the dynamics of neighborhoods of these candidates are shown in the matrices view to support users to explore, compare, and identify rare categories among these candidates. We design a multi-focus tree cut algorithm to enable the matrices view to provide fine-grained details around selected nodes and coarse-grained details around unrelated nodes, which supports users in identifying rare categories among candidates (T2) by comparing the features of candidates, labeled rare categories and non-rare categories. Different from existing multi-focus+context approaches (Gansner et al., 2005; Feng et al., 2012; Sundararajan et al., 2013), which work on the layout result of networks, our method directly works on the network topology and thus does not depend on the layout of networks.

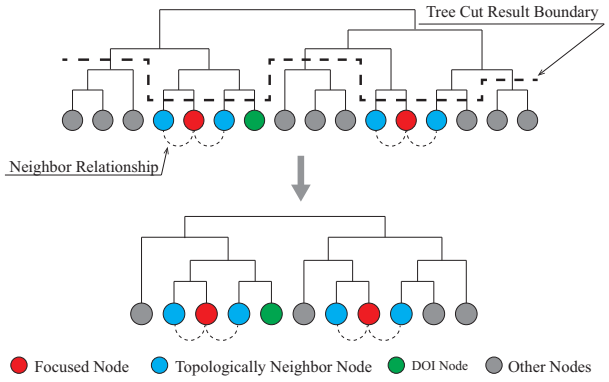


**Fig. 5 First stage of the tree cut algorithm: keep the details of all focused nodes.**

Suppose we are given a dynamic network, which consists of a series of snapshots, $\mathbb{G} = \{G^1, G^2, ..., G^t\}$. The multi-focus tree cutting algorithm works on each snapshot. The algorithm consists of two stages. In the first stage, details around all focused nodes are cut out from the tree; in the second stage, a merge operation is applied to prevent the result containing too many non-relevant single-node clusters.

**First stage: multi-focus tree cutting.** The procedure of the first stage is shown in Fig. 5. For a specific snapshot $G^i = (V, E)$, hierarchical clustering is applied first to obtain a tree structure based on modularity (Newman and Girvan, 2004). In order to cut the tree with multiple-focused nodes, we modified the original modularity. The set of focused nodes can be written as $F = \{n|focused\ nodes\}$. The cut of the tree structure is an optimization of an energy function based on the tree structure and the network topology. Suppose the cutting result is $C = \{N_1, N_2, ..., N_N\}$, where $N_i$ is a group of nodes in the tree. Then

$$C = \arg\min \Sigma_{i=1,2,...,N}(E(N_i)) \tag{2}$$

where

$$\begin{cases} E(N) & = & \Sigma_{e\in N}\frac{D(e,N)}{\|N\|} - \Sigma_{e\in N}(\frac{S(e,N)}{\|N\|})^2 \\ D(e,N) & = & \begin{cases} Weight(e), & if \forall v \in e, v \in N \\ 0, & else. \end{cases} \\ S(e,N) & = & \begin{cases} Weight(e), & if \exists v \in e, v \in N \\ 0, & else. \end{cases} \end{cases} \tag{3}$$

We defined the weight of an edge as the minimum of the weights of the node it links: suppose $e = (v1, v2)$, then $Weight(e) = min(Weight(v1), Weight(v2))$. The weight of a node is defined based on the distance between the node and the focus nodes both in the tree structure and the network topology:

$$\begin{cases} Weight(v) & = & \alpha_1 W_{DOI}(v) + \alpha_2 W_{Topology}(v) \\ W_{DOI}(v) & = & min_{n\in F}(D_{DOI}(n, v)) \\ W_{Topology}(v) & = & min_{n\in F}(D_{Topology}(n, v)) \end{cases} \tag{4}$$

, where $D_{DOI}(n, v)$ is the degree of interest distance between n and focused node v in the tree structure, $D_{Topology}(n, v)$ is the shortest distance between n and focused node v in the network topology, and $\alpha_1$ and $\alpha_2$ are weights of the two distances.

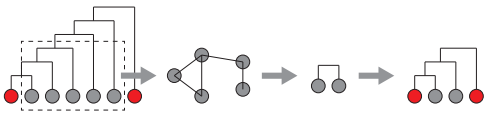

**Fig. 6 Second stage of tree cut algorithm: re-group the unrelated nodes according to the network structure.**

**Second stage: re-clustering of non-relevant nodes in the partial structure.** When

the structure of a hierarchical clustering tree is partial and the focused nodes are deep in the tree, a large number of non-relevant nodes might be cut out from the tree, which makes the cut result tall. To avoid this problem, we apply a re-cluster procedure to the non-relevant nodes. The continuous non-relevant single node sequences are first detected and cut out from the tree. Then the tree cut algorithm is applied again to the sub-tree based on the network topology. Last, hierarchies are inserted back into the tree. The procedure of this stage is shown in Fig. 6.

#### 4.4.2 Visual designs in the matrices view

We use a combination of matrix, Sankey diagram and dendrogram as the basic representation of dynamic networks(see Fig. 1). Sankey diagrams are added between each pair of adjacent matrices to show the evolution of these groups. The hierarchy of clusters represents the relationships among clusters and the structure of the network. In the RCAnalyzer, all networks are treated as undirected networks, and thus the adjacent matrices are symmetric. We use dendrograms to replace the upper (lower) triangular matrices and show the hierarchy of clustering result for space efficiency. The sequence of upper and lower triangular matrices are laid in a zigzag shape (see Fig. 1).

Due to the tree cut algorithm, there are different granularity details. This leads to different numbers of nodes in different clusters. The opacity and color of triangles on the diagonal of matrices encode the number of nodes, as shown in Fig. 7. We use blue and red (shown in Fig. 7) to distinguish a group of nodes and a single node. The gradient of blue in Fig. 7 is used to encode the number of nodes in groups. Rectangles inside matrices represent three categories of connections: a single node to a single node, a single node to a group of nodes, and a group of nodes to a group of nodes. For consistency, we use blue to encode group-to-group relations, orange to encode one-to-one relations, and purple to encode one-to-group relations. The gradient of colors (Fig. 7) represents the actual number of connections between the corresponding nodes.

Due to the importance of node anomalies in this work, we decide to use the size of triangles on the diagonal of matrices to encode the anomalous scores output by the BIRD algorithm (R3). If a large number of clusters is generated by the tree cut algorithm, sizes of single node clusters will be small under the limited size of matrices, which impedes the analysis of the nodes in which users are interested. We use three methods simultaneously to solve this problem. First, freely zooming and dragging are supported in this view. When the matrices are enlarged, the sequence of matrices cannot be fully displayed because of the limitation of space. Thus, we implement a special scale interaction with the scale functions shown in Fig. 8 to enable local scaling without changing the size of matrices.

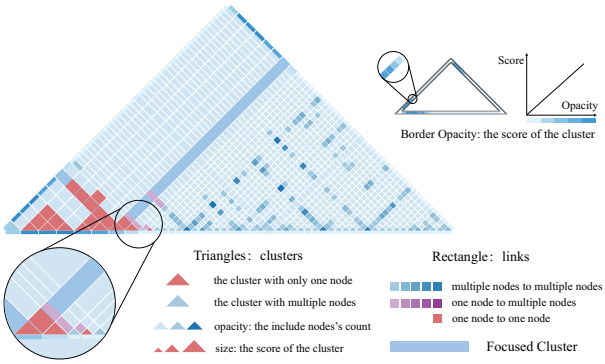


**Fig. 7 Visual encodings inside a matrix. Triangles represent a single node (red) or a group of nodes (gradient blue showing size). A red rectangle represents the connection between two single nodes; a purple rectangle represents the connections between a single node and a group of nodes; a blue rectangle represents the connections between two groups of nodes. Scores are encoded both by size of rectangles and triangles and the color on the matrix border.**

When the scale interaction is activated, the distortion of the size of the triangles and rectangles may mislead users, although we maintain the size ratio in the scaled local area. Thus, we encode the scores on the borders of the matrices by color, which brings two benefits: 1. users will clearly distinguish to which clusters the bands in Sankey diagrams belong when matrices are sparse; 2. users will observe the changes of scores over time stamps more easily.

**Design Considerations** Node-link Diagram and matrix representation are two common techniques to visualize networks. We choose the matrix as the basic representation of networks instead of the node-link diagram because the matrix representation can be better combined with a dendrogram.

Although same clusters or nodes can be linked together in a series of node-link diagrams to visualize a dynamic network, overlap of lines in this solution will be severe and significantly reduce the readability of the visualization.

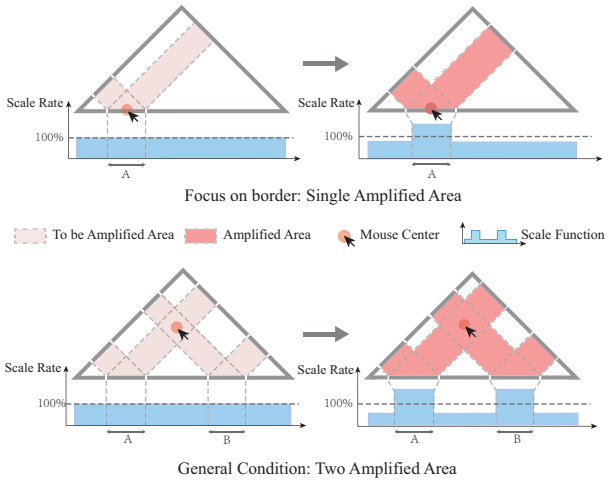


**Fig. 8 Scale functions when focus on the border of a matrix and focus inside a matrix.**

### 4.5 The rare category candidates view

The rare category candidates view is designed to reveal the features of candidates (R3). It contains two components: small multiples of candidate feature panels, which visualize the neighborhood information of candidates, and a sub-network view, which shows the sub-network formed by all detected candidates and their first-hop friends.

**Representation of Ego Network** of candidates consists of two visualization forms: a node-link diagram and a matrix. The coexistence of node-link diagrams and matrices is not considered as redundant because we think the two visualization forms have different emphases: the former emphasizes vertices while the latter emphasizes links. Because BIRD detects rare categories between two time steps, changes of the candidates' ego networks at the two time steps are shown in Fig. 3. The state of vertices and links are encoded by colors: blue indicates appearance, green indicates disappearance, and grey indicates fixedness.

**Sub-network of Candidates** shows the query process of BIRD by visualizing all the candidates together with their first-hop-neighbors (R3) and helps users to compare the candidates in the local area of the network. The color encoding is similar to the encoding in ego networks. Except for the color of links and nodes, we use red border of nodes to demonstrate the candidates detected in the current iteration and light red border of nodes to demonstrate the candidates detected in previous iterations. When an instance is hovered, both itself and its kNN will be enlarged, as shown in Fig. 9.

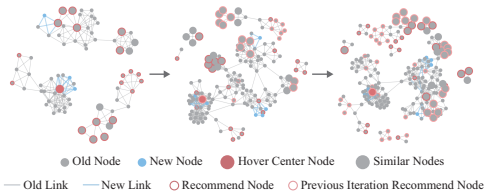


**Fig. 9 The sub-network view shows the query process of the BIRD algorithm by a node-link diagram formed by all the candidates ever queried by BIRD.**

### 4.6 Other panels

**The Label Result View** The label result view shows detected rare categories by recording label results of rare category candidates in a list of candidate feature panels (R4), as shown in Fig. 3. Users can review the detected rare categories at any time during the analysis procedure.

**The encoding panel** shows the color encodings used in the system (see Fig. 3 (G)). **The information panel** shows the detail information of selected blocks in the matrices view, as shown in Fig. 3 (H). When hovering on triangles on the diagonal of a matrix, node count and node list are shown in the panel. When hovering on rectangles inside a matrix, the information panel is divided into two parts, each of which shows the node count and the nodes that have connections to the other cluster. The link count between two clusters is also shown(see Fig. 3 (H)).

### 4.7 User interaction

The system implements a series of user interactions to support users to analyze the rare categories.

**Detail on demand** The instance view and the matrices view show the information of rare candidates at different levels of detail. Once nodes are selected in the instance view, the tree cut algorithm will be applied and the detail information of the selected candidates and their related nodes will be shown in the matrices sequence view with the context of the entire dynamic network.

**Highlighting & Pinning** All views in the RCAnalyzer are linked. Whenever and wherever a node is hovered over by users, other views will highlight the node and its related nodes. Users can pin the block by clicking on it and then explore the details in the information panel.

**Dragging & Zooming** The matrices view supports users in freely dragging and zooming the matrices sequence.

**Rare Category Labeling** Users can label each candidate with a specific number, which helps BIRD distinguish different rare categories in the feature panel.

## 5 System evaluation

In this section, we conducted one use scenario and a controlled user study to demonstrate the effectiveness of the RCAnalyzer. The use scenario is based on a dynamic network extracted from the collaboration among authors of visualization publications (Isenberg et al., 2017).

We developed a prototype system to do all the experiments. The RCAnalyzer is a web application which supports multiple users in analyzing the rare categories in dynamic networks. The front-end visualization is implemented by AngularJS, D3, and CSS. The back-end server is implemented by Python with Flask, Neo4J, numpy, igraph, and networkx. Use scenarios and the user study run on a PC with Intel(R) Core(TM) i7-4770 CPU, 20 GB RAM, and Windows10.

### 5.1 Use scenario: collaboration network in visualization publications

**Dataset** We extract all co-authorship in IEEE VIS dataset (Isenberg et al., 2017) from 1990 to 2015. An incremental collaboration network is constructed based on co-authorship, in which a link at timestamp $t$ indicates two authors have coauthored at $t$ or before $t$. We filtered the authors by taking the largest connected component in 2015 and there are 3640 authors left in the network. The number of links varies from 43(1990) to 11848(2015).

The timeline view and the matrices view show the basic information of the network (see Fig. 3 (a) and (b)). Note that the time axis is initially divided into 5 segments to show the condition of the dynamic network in periods of time. The heatmap and the matrices show that before 2000, both the number and the increment of nodes and links are small; after 2000, the network grows faster, and after 2004, the network grows significantly.

After initializing the BIRD with the data in 2014 and 2015, W. D., X. W., and H., L. are selected to be the focused nodes in the instance view, as shown in Fig. 3. They and their neighbors form a compact area in the sub-network view (Fig. 3 (i)). Their surrounding areas from 2013 to 2015 are shown in the matrices view. Focused nodes are highlighted by the blue lines. Area (j) in Fig. 3 is their surrounding area in 2013. The large link density in this area indicates that nodes in this area have close collaboration relationships. Thus, these nodes can be regarded as a small collaboration group. The Sankey diagram between 2013 and 2014 shows area (k) is almost the same as area (j). A dense structure in area (l) appeared beside area (k). Meanwhile, area (m) shows that two nodes, including X. W., in area (k) connect to most nodes in area (l). The blank of the Sankey diagram (labeled by (n)) on the left of the matrix in 2014 indicates that 8 nodes in area (l) are new nodes. The clique structure in area (l) indicates these nodes collaborated in the same paper. Large numbers of authors of the paper indicates that the paper might be the result of multilateral cooperation. The appearance of this uncommon cooperation causes W. D., X. W., H. L. to be identified as a rare category.

Between 2012 and 2013, D. J., S. A., and I. P. constitute a large and dense sub-network (Fig. 2). However, there is a small gap between the first three authors (Fig. 2 (A)) and the last author (Fig. 2 (B)). Thus, whether they belong to the same category cannot be decided. The matrices view shows the dynamic changes in surrounding areas around them . In 2011, I., P. is in the area (A), and D., J. and S., A. are in area (B). It is clear that these two areas have no connections. In 2012, area (C) shows that the two areas in 2011 merged into one because of the new connections in area (D). However, a large number of new connections appeared in area E in 2013. From the Sankey diagram between 2013 and 2014, we know that authors newly connected to D., J. and S., A. in 2013 also appeared in the area G in 2014. From the matrix of 2014, we can see that area G and area H are separated from each other. Thus, the merging and splitting behaviors of the surrounding areas of D. J., S. A., and I. P. along time are the

reasons why D. J., S. A., and I. P. are identified as a rare category.

### 5.2 User Study

We conducted a user study to verify the usability of the RCAnalyzer. We introduce the user study following the order of assumptions, datasets, participants, procedure, and result.

**Assumptions.** As there is no existing work supporting similar tasks to the RCAnalyzer, we do not use a baseline system in this user study and only test if the RCAnalyzer could help users to explore, analyze, and identify rare categories in dynamic networks and collect users' qualitative feedback. We first make three assumptions about the usability of the RCAnalyzer.

1 RCAnalyzer helps users identify examples of rare categories among the query result of the BIRD algorithm in each iteration.
2 RCAnalyzer helps users distinguish examples of rare categories and examples of major categories.
3 RCAnalyzer helps users distinguish examples of different rare categories.

For a dataset with ground truth, we can count the minimal number of iterations within which the BIRD algorithm can detect at least one example in each of the rare categories in the dataset. By comparing this minimal number and the actual number of iterations users use in the study, we can validate the assumption 1. If the number of iterations used by users is close to the minimal number, the RCAnalyzer efficiently supports users to identify rare categories. We validate assumption 2 and 3 by calculating the accuracy of the rare categories labels labeled by users in the user study.

**Synthetic Data.** Because of the high complexity of the real datasets used in the case studies, it is hard to control the test and quantify the actual efficiency of rare category detection with the RCAnalyzer. Thus, we use synthetic datasets in the user study. All the synthetic datasets have two time stamps. Each synthetic dataset is constructed by the following procedure: 1) generating a grid network with $N$ nodes at each time stamp; 2) adding edges among nodes in the network to form four different special structures: a clique, a bipartite graph, a star structure, and a circle, at the second time stamp. Special structures are treated as rare categories and other nodes are treated as the major category. We constructed four synthetic datasets with $N = 100$, 200, 500, and 1000. The dataset with $N = 100$ is used in the tutorial of the user study. The minimal numbers of iterations on datasets with $N = 200$, 500, and 1000 are 5, 5, and 11 respectively.

**Participants.** We recruited 12 participants for the evaluation, including 9 males and 3 females. All of them have background in visualization, and one of them has a background in anomaly detection.

**Tasks.** The participants are asked to complete the following tasks in the user study:

T1 Identify rare categories in the examples detected by BIRD in each iteration.
T2 Label examples identified as rare categories.

**Procedure.** The user study has three stages. In the first stage, we introduce the basic concept of this work and the tasks of the user study to participants with a 10-minute tutorial. In the second stage, we introduce the RCAnalyzer to participants and let them explore the system with the synthetic dataset with $N = 100$ for 15 minutes. Participants are allowed to ask any questions about the system and the tasks in the first and the second stages. In the third stage, participants are asked to analyze the synthetic datasets with $N$ =200, 500, and 1000, label rare categories they identified in the RCAnalyzer, and write down their labeling results on an answer sheet. In order to ensure that participants will not give answers arbitrarily, they are asked to describe the reason why a detected example is identified as a rare category.

**Result** The accuracy of labeling rare categories is shown in Table 1. The results show that the detection of the clique, bipartite graph, and star graph is accurate (86.11%, 86.11%, and 91.67%) while the accuracy of detection of the circle is not very good (77.87%). Detection of a circle structure is really hard because the surrounding area of a node on the circle is unobtrusive in the matrices view and nodes on the circle are queried by BIRD discontinuously, forming several segments of line instead of a circle in the sub-network view. To identify the circle structure, participants need to select a series of instances on the circle, but some of the participants missed too many instances on the circle, and thus were not able to label the circle structure correctly. The distribution of participants' query number is shown in Fig. 10. The result shows that participants can finish the labeling process in 4-5 iterations in datasets with 200 and 500 nodes. For the dataset with 1000 nodes,

**Table 1 Accuracy of labeling.**

| Size | Clique | Bipartite | Star | Circle |
|---|---|---|---|---|
| 200 | 91.67% | 83.33% | 83.33% | 58.33% |
| 500 | 83.33% | 83.33% | 100.00% | 91.67% |
| 1000 | 83.33% | 91.67% | 91.67% | 83.33% |
| Avg. | 86.11% | 86.11% | 91.67% | 77.78% |

many of the participants can finish the labeling processing in 11 to 15 iterations, while two outliers finished the labeling process in 2 and 6 iterations. This was because they labelled normal nodes as rare categories. Some of the participants finished the labeling result after over 20 iterations. This is because they did not label the rare categories promptly. Overall, the accuracy and the average query numbers show that most of the participants are able to identify rare categories promptly and correctly.

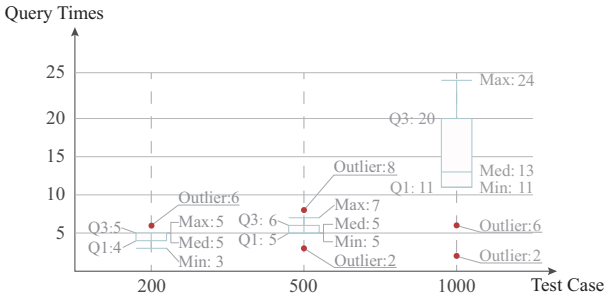


**Fig. 10 The query numbers of participants when they labeled all rare categories in the data.**

**Qualitative feedback** In order to assess the learnability, usability and other perception aspects of the RCAnalyzer, users were asked to give some qualitative feedback after the formal user study. The most frequent complaint was that the encodings in our matrices view were too complex. We used both the size and the color of each cell to encode different information. Users had to recognize all the encodings at the beginning of the user study. It would lead to confusion because they would forget the encodings. Some users said that the parameters were hard to comprehend. They said that it was hard to learn what will happen if the parameters were adjusted. It took a long time for them to learn how the system worked. Learnability and usability were both important problems which were hard to cover. One of the solutions for improvement is to reduce the complexity of our visual design. However, it takes much more time to know which visual design is less efficient and can be abandoned. In the future, we will redesign our visual design based on more user behaviors. For example, the color encoding on the border can be removed if users do not care about the border color encoding.

## 6 Discussion

**Generalizability** In our paper, we used a collaboration network to evaluate our system. However, the RCAnalyzer supports rare category analysis in other networks.Although we only support the BIRD algorithm in our system, the RCAnalyzer can work based on other RCD algorithms as long as they are based on the topology of dynamic networks. The matrices view with the tree cut algorithm can be applied in other applications for analyzing dynamic networks. For example, tracking the time-varying pattern of multiple nodes and comparing the change of ego-networks of multiple nodes. We believe that the combination of matrix sequence and multi-focus tree cut algorithm is a useful method as it enables simultaneous comparison of multiple nodes.

**Scalability** In use scenarios, we tested the effectiveness of the RCAnalyzer on a network with 8319 nodes, 210625 edges, and 6 time steps, which indicates that the RCAnalyzer has good scalability on large datasets. As for larger datasets, the major bottleneck would be the running time of initialization of the BIRD algorithm and tree cut algorithm due to the limitation of execution efficiency of Python. In the future, we plan to use pre-computation and server-side cache to support the analysis of larger datasets. As for the scalability of our visual design, it is related to the granularity of our tree cut algorithm, and the scale of the input dynamic network. From our experience, it is hard to show more than 6 time-steps with around 50 rows in each matrix at the same time in the matrices view (with 1,360 $\times$ 635 pixels). Interactions such as dragging and zooming to improve the readability of matrices have been discussed in 4.4.2. For dynamic graphs with more time-steps, the tree cut algorithm should be more coarse-grained to show all time-steps in the meantime. However, the coarse-grained tree cut algorithm reduces the information of the dynamic networks.

**Limitations** Although the RCAnalyzer is able to help users to analyze and label rare categories in dynamic networks, it still has several limitations. First, more interactions should be supported, such

as querying and filtering. Interactions in the the RCAnalyzer are enough to support the detection of rare categories, but more complete interactions can significantly improve user experience. Second, the process of interactions and visual encodings in the RCAnalyzer are a little complicated. During the user study, it takes 15-25 minutes to train subjects to let them fully understand how to use the system. Third, the RCAnalyzer only supports screens with $1920 \times 1080$ resolution. More adaptive layout should be supported to enable users to label rare categories at different resolutions.

**Future Work** First, we plan to add context information of nodes in the RCAnalyzer. The RCAnalyzer is based on the topology of dynamic networks currently because the BIRD algorithm detects rare categories by checking the changes of topological structure around nodes. However, nodes with the same topology may have completely different context information. We believe context information will help users distinguish different rare categories. Second, we plan to add data filtering to the RCAnalyzer. Sometimes, users might be interested in only a special area in the network. A data filtering module can help them analyze the desired areas of data.

## 7 Conclusion

In this paper, we present the RCAnalyzer, a novel visual analytics system which helps oracles to analyze the result of RCD methods and label the rare categories in dynamic networks. It consists of five linked views: a timeline view, a matrices view, an instance view, a sub-network view, and a label result view, and it shows the information of rare categories in different levels of detail. In addition, we present a multi-focus tree cut algorithm and a tree-structure constrained layout optimization algorithm to support the comparison of instances in the context of their surrounding structures. We use one use scenario, and one user study to demonstrate the usability and effectiveness in analyzing rare categories in dynamic networks.

### References


Archambault D, Purchase H, Pinaud B, 2011. Animation, small multiples, and the effect of mental map preservation in dynamic graphs. *IEEE Transactions on Visualization and Computer Graphics*, 17(4):539-552.

Bach B, Pietriga E, Fekete JD, 2013. GraphDiaries: Animated transitions and temporal navigation for dynamic networks. *IEEE transactions on visualization and computer graphics*, 20(5):740-754.

Bach B, Pietriga E, Fekete JD, 2014. Visualizing dynamic networks with matrix cubes. Proceedings of the SIGCHI conference on Human Factors in Computing Systems, p.877-886.

Bach B, Henry-Riche N, Dwyer T, et al., 2015. Small MultiPiles: Piling time to explore temporal patterns in dynamic networks. Computer Graphics Forum, 34(3):31-40.

Beck F, Burch M, Diehl S, et al., 2014. The state of the art in visualizing dynamic graphs. EuroVis (STARs).

Bhuyan MH, Bhattacharyya DK, Kalita JK, 2013. Network anomaly detection: methods, systems and tools. *Ieee communications surveys & tutorials*, 16(1):303-336.

Blanch R, Dautriche R, Bisson G, 2015. Dendrogramix: A hybrid tree-matrix visualization technique to support interactive exploration of dendrograms. 2015 IEEE Pacific Visualization Symposium (PacificVis), p.31-38.

Brandes U, Nick B, 2011. Asymmetric relations in longitudinal social networks. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2283-2290.

Burch M, Schmidt B, Weiskopf D, 2013. A matrix-based visualization for exploring dynamic compound digraphs. 2013 17th International Conference on Information Visualisation, p.66-73.

Cao N, Gotz D, Sun J, et al., 2011. DICON: Interactive visual analysis of multidimensional clusters. *IEEE Transactions on Visualization & Computer Graphics*, 17(12):2581-2590.

Cao N, Shi C, Lin S, et al., 2016. TargetVue: Visual analysis of anomalous user behaviors in online communication systems. *IEEE transactions on visualization & computer graphics*, 22(1):280-289.

Chandola V, Banerjee A, Kumar V, 2009. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15.

Corchado E, Herrero Á, 2011. Neural visualization of network traffic data for intrusion detection. *Applied Soft Computing*, 11(2):2042-2056.

van den Elzen S, Holten D, Blaas J, et al., 2015. Reducing snapshots to points: A visual analytics approach to dynamic network exploration. *IEEE transactions on visualization and computer graphics*, 22(1):1-10.

Fan X, Li C, Yuan X, et al. An interactive visual analytics approach for network anomaly detection through smart labeling. *Journal of Visualization*, :1-17.

Feng KC, Wang C, Shen HW, et al., 2012. Coherent time-varying graph drawing with multifocus+ context interaction. *IEEE Transactions on Visualization and Computer Graphics*, 18(8):1330-1342.

Gansner ER, Koren Y, North SC, 2005. Topological fisheye views for visualizing large graphs. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):457-468.

Haberkorn T, Koglbauer I, Braunstingl R, 2014. Traffic displays for visual flight indicating track and priority cues. *IEEE transactions on human-machine systems*, 44(6):755-766.

Havre S, Hetzler B, Nowell L, 2000. ThemeRiver: Visualizing theme changes over time. IEEE Symposium on Information Visualization 2000 INFOVIS 2000 Proceedings, p.115-123.

He J, Carbonell J, 2009. Prior-free rare category detection. Proceedings of the 2009 SIAM International Conference on Data Mining, p.155-163.

He J, Carbonell JG, 2008. Nearest-neighbor-based active learning for rare category detection. Advances in neural information processing systems, p.633-640.

He J, Liu Y, Lawrence R, 2008. Graph-based rare category detection. 2008 Eighth IEEE International Conference on Data Mining, p.833-838.

He J, Tong H, Carbonell J, 2010. Rare category characterization. 2010 IEEE international conference on data mining, p.226-235.

Heard NA, Weston DJ, Platanioti K, et al., 2010. Bayesian anomaly detection methods for social networks. *The Annals of Applied Statistics*, 4(2):645-662.

Henry N, Fekete JD, McGuffin MJ, 2007. NodeTrix: a hybrid visualization of social networks. *IEEE transactions on visualization and computer graphics*, 13(6):1302-1309.

Hlawatsch M, Burch M, Weiskopf D, 2014. Visual adjacency lists for dynamic graphs. *IEEE transactions on visualization and computer graphics*, 20(11):1590-1603.

Huang H, He Q, He J, et al., 2011. RADAR: Rare category detection via computation of boundary degree. Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, p.258-269.

Huang H, He Q, Chiew K, et al., 2013. CLOVER: a faster prior-free approach to rare-category detection. *Knowledge and information systems*, 35(3):713-736.

Inselberg A, 2009. Parallel Coordinates. Springer New York, p.2018-2024.

Isenberg P, Heimerl F, Koch S, et al., 2017. vispubdata.org: A Metadata Collection about IEEE Visualization (VIS) Publications. *IEEE Transactions on Visualization and Computer Graphics*, 23 (To appear).
https://hal.inria.fr/hal-01376597
https://doi.org/http://dx.doi.org/10.1109/TVCG.2016.2615308

Jolliffe, Ian, 1986. Principal component analysis. *Springer Berlin*, 87(100):41-64.

Jovanovic J, Bagheri E, Gasevic D, 2015. Comprehension and learning of social goals through visualization. *IEEE Transactions on Human-Machine Systems*, 45(4):478-489.

Lin H, Gao S, Gotz D, et al., 2017. RCLens: Interactive rare category exploration and identification. *IEEE transactions on visualization and computer graphics*, .

Liu Y, Dai S, Wang C, et al., 2017. GenealogyVis: A system for visual analysis of multidimensional genealogical data. *IEEE Transactions on Human-Machine Systems*, 47(6):873-885.

Newman ME, Girvan M, 2004. Finding and evaluating community structure in networks. *Physical Review E Statistical Nonlinear & Soft Matter Physics*, 69(2):026113.

Oelke D, Kokkinakis D, Keim DA, 2013. Fingerprint Matrices: Uncovering the dynamics of social networks in prose literature. Computer Graphics Forum, 32(3pt4):371-380.

Pelleg D, Moore AW, 2005. Active learning for anomaly and rare-category detection. Advances in neural information processing systems, p.1073-1080.

Ranshous S, Shen S, Koutra D, et al., 2015. Anomaly detection in dynamic networks: a survey. *Wiley Interdisciplinary Reviews: Computational Statistics*, 7(3):223-247.

Riehmann P, Hanfler M, Froehlich B, 2005. Interactive sankey diagrams. IEEE Symposium on Information Visualization, 2005 INFOVIS 2005, p.233-240.

Sun J, Faloutsos C, Faloutsos C, et al., 2007. Graphscope: parameter-free mining of large time-evolving graphs. Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, p.687-696.

Sundararajan PK, Mengshoel OJ, Selker T, 2013. Multi-focus and multi-window techniques for interactive network exploration. Visualization and Data Analysis 2013, 8654:86540O.

Teoh ST, Ma KL, Wu SF, et al., 2002. Case study: Interactive visualization for internet security. Proceedings of the conference on Visualization'02, p.505-508.

Thom D, Bosch H, Koch S, et al., 2012. Spatiotemporal anomaly detection through visual analysis of geolocated twitter messages. Pacific visualization symposium (PacificVis), 2012 IEEE, p.41-48.

Tsai CF, Hsu YF, Lin CY, et al., 2009. Intrusion detection by machine learning: A review. *Expert Systems with Applications*, 36(10):11994-12000.

Vatturi P, Wong WK, 2008. Category detection using hierarchical mean shift. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July, p.847-856.

Vehlow C, Beck F, Auwärter P, et al., 2015. Visualizing the evolution of communities in dynamic graphs. Computer Graphics Forum, 34(1):277-288.

Wang C, Xiao Z, Liu Y, et al., 2013. SentiView: Sentiment analysis and visualization for internet popular topics. *IEEE transactions on human-machine systems*, 43(6):620-630.

Xu P, Mei H, Liu R, et al., 2017. ViDX: Visual diagnostics of assembly line performance in smart factories. *IEEE Transactions on Visualization & Computer Graphics*, 23(1):291.

Yee KP, Fisher D, Dhamija R, et al., 2001. Animated exploration of dynamic graphs with radial layout. IEEE Symposium on Information Visualization, p.43-43.

Zhang T, Wang X, Li Z, et al., 2017. A survey of network anomaly visualization. *Science China Information Sciences*, 60(12):121101.

Zhao J, Cao N, Wen Z, et al., 2014. FluxFlow: Visual analysis of anomalous information spreading on social media. *IEEE Transactions on Visualization & Computer Graphics*, 20(12):1773-1782.

Zhao J, Liu Z, Dontcheva M, et al., 2015. MatrixWave: Visual comparison of event sequence data. Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, p.259-268.

Zhou D, He J, Candan KS, et al., 2015a. MUVIR: multi-view rare category detection. Twenty-Fourth International Joint Conference on Artificial Intelligence.

Zhou D, Wang K, Cao N, et al., 2015b. Rare category detection on time-evolving graphs. 2015 IEEE International Conference on Data Mining, p.1135-1140.

Zhou D, Karthikeyan A, Wang K, et al., 2017. Discovering rare categories from graph streams. *Data mining and knowledge discovery*, 31(2):400-423.