The VLDB Journal
https://doi.org/10.1007/s00778-021-00723-z

SPECIAL ISSUE PAPER

eRiskCom: an e-commerce risky community detection platform

Fanzhen Liu[1] · Zhao Li[2,3] · Baokun Wang[4] · Jia Wu[1] · Jian Yang[1] · Jiaming Huang[3] · Yiqing Zhang[4] · Weiqiang Wang[4] · Shan Xue[1] · Surya Nepal[5] · Quan Z. Sheng[1]

Received: 21 February 2021 / Revised: 5 October 2021 / Accepted: 3 December 2021

Abstract

In e-commerce scenarios, frauds events such as telecom fraud, insurance fraud, and fraudulent transactions, bring a huge amount of loss to merchants or users. Identification of fraudsters helps regulators take measures for targeted control. Given a set of fraudsters and suspicious users observed from victims' reports, how can we effectively distinguish risky users closely related to them from the others for further investigation by human experts? Fraudsters take camouflage actions to hide from being discovered; complex features on users are hard to deal with; patterns of fraudsters are sometimes difficult to explain by human knowledge; and real-world applications involve millions of users. All this makes the question hard to answer. To this end, we design eRiskCom, an e-commerce risky community detection platform to detect risky groups containing identified fraudsters and other closely related users. With the hypothesis that users who interact frequently with fraudsters are more likely to come from the same "risky community," we construct a connected graph expanded from the identified fraudsters and suspicious users. Next, graph partition is employed to get knowledge of assignment of identified users to potential risky communities, followed by pruning to discover the core members of each community. Finally, top-K users with a high risk score in the neighborhood of core members of each potential community form a final risky community. The extensive experiments are conducted to analyze the effect of our platform components on the alignment with requirements of practical scenarios, and experimental results further demonstrate that eRiskCom is effective and easy to deploy for real-world applications.

Keywords Community detection · E-commerce · Telecom fraud · Insurance fraud · Transaction fraud · Fraud detection · Subgraph pattern · Graph mining

1 Introduction

As e-commerce is being integrated into our lives today, we enjoy the convenience brought by it. However, various fraud events such as credit card fraud [33], money laundering [42], and insurance fraud [35], cause a huge amount of loss to merchants or users. Fraud detection [73] offers a solution to

✉ Zhao Li
zhao_li@zju.edu.cn

✉ Jia Wu
jia.wu@mq.edu.au

Fanzhen Liu
fanzhen.liu@hdr.mq.edu.au

Baokun Wang
yike.wbk@antgroup.com

Jian Yang
jian.yang@mq.edu.au

Jiaming Huang
jimmy.hjm@alibaba-inc.com

Yiqing Zhang
zhongjun.zyq@antgroup.com

Weiqiang Wang
weiqiang.wwq@antgroup.com

Shan Xue
emma.xue@mq.edu.au

Surya Nepal
surya.nepal@data61.csiro.au

Quan Z. Sheng
michael.sheng@mq.edu.au

[1] School of Computing, Macquarie University, Sydney, Australia

[2] Zhejiang University, Hangzhou, China

[3] Alibaba Group, Hangzhou, China

[4] Ant Financial Services Group, Shanghai, China

[5] CSIRO's Data61, Sydney, Australia

Published online: 17 January 2022

this, which is expected to uncover fraudsters and fraud events and expose them to decision-makers who take measures for targeted control.

In most real-world applications, fraud events are always planned by groups of fraudsters who share close ties and similar information within groups. Recent study [47] benefits from the hypothesis that fraudsters share dense connections and form communities. In the following, three real-life scenarios studied in this work are given for further explanation.

In the online payment scenario, telecom fraud accounts for the majority of fraud events. Fraudsters pretend to be customer service staff or officers from public security organizations to commit telecom fraud. Statically, around 90% of victims did not report through payment platforms like Alipay,[1] which encourages the growth of such kind of fraud. Therefore, there is an urgent demand for the efficient detection of fraudsters within financial payment platforms. In the financial insurance industry, insurance fraud happens all the time. Fraudsters commit fraud by applying for insurance claims to insurance platforms by dishonest means. The identification of fraudsters can help find the potential fraudsters and get knowledge of fraud patterns, which guides the future insurance claim services and cuts the loss. In the above two scenarios, fraud is usually committed by fraud groups whose members perform their duties and cooperate with each other. It is easily observed that within a fraud group, fraudsters are more likely to have close social connections, e.g., family membership, friendship, and colleague relationships. Besides, money transfer sometimes with a large amount occurs among their accounts.

On online shopping platforms like Taobao,[2] to promote its target products, a seller could hire many fraudsters, namely *brushers*, to purchase these products online and give them a high (usually, 5-star) rating [74]. As a result, the promoted products would rank at the forefront of similar products in terms of sales and ratings. Analogous to what we observe in the telecom fraud and insurance fraud, a brusher who comes from a fraud group probably has social connections with other members and share similar features, such as purchasing preferences and education background. Therefore, fraud detection helps the regulator find those brushers and impose penalties on them and the sellers who hire them.

In most cases, only a small number of fraudsters and suspicious users can be identified or observed according to reports from victims of fraud. As shown in Fig. 1, exploiting the limited information like interactions between them, our task is to detect the communities containing identified seed fraudsters and other users closely related to the fraudsters. For this problem, the following challenges remain to address:

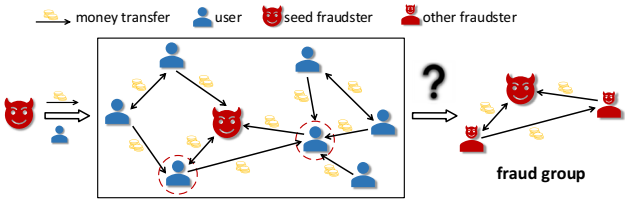

Fig. 1 A toy example of fraud group detection with the seed fraudster known in e-commerce scenarios

- **Hidden Fraudsters.** As some fraudsters are aware of some rules followed by fraud detection approaches, they may take some camouflage actions to hide from being discovered. For example, they intentionally avoid committing fraud within a very short time and try to establish a connection with normal users so that they cannot be easily found.
- **Complex Information.** Users are characterized by complex information such as social connections and semantic features. How to utilize complex information from different sources in a proper way is still being explored.
- **Interpretability.** Most existing fraud detection approaches do not provide interpretability of the results. A potential fraudster could be detected, but this cannot be readily explained by specific rules and human knowledge.
- **Scalability.** In real-world applications, we always need to deal with datasets larger and more complicated than those in most related studies, so most existing fraud detection approaches are difficult to meet the demand of effectiveness and efficiency.

Therefore, in the face of the above challenges, developing a platform for risky community detection is a non-trivial matter. Different from GraphRAD [47], a graph-based risky account detection system recently proposed by Amazon, which has a high time complexity due to the high-dimensional node features to deal with in the clustering process, here we propose eRiskCom, an e-commerce risky community detection platform to meet our work-flow requirements. This framework consists of four major modules as follows: (1) the Graph Construction module that builds a weighted node-link graph to describe the relationships or interactions between users by edge weight assignment, (2) the Query Decomposition module that divides users into several subsets by graph partition based on global topological structure and performs the pruning to further find the core members of potential risky communities, (3) the Community Search module that searches the neighborhood of core members and analyzes the roles of authority and hub for each user in the neighborhood, and (4) the Risk Scoring module that provides an assessment of risk level for each user

[1] www.alipay.com.

[2] www.taobao.com.

in communities and picks users with a high risk level to form the risky communities for further human expert check.

In summary, our work makes four major contributions as follows:

- We summarize the challenges facing the fraud detection problem in real-world applications and propose a corresponding solution.
- We put forward eRiskCom, an e-commerce risky community detection platform to discover groups containing identified fraudsters and other closely related users for further investigation. eRiskCom is composed of four major modules which utilize the label information of identified users and interactions between them to detect risky communities.
- We leverage the complex information on identified fraudsters and suspicious users through different edge weight assignment strategies, while other works only make use of identified fraudsters.
- We perform a series of experiments on three real-world datasets to analyze the power of eRiskCom's components in the alignment with requirements of practical applications and look into the patterns of detected risky communities, which demonstrates the effectiveness of eRiskCom.

The rest of this paper runs as follows. Section 2 clarifies the problem we focus on in this paper and outlines the architecture of the proposed platform, eRiskCom whose four major modules are introduced in Sect. 3. Section 4 reports the experimental settings and discusses the results, followed by the related work on anomalous subgraph detection, community detection, and community search in Sect. 5. Eventually, Sect. 6 concludes this paper and looks forward to the future work.

2 Proposed platform

2.1 Problem definition

A weighted graph is formulated as $G = (V, E, W)$ based on graph theory, where a node set V and an edge set E denote users and interactions between them, respectively, and W denotes the corresponding weights on E. A weight value describes the intensity or capacity of a connection. Given a seed set $\mathcal{S}$ composed of observed fraudsters F and suspicious users S, we aim to detect a set of risky communities $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$, where each community C_i is a separate subgraph containing users with a high risk score for human experts for further investigation.

2.2 The proposed platform: eRiskCom

This section briefly introduces four main modules of eRiskCom whose architecture and pipeline are shown in Fig. 2 and Algorithm 1, respectively.

Algorithm 1 eRiskCom

Input: A big graph $\mathcal{G}$ and a seed set $\mathcal{S} = F \cup S$
Output: A set of risky communities $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$
1: Generate a graph $G = (V, E, W)$ in the Graph Construction module (Sect. 3.1)
2: Obtain query subgraphs each with unique seed node(s) from $\mathcal{S}$ in the Query Decomposition module (Sect. 3.2)
3: Detect candidate communities expanded from query subgraphs in the Community Search module (Sect. 3.3)
4: Select nodes with a high risk score for each candidate community to form the final community set $\mathcal{C}$ in the Risk Scoring module (Sect. 3.4)

- **Graph Construction.** A big connected graph is first constructed on the presence of a certain relationship or interaction between users. Given a set of seed nodes representing observed fraudsters and suspicious individuals (if observed) in a specific scenario, the graph we focus on is then constructed based on 1- or 2-hop neighborhood of seed nodes. Besides the topological structure, the user behavior pattern described by selected semantic features is considered by weight setting on edges between users.
- **Query Decomposition.** The Query Decomposition module aims to divide the seed nodes into subsets for the subsequent search process. It takes in the constructed graph from the Graph Construction module and performs community detection to partition the whole graph into a set of query subgraphs each of which gathers some nodes from the seed set and unobserved fraudsters. Then, we perform pruning of these subgraphs to ensure that normal users are filtered out as many as possible and reduce the workload of human experts for further check.
- **Community Search.** Based on the results provided by the Query Decomposition module, we expand the search area by adding the 1- and 2-hop neighbors of members into the query subgraphs to obtain the candidate communities. Then we quantify the risk level of each user with respect to the roles of authority and hub.
- **Risk Scoring.** For each candidate community, we predefine a parameter K by observing the curve of a local metric for communities. Then top-K users with a high risk score are selected to form a risky community.

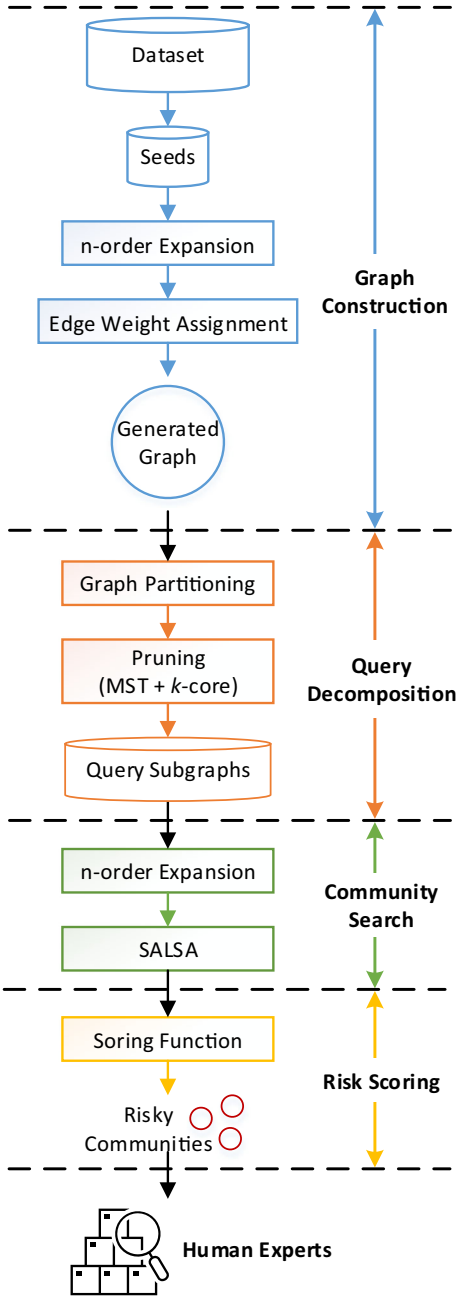

Fig. 2 The architecture of eRiskCom

3 Proposed modules

This section details the corresponding methods applied in Graph Construction, Query Decomposition, Community Search, and Risk Scoring modules. The detection process of a risky community is visualized in Fig. 3 to help understand the process and clarify various types of nodes in each module.

3.1 Graph construction

Besides the topological structure describing the relationships or interactions between users, semantic features provide a view to learning the user behavior. On the other hand, based on the detection results, identifying the fraudsters by human expert check is far from enough. Some unknown fraud patterns in risky communities could be further studied from semantic features, which would help our future work. To this end, following the steps of Algorithm 2, a graph G is derived from an original big graph $\mathcal{G}$ in the dataset to serve practical purposes.

Algorithm 2 Graph Construction

Input: A big graph $\mathcal{G}$ and a seed set $\mathcal{S} = F \cup S$
Output: A graph $G = (V, E, W)$
Initialize $V, E, W = \emptyset$
for each node $v_i \in \mathcal{S}$ **do**
  for each node v_j in $\mathcal{G}$ **do**
    if v_j is in the n-hop neighborhood of v_i **then**
      $V \leftarrow V \cup v_j$
    end if
  end for
end for
for each edge (i, j) in $\mathcal{G}$ **do**
  if (i, j) connects any pair of nodes $v_i, v_j \in V$ **then**
    $E \leftarrow E \cup (i, j)$
    Calculate weight w_{ij} of (i, j) based on the RFM model
    Calculate weight $w_{ij}^{'}$ of (i, j) by Eq. (1)
    $W \leftarrow W \cup w_{ij}^{'}$
  end if
end for
return $G = (V, E, W)$

Since our work focuses on e-commerce scenarios, the RFM (Recency, Frequency and Monetary) model [8] that has been widely used to analyze the user behavior pattern in business is employed to guide the edge weight setting. Based on the RFM values and human knowledge, we follow a rule that transforms the edge-based features to the edge weight. Briefly speaking, if there exist frequent transactions between two users, or the transaction amount is somehow large, a high weight is assigned to the edge between them; on the contrary, if transactions with a small amount between them occur occasionally, a low weight is assigned to the edge between them. Another idea comes from the phenomenon that adjacent nodes denoting identified fraudsters or suspicious users are more likely to be found in the same group. Risky community detection is closely related to density within communities, so we make the edge weights between adjacent nodes from seed sets double those between other connected nodes, as shown in

$$w_{ij}^{'} = \begin{cases} 2w_{ij}, & \text{if } (v_i, v_j) \in E \text{ and } v_i, v_j \in F \cup S \\ w_{ij}. & \text{otherwise} \end{cases} \quad (1)$$

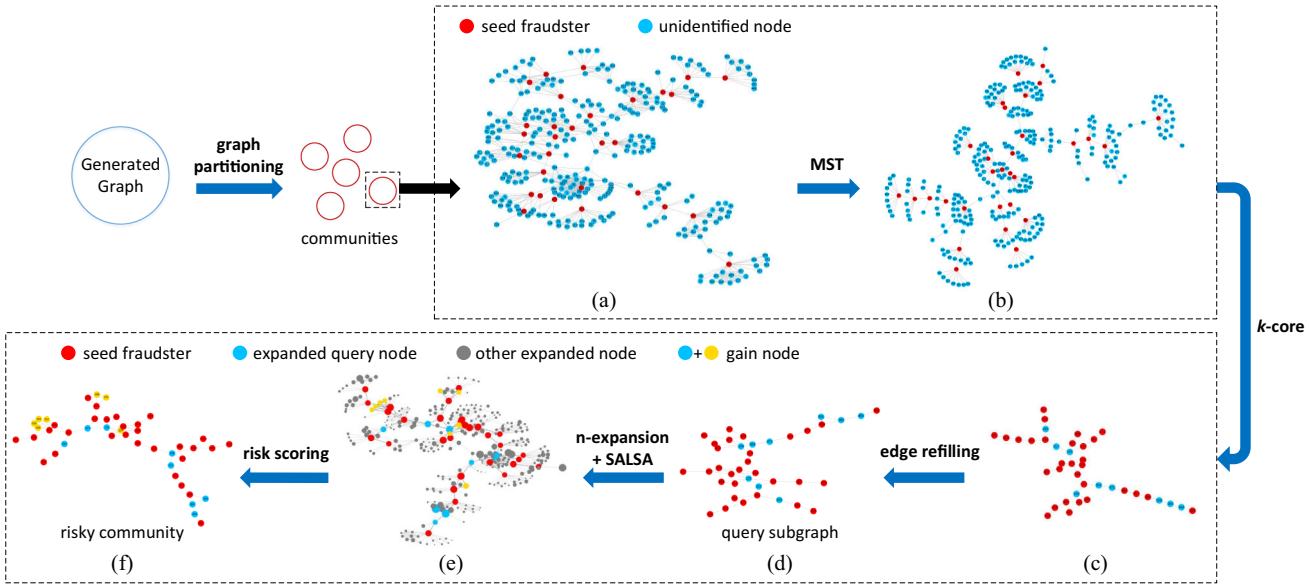

Fig. 3 The detection process of a risky community, given observed seed fraudsters. Graph partitioning is applied to divide the generated graphs into communities. For each community containing seed fraudsters and other unidentified nodes as shown in (**a**), we perform a pruning operation which applies a weighted minimum spanning tree (MST) algorithm and k-core to filter out the unidentified nodes less related to the seed fraudsters so that the remaining nodes are probably core members of the final risky community. Taking into account the appearance of potential suspicious nodes around these core members, we take the pruned community as a query subgraph shown in (**d**) and involve n-hop neighborhood into the query subgraph. Then, with respect to the roles of hub and authority, we evaluate every node in the candidate community shown in (**e**) expanded from the query subgraph. Finally, top-K nodes with a high risk score including seed fraudsters and some gain nodes representing unidentified users are selected to form the final risky community shown in (**f**)

3.2 Query decomposition

The Query Decomposition module collects the seed set into subsets by community detection partitioning the whole graph into query subgraphs, following the steps provided by Algorithm 3.

Algorithm 3 Query Decomposition

Input: A graph $G = (V, E, W)$ and a seed set $\mathcal{S}$
Output: A set of query subgraphs SG_q

```
Employ Louvain to partition G into subgraphs by optimizing Eq. (2)
for each subgraph do
    Generate the MST after edge weights are processed by f(x) = e^{-x}
    for each node v_u ∈ V\S in the MST do
        if degree(v_u) < k then
            Remove v_u from the MST
        end if
    end for
    for each pair of nodes, v_i and v_j in the MST do
        if (i, j) ∈ E then
            Fill an edge between v_i and v_j in the MST
        end if
    end for
    Add the MST as a query subgaph into SG_q
end for
return SG_q
```

Without ground truth communities known, modularity [54] is proposed as a global metric that describes the density within communities and the degree of separation between communities. The higher the value of modularity is, the higher the qualitiy of discovered communities is. In order to facilitate application in real scenarios, the size of final communities should be constrained, so we employ a modified modularity as shown in Eq. (2) by introducing a resolution parameter γ. The larger γ is, the smaller query subgraphs we get [58].

$$Q = \frac{1}{2|E|} \sum_{i,j} \left[a_{ij} - \gamma \frac{k_i k_j}{2|E|} \right] \Psi\left(C_i, C_j\right), \tag{2}$$

where $|E|$ denotes the number of edges in the whole graph; the entry a_{ij} of adjacency matrix A serves as the edge weight between nodes v_i and v_j; k_i represents the sum of weights on edges connecting v_i; and C_i determines v_i's community membership. If $C_i = C_j$, i.e., v_i and v_j coming from the same community, $\Psi(C_i, C_j) = 1$; otherwise, $\Psi(C_i, C_j) = 0$.

In eRiskCom, for its scalability to large graphs and high speed, we employ Louvain [6] which performs a greedy optimization of modularity for community detection, a.k.a. graph partition. Louvain is composed of two major steps as follows:

Step 1: *For each node, compare the local contribution to the global modularity value after assigning a node to the community its 1-hop neighbors belong to. Then add the node to a community so that the new graph partition result yields the highest modularity value.*
Step 2: *Create a weighted super-network whose nodes are the subgraphs/communities found in Step 1. Weights between two nodes in the super-network are the sum of weights on edges across the corresponding subgraphs.*
After every node is initially assigned to a separate community, Louvain performs the above two steps alternately until there is no room for modularity increasing and subgraph expansion after shrinking the communities to nodes. Finally, communities with the highest modularity value are returned.

Looking into query subgraphs (i.e., communities returned by Louvain), some could contain many nodes denoting normal users. Thus, it is necessary to perform a pruning operation on these subgraphs to exclude normal users. For this purpose, Kruskal's algorithm [38] and a k-core based filter are combined. Since the pruned subgraphs should maintain the density within, edge weights describing the closeness between nodes need to be processed by a monotonically decreasing function $f(x) = e^{-x}$ before Kruskal's algorithm finds a weighted minimum spanning tree (MST) of each subgraph. Then, for nodes not coming from the seed set, those with a low degree, i.e., users who are less relevant to the risky community, are eliminated by a k-core based filter such that every remaining non-seed node in the MST is connected to at least k ($k = 2$ by default) other nodes. Finally, edges existing between the nodes in the original big graph are refilled into the spanning tree of each subgraph to form the final pruned query subgraphs for subsequent community search.

3.3 Community search

The Community Search module follows the steps detailed in Algorithm 4. It aims to further search the query subgraphs returned by Query Decomposition for discovering risky communities, as shown in Fig. 4. First, we add the 1- and 2-hop neighbors of every node in query subgraphs and refill the corresponding edges for a larger search space. To analyze the topological information, we employ SALSA [40] algorithm based on the stochastic characteristics of random walks on graphs and the theory of Markov chains, to evaluate nodes with respect to the roles of authority and hub in each expanded query subgraph.

HITS (Hyperlink-Induced Topic Search) [36] provides a general framework for finding authoritative webpages by link-structure analysis. It introduces two distinct types of webpages, hubs and authorities, and follows a mutually reinforcing relationship: a good hub directs to numerous authorities, and a good authority is directed to by lots of hubs. In view of such relationship, hubs and authorities are

Algorithm 4 Community Search

Input: A set of query subgraphs SG_q
Output: Candidate risky communities
1: **for** each query subgraph in SG_q **do**
2: Add the n-hop neighborhood of its node to the query subgraph
3: Construct a bipartite graph from the query subgraph
4: Assign an authority score to each node in the query subgraph
5: **end for**
6: **return** updated query subgraphs as candidate risky communities

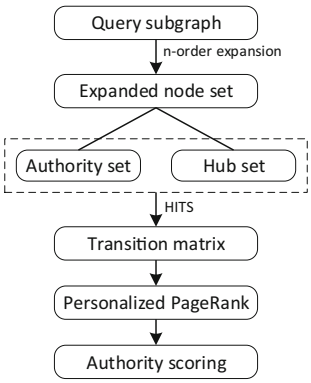

Fig. 4 Details about the Community Search module

supposed to form communities as a dense bipartite graph, where the hubs connect densely to the authorities.

Returned by graph partitioning, the query subgraphs are communities that share dense connections within, so we reshape these subgraphs into authority-hub bipartite graphs, followed by HITS-based approaches. To construct such bipartite graph from a query subgraph, the role of each node should be first identified. The process can be pictured by an example given in Fig. 5. Adjusting to our case, we consider undirected edges as bidirectional links between nodes, so every node in the query subgraph is collected as authority and hub.

However, HITS-based approaches that highlight the mutual reinforcement relationship are susceptible to the Tightly Knit Community (TKC) effect, which sometimes leads to unjustified high ranking positions of webpages of a TKC. Those methods could be stopped from identifying meaningful authoritative webpages [40] by TKC effect. To this end, we consider SALSA [40] algorithm based on the theory of Markov chains that studies random walks on graphs, to evaluate nodes with respect to the roles of authority and hub in each query subgraph.

On a bipartite graph, SALAS performs two diverse random walks, each of which only visits nodes coming from one side of the graph. In each step, traversal paths are composed of two edges, which results in that a walk starts from one side and goes back to this side via a node on the other side. Separate analyses of the two different random walks spontaneously discriminate between the two roles of nodes.

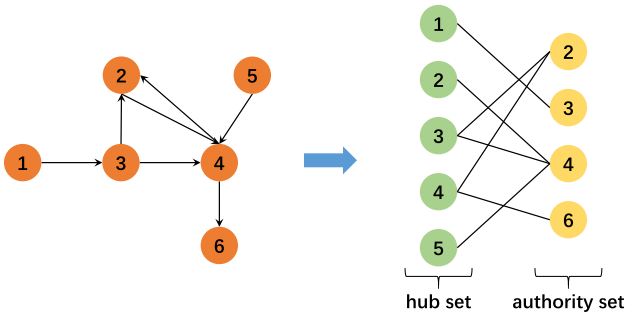

Fig. 5 An example of the bipartite graph transformed from a directed graph, concerning roles of hub and authority for nodes. Nodes with the in-degree larger than 0 are collected into the authority set and ones with the out-degree larger than 0 are collected into the hub set. Then edges between two sets are filled according to edges in the directed graph

Given a bipartite graph $\tilde{G} = \left\{\tilde{V}^{(a)}, \tilde{V}^{(h)}, \tilde{E}\right\}$, where $\tilde{V}^{(a)}$ and $\tilde{V}^{(h)}$ represent a set of nodes serving as authority and hub, respectively, and $\tilde{E}$ denotes the connections between the two sides, SALAS calculates two state transition matrices. The entry of authority matrix $\tilde{A}$ is calculated by

$$\tilde{a}_{i,j} = \sum_{\left\{k|\left(v_k^{(h)}, v_i^{(a)}\right), \left(v_k^{(h)}, v_j^{(a)}\right) \in \tilde{E}\right\}} \frac{1}{w\left(v_i^{(a)}\right) \cdot w\left(v_k^{(h)}\right)}, \quad (3)$$

and the entry of hub matrix $\tilde{H}$ is defined as

$$\tilde{h}_{i,j} = \sum_{\left\{k|\left(v_i^{(h)}, v_k^{(a)}\right), \left(v_j^{(h)}, v_k^{(a)}\right) \in \tilde{E}\right\}} \frac{1}{w\left(v_i^{(h)}\right) \cdot w\left(v_k^{(a)}\right)}, \quad (4)$$

where $v_i^{(a)}$ and $v_i^{(h)}$ denote the node v_i as authority and hub in $\tilde{G}$, respectively, and $w(v_i)$ is the sum of the weights of the edges connecting v_i.

Feeding the authority matrix $\tilde{A}$ as the initial transition matrix, we devise a personalized PageRank (PPR) to calculate the authority score of nodes in the query subgraph. The PPR value of a node u is iteratively updated by

$$\begin{aligned} \mathrm{PPR}_S(u) =& \alpha \sum_{v \in N_{in}(u)}^{N} \frac{1}{|N_{out}(v)|} \mathrm{PPR}_S(v) \\ &+ (1-\alpha) r(u) \end{aligned} \quad (5)$$

until convergence, where $N_{in}(u)$ denotes the set of neighbors with an edge pointing to the node u, $N_{out}(v)$ denotes the set of neighbors that are pointed at by an edge from the node v, and α with the default value 0.85 describes the possibility that the random walk continues to visit for the next step. $r(u)$ is the possibility that the random walk restarts at any node, calculated by

$$r(u) = \begin{cases} \frac{1}{n_q}, & \text{if } u \text{ is in the query subgraph} \\ 0, & \text{otherwise} \end{cases}, \quad (6)$$

where n_q counts the nodes in the query subgraph before expansion. Finally, the final PPR values are output as authority scores for nodes in expanded subgraphs. However, the chances are that some nodes are added to more than one community during expansion. To address this problem, some measures like degree are used to determine which candidate community the node belongs to.

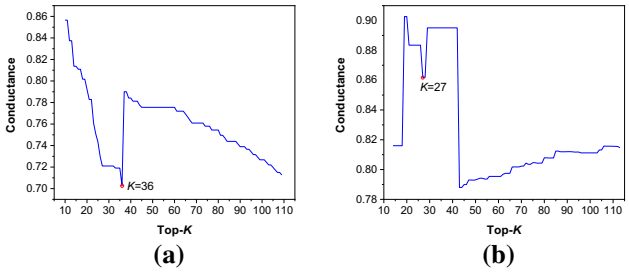

Fig. 6 Two common curves of conductance for a community containing different top-K nodes with a high authority score. The top-K nodes corresponding to a smaller value of conductance is a better choice for risky communities. Considering the difference between curves (**a**, **b**) and the trade-off between the community size and conductance, we uniformly select the value of K at the first local minimum point

3.4 Risk scoring

For the sake of simplicity, we sort the nodes within candidate communities in descending order according to the authority score. Then top-K nodes in each community are selected to compose risky communities fed to human experts for further check.

Conductance has been studied as a scoring function beneficial for the detection of well-separated disjoint communities [72], which meets the requirements of risky communities in industrial scenarios. It describes the connectivity of a community to the rest of the graph relative to the connectivity of the community. To identify top-K for communities in an appropriate way, we calculate the values of conductance under different K. Then we draw a curve of conductance value calculated by Eq. (7) for various K, as an example shown in Fig. 6. A smaller value of conductance means a better community, but meanwhile, a number of unidentified users should be taken into the risky community for further investigation. To make a trade-off between K and conductance value, we choose the K corresponding to the first minimum value of conductance.

$$\Phi(C) = \frac{\sum_{(i,j) \in \partial(C)} w_{ij}}{\min(\sum_{(i,j) \in E_q(C)} w_{ij}, \delta)} \quad (7)$$

Table 1 Statistic of datasets with selected communities for community search

Dataset	#Nodes	#Edges	#Communities
Email	1005	25,571	21
Amazon	334,863	925,872	4129
DBLP	317,080	1,049,866	4748
Youtube	1,134,890	2,987,624	2739

where E_q denotes the edge set of the graph G_q (an expanded query subgrpah in eRiskCom), $E_q(C) = \{(x, y) \in E_q \mid x, y \in C\}$ represents the set of edges connecting nodes in the community C, $\partial(C) = \{(x, y) \in E_q \mid x \in C, y \notin C\}$ stands for a set of edges connecting any node in C and a node out of C, and $\delta = \sum_{(i,j)\in E_q} w_{ij} - \sum_{(i,j)\in E_q(C)} w_{ij}$.

4 Experiments

This section introduces the details of real-world datasets used in experiments in Sect. 4.1. Apart from performance evaluation on the community search task detailed in Sect. 4.2, a series of experiments are conducted to test the effectiveness of eRiskCom with respect to edge weight assignment in Sect. 4.3 and graph partition in Sect. 4.4, and provide an evaluation of gain nodes representing unidentified users in Sect. 4.5. In addition, the benefits that the introduction of suspicious users brings are analyzed in Sect. 4.6 and the patterns of risky communities are studied in Sect. 4.7.

4.1 Experimental setups

4.1.1 Datasets

Since community search is close to risky community detection this work focuses on, we use four existing real-world datasets with ground truth communities from Stanford Large Network Dataset Collection[3] to test the performance of eRiskCom and baselines on the community search task. For three larger datasets (i.e., Amazon, DBLP, and Youtube), we collect their top 5000 communities with highest quality. To further focus on small communities, we select the communities with between 3 and 20 members and randomly choose 30% of nodes from each of them as seed nodes like identified fraudsters in fraud detection scenarios. The statistics of the datasets for community search are summarized in Table 1.

Besides, we collect three real-world datasets involving different fraud detection scenarios from Ant Financial and Taobao for risky community detection. A small number of fraudsters and suspicious users are collected as seed nodes according to reports from victims of fraud. For research purposes, all datasets have been preprocessed to meet the following requirements.

1. No Personal Identifiable Information (PII) is involved in the datasets.
2. The datasets have been desensitized and encrypted.
3. Adequate data protection was provided during the experiment to prevent data copy leakage, and the datasets were destroyed after the experiment.
4. The datasets only serve the academic research purpose and do not represent any real business situation.

- The telecom fraud dataset was generated from the records of users from Alipay within several months in the year of 2020. The semantic information of each user includes account level, user location, education background, etc. To model the interactions between users as a graph, a node represents a unique user and an edge between them indicates the money transfer between user accounts. By reviewing and verifying the reports received from victims of fraud, we identified 16,537 fraudsters who committed any telecom fraud in the following month. After 1-order expansion from identified fraudsters, there are 1,331,393 users and 23,607,526 edges in total.
- The insurance fraud dataset records the user information from Alipay within several months from 2019 to 2020. The type of semantic features of users in this dataset is the same as that in the telecom fraud dataset. Interactions between users are modeled as graphs in the same way as these in the telecom fraud dataset. By reviewing the insurance claim records, we identified 13,680 fraudsters who committed any insurance fraud within seven months from 2019 to 2020, and found 10,141 suspicious users. After 1-order expansion from identified fraudsters and suspicious users, there is a total of 3,917,254 users and 62,805,859 edges.
- The Taobao transaction dataset records the information of online transactions on Taobao, including semantic features of users such as purchased item, the number of user's orders, and payment amount. To find the risky communities containing brushers, we model the relationships between users as a graph according to the similarity of their purchasing preference for the reason that members of brusher groups always purchase some common target items. To be specific, nodes represent buyers and an edge exists between two buyers if they have at least one purchased item in common. However, the co-purchasing graph is very sparse so that it is difficult to detect more risky communities with local density. Thus, we consider filling edges between two nodes if they share at least three neighbors in the co-purchasing graph. By review-

[3] http://snap.stanford.edu/data.

ing the transaction records of a certain type of products within one day in 2020, we identified 22,516 brushers and 34,307 suspicious buyers. After 2-order expansion from identified fraudsters and suspicious users, there are 717,074 nodes and 756,654,814 edges in total.

4.1.2 Baselines

We compare eRiskCom with baselines including:

- LEMON [43] that explores a sparse vector in the span of the local spectra to discover the community,
- MULTICOM [25] which detects multiple local communities by expanding the seed set,
- LEKS [64] that expands a small weighted tree to a connected k-core in a hierarchical manner finally to find an intimate-core group, and
- GraphSage_KNN which finds the K nearest neighbors of a seed node in the embedding space learned by GraphSage [24].

4.1.3 Metrics

The metrics used to analyze the effect of our platform components on the alignment with requirements of real-world applications are specified as below. We employ F1-socre [63] to evaluate the communities searched on datasets with ground truth. From the perspective of practical application, we prefer to get small communities for easy further check by human experts, and focus on communities containing a higher proportion of identified fraudsters and suspicious users since other users close to more fraudsters and suspicious users are more likely to be members of fraud groups. Therefore, we define the following metrics for better comparative analysis.

- $\#SG$ indicates the number of subgraphs returned by graph partition.
- $\#SG_q$ indicates the number of the query subgraphs containing seed nodes.
- $\%SG_q$ denotes the percentage of seed nodes in query subgraphs.
- $\#U_q$ denotes the total number of users in query subgraphs.
- $\#SU_q$ denotes the total number of seed nodes (identified users) in query subgraphs.
- $\%SU_q 2SU$ denotes the percentage of seed nodes in query subgraphs to all seed nodes.
- $\#GU_q$ denotes the total number of the gain nodes (unidentified users) caught in query subgraphs.
- $p(SU)_q$ denotes the proportion of seed nodes in a query subgraph.
- $Size(SG_q)$ denotes the size of a query subgraph, i.e., the number of users within.

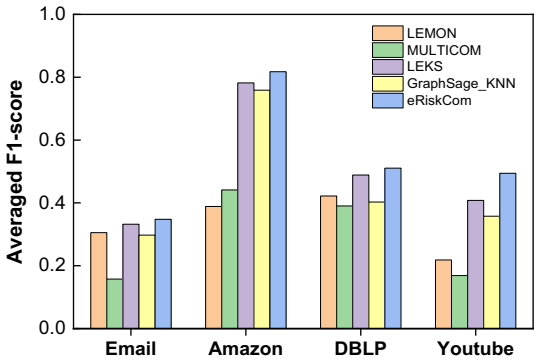

Fig. 7 Evaluation of methods on the community search task

- $\#GU_r$ denotes the total number of the gain nodes caught in risky communities.
- $\#GU_{susp}$ indicates the number of the unidentified users in risky communities who conduct a fraud-like event within a f uture period.
- $CR(GU_{susp})$ denotes the coverage rate of users who conduct a future fraud-like event among gain nodes caught in risky communities.
- $\#GU_{fraud}$ denotes the total number of unidentified users who are real fraudsters caught in risky communities.
- $CR(GU_{fraud})$ denotes the coverage rate of users who are real fraudsters among gain nodes caught in risky communities.
- AC denotes the proportion of real fraudsters in users caught in risky communities.
- $\#SU_r$ denotes the total number of seed nodes in risky communities.

4.2 Community search task

We compare the performance of methods on the community search task to evaluate the effectiveness of our platform eRiskCom in searching the communities of seed nodes. Figure 7 reports the statistics of averaged F1-score on four datasets with ground truth communities, which suggests that eRiskCom can discover communities that mostly resemble the ground truth ones in different scenarios.

4.3 Edge weight assignment

We analyze the effectiveness of edge weight assignment (denoted as RFM&fraud-based and detailed in Sect. 3.1) adopted in eRiskCom by comparing it with three following strategies on the telecom fraud dataset. To figure out whether the labels of seed nodes can provide information of fraud patterns to help risky community detection, a strategy (denoted as RFM-based) only based on the RFM model is involved to compare. Since semantic features could provide useful infor-

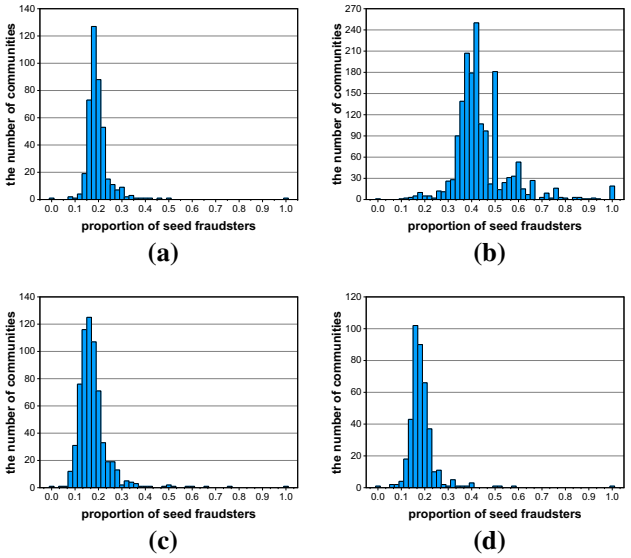

Fig. 8 The distributions of risky communities with different proportions of seed fraudsters on the telecom fraud dataset by employing four edge weight assignment strategies: **a** RFM-based, **b** RFM&fraud-based, **c** first-order similarity, and **d** second-order similarity

mation, we also consider the first- or second-order similarity of semantic features on nodes to set edge weights.

- RFM-based: It only follows the rule that a higher weight is set to the edge between a pair of nodes if frequent or large-amount transactions exist between them.
- First-order similarity: The weight on an edge connecting two nodes equal to the similarity of their semantic features, which is defined based on the normalized cosine distance in the feature space, calculated by $w_{ij} = (cos(x_i, x_j) + 1)/2$ where x_i and $x_j \in R^m$ represent attributes on a pair of connected nodes v_i and v_j, respectively.
- Second-order similarity: Considering the averaged attention of neighborhood, semantic features of nodes and their 1-hop neighbors are averaged to represent node representations. Then edge weights between connected nodes are equal to the similarity of their representations, calculated by normalized cosine distance.

Since an unidentified user close to more fraudsters seems more likely to be potential fraudsters in the group, we prefer to focus on the communities containing a higher proportion of identified fraudsters. Figure 8a, b indicate that the label information of identified users and RFM model can help to divide seed nodes into more subsets and uncover more communities with a higher proportion of seed nodes as we want. Additionally, from Fig. 8a, d, we can also draw a conclusion that RFM-based strategy behaves similarly to the strategy based on the second-order similarity. As observed from Fig. 8, the RFM&fraud strategy performs a priority among all strategies, so eRiskCom employs RFM&fraud strategy to set edge weights.

4.4 Graph partition & pruning

We compare the performance of two different community detection methods, that is, Louvain used in eRiskCom and label propagation algorithm (LPA) [56], applied to graph partition in the Query Decomposition module. The effect of the pruning in the Query Decomposition module is also discussed by comparison of graph partition results with and without pruning.

Table 2 reports some statistics about detected query subgraphs on the telecom fraud dataset and the insurance fraud dataset. It demonstrates that the pruning on subgraphs returned by graph partition helps filter out quantities of nodes that are not closely related to seed fraudsters to find the core members of potential risky communities, and also increases the proportion of identified users in potential risky communities. As observed from Tables 3 and 4, LEMON can detect more query subgraphs with a higher proportion of identified fraudsters than other algorithms without pruning. However, the pruning operation helps LPA achieve the best performance. Identified fraudsters found by LPA account for over 60% in more than half of query subgraphs after pruning in the insurance fraud dataset, and over 70% in more than half of query subgraphs after pruning in the telecom fraud dataset.

Looking into these query subgraphs detected by Louvain and LPA, we find that some communities detected by LPA are merged into a community detected by Louvain. As an example in Fig. 9, the risky community detected by Louvain contains not merely the four communities C_1 to C_4 detected by LPA but also other six seed fraudsters and seven non-seed nodes called gain nodes in this paper.

In terms of performances of two graph partition methods shown in Tables 5 and 6, LPA tends to discover smaller communities than Louvain, but it is flawed in finding the intrinsic connections between fraudsters and unidentified users. Detected by LPA with pruning, around 47.06% and 64.22% of query subgraphs compose of no more than 5 users in the insurance fraud and telecom fraud datasets, respectively, which causes that the proportions of seed fraudsters in most communities are very high and gain nodes are few. In contrast, Louvain can detect larger communities of various sizes where more unidentified users close to seed fraudsters are captured. Although MULTICOM can compete with Louvain_prun with respect to the size of query subgraphs, it performs worse regarding the proportion of identified users in query subgraphs. Therefore, eRiskCom employs Louvain with the pruning operation for practice.

Table 2 Statistics about risky communities on the telecom fraud and insurance fraud datasets

Metric	Telecom fraud				Insurance fraud			
	Louvain	LPA	Louvain_prun	LPA_prun	Louvain	LPA	Louvain_prun	LPA_prun
#SG	17,469	76,610	17,469	76,610	24,857	29,684	24,857	29,684
$\#SG_q$	1628	364	1628	364	1481	58	1481	58
$\%SG_q$	1.19%	2.41%	41.78%	90.88%	0.40%	0.56%	37.02%	91.38%
$\#U_q$	1,019,801	67,845	28,995	1634	2,914,519	38,173	31,805	232
$\#SU_q$	12,114	1485	12,114	1485	11,775	212	11,775	212
$\%SU_q$2SU	73.25%	8.98%	73.25%	8.98%	86.07%	1.55%	86.07%	1.55%
$\#GU_q$	1,007,687	66,211	16,881	149	2,902,744	37,961	20,030	20

Table 3 Distribution of query subgraphs with different proportions of identified users within on the insurance fraud dataset

$p(SU)_q$	Method					
	LEMON	MULTICOM	Louvain	LPA	Louvain_prun	LPA_prun
0% ∼ 2%	0	0.4369	0.9268	0.8315	0	0
2% ∼ 4%	0.3480	0.2816	0.0207	0.0892	0	0
4% ∼ 6%	0.2339	0.1262	0.0193	0.0341	0	0
6% ∼ 8%	0.2251	0.0437	0.0028	0.0105	0	0
8% ∼ 10%	0.0906	0.0291	0.0069	0.0081	0	0
10% ∼ 20%	0.0848	0.0680	0.0193	0.0204	0.0058	0
20% ∼ 30%	0.0117	0.0049	0.0041	0.0062	0.0669	0
30% ∼ 40%	0.0029	0.0097	0	0	0.4448	0
40% ∼ 50%	0	0	0	0	0.3953	0.0588
50% ∼ 60%	0	0	0	0	0.0727	0.4118
60% ∼ 70%	0	0	0	0	0.0145	0.4706
70% ∼ 80%	0	0	0	0	0	0.0294
80% ∼ 90%	0	0	0	0	0	0
90% ∼ 100%	0.0029	0	0	0	0	0.0294

Table 4 Distribution of query subgraphs with different proportions of identified users within on the telecom fraud dataset

$p(SU)_q$	Method					
	LEMON	MULTICOM	Louvain	LPA	Louvain_prun	LPA_prun
0% ∼ 2%	0	0.0382	0.7372	0.6009	0	0
2% ∼ 4%	0.1431	0.1431	0.1865	0.1674	0	0
4% ∼ 6%	0.1755	0.1176	0.0401	0.1030	0	0
6% ∼ 8%	0.2522	0.1510	0.0200	0.0687	0	0
8% ∼ 10%	0.1563	0.1669	0.0100	0.0215	0	0
10% ∼ 20%	0.2065	0.2925	0.0063	0.0343	0.0038	0
20% ∼ 30%	0.0383	0.0556	0	0	0.0289	0
30% ∼ 40%	0.0206	0.0286	0	0	0.3036	0.0172
40% ∼ 50%	0.0044	0.0016	0	0.0043	0.3877	0.0172
50% ∼ 60%	0	0.0016	0	0	0.1644	0.2241
60% ∼ 70%	0.0015	0.0016	0	0	0.0590	0.1940
70% ∼ 80%	0	0.0016	0	0	0.0176	0.0819
80% ∼ 90%	0	0	0	0	0.0050	0.0517
90% ∼ 100%	0.0015	0	0	0	0.0301	0.4138

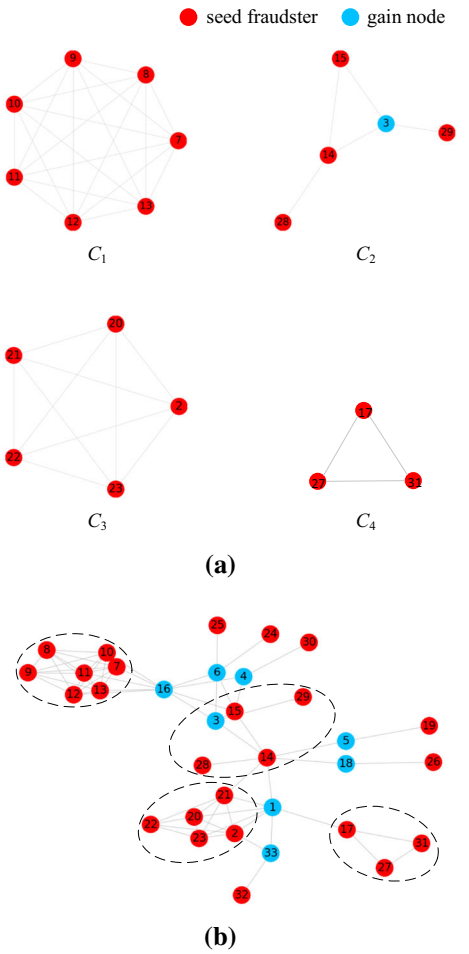

Fig. 9 Comparison of query subgraphs detected by **a** LPA and **b** Louvain on the telecom fraud dataset. Nodes in red represent seed fraudsters and nodes in blue denote gain nodes (color figure online)

For the insurance fraud dataset, one of the communities detected by Louvain is visualized as an example shown in Fig. 10. The three gain nodes in blue are connected to multiple fraudsters, which indicates that the three users in this risky community are active. Nodes 23 and 4 may be hubs of which one introduces nodes 7 and 19 to join this community and the other introduces node 3.

4.5 Evaluation on gain nodes

To verify the effectiveness of our platform eRiskCom for risky community detection, we investigate the gain nodes that represent the unidentified users in risky communities. Table 7 reports how many gain users would apply for an insurance claim in the next three months. Compared with k-core, a rule-based method, eRiskCom can find more unidentified users closely related to risky communities and meanwhile, cover more users who would apply for an insurance claim in the future. Unidentified users in risky communities may not conduct an insurance fraud event at this time, but they are likely to commit fraud when applying for an insurance claim in the future. Therefore, eRiskCom can help human experts focus on these unidentified users who would conduct a future fraud.

Table 8 reports how many gain nodes caught by k-core (denoted as Rule-based) and eRiskCom in the telecom fraud dataset are real fraudsters, which is checked by human experts. In the Query Decomposition module, k-core is used to prune nodes less related to risky communities, and meanwhile, the gain nodes should be connected to at least two seed nodes for a practical purpose. With various settings of k except for $k = 2$, eRiskCom can detect more fraudsters more accurately. It can be concluded that the introduction of community information could help improve the accuracy of fraudster detection. However, as k increases, eRiskCom detects fraudsters by risky community detection more accurately while the number of discovered fraudsters goes down. Hence, a trade-off between accuracy and quantity should be considered for real-world applications.

Table 5 Distribution of query subgraphs of different sizes on the insurance fraud dataset

$Size(SG_q)$	Method					
	LEMON	MULTICOM	Louvain	LPA	Louvain_prun	LPA_prun
$3 \sim 5$	0	0.0147	0.0041	0.0062	0.0145	0.4706
$6 \sim 10$	0	0.0245	0.0193	0.0204	0.4971	0.3824
$11 \sim 20$	0	0.0392	0.0207	0.0341	0.3459	0.0588
$21 \sim 30$	0.0029	0.1422	0.0124	0.0384	0.0843	0.0294
$31 \sim 50$	0.0117	0.3529	0.0138	0.0700	0.0494	0
$51 \sim 100$	0.9854	0.2843	0.0373	0.1772	0.0058	0.0588
$101 \sim 200$	0	0.1225	0.0704	0.2596	0.0029	0
$201 \sim 500$	0	0.0196	0.2818	0.2763	0	0
$500 \sim 1000$	0	0	0.2086	0.0799	0	0
> 1000	0	0	0.3315	0.0378	0	0

Table 6 Distribution of query subgraphs of different sizes on the telecom fraud dataset

$Size(SG_q)$	Method					
	LEMON	MULTICOM	Louvain	LPA	Louvain_prun	LPA_prun
$3 \sim 5$	0	0.0827	0	0	0.0652	0.6422
$6 \sim 10$	0	0.0535	0	0	0.3915	0.2543
$11 \sim 20$	0	0.2496	0	0.0043	0.3965	0.0948
$21 \sim 30$	0.0118	0.3112	0.0038	0.0258	0.1029	0.0043
$31 \sim 50$	0.0369	0.2545	0.0175	0.0687	0.0326	0
$51 \sim 100$	0.9513	0.0389	0.0538	0.1502	0.0113	0.0043
$101 \sim 200$	0	0.0097	0.1302	0.1760	0	0
$201 \sim 500$	0	0	0.3830	0.2575	0	0
$500 \sim 1000$	0	0	0.2616	0.1459	0	0
> 1000	0	0	0.1502	0.1717	0	0

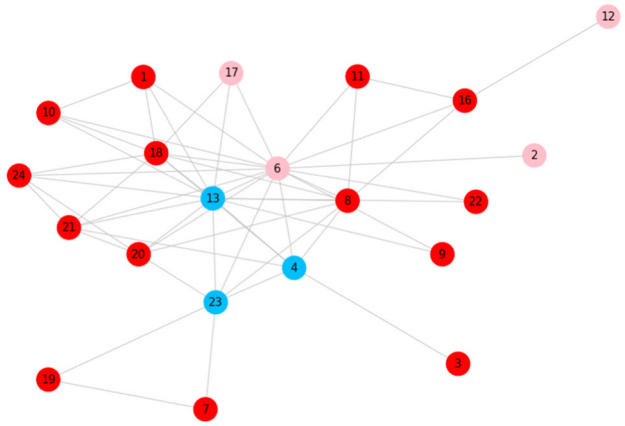

Fig. 10 A query subgraph detected by Louvain on the insurance fraud dataset. Nodes in red, pink, and blue represent seed fraudsters, suspicious nodes, and gain nodes, respectively (color figure online)

Table 7 Evaluation on unidentified users in risky communities for future insurance fraud

Method	$\#GU_r$	$\#GU_{susp}$	$CR(GU_{susp})$
k-core ($k = 2$)	21,457	358	9.74%
k-core ($k = 3$)	939	91	2.47%
eRiskCom	**91,971**	**664**	**18.06%**

Better results are highlighted in bold

Table 8 Statistics about gain nodes and the detection accuracy in the telecom fraud dataset

k-core	Rule-based			
	$\#GU_r$	$\#GU_{fraud}$	$CR(GU_{fraud})$	AC
2	**15,871**	5870	36.99%	49.85%
3	2268	1019	44.93%	74.67%
4	613	255	41.60%	77.81%
5	204	86	42.16%	88.43%
k-core	eRiskCom			
	$\#GU_r$	$\#GU_{fraud}$	$CR(GU_{fraud})$	AC
2	5452	2165	**39.71%**	**63.26%**
3	**2597**	1198	**46.13%**	**79.53%**
4	**1279**	641	**50.12%**	**82.04%**
5	**710**	361	**50.85%**	**91.15%**

Better results are highlighted in bold

Table 9 Evaluation on the effect of identified suspicious users on the Taobao transaction dataset (SSU is short for 'seed suspicious users')

SSU	$\#SU_r$	$\#GU_r$	AC
×	14,711	13,580	63.40%
√	**28,388**	**34,389**	**70.80%**

Better results are highlighted in bold

4.6 Introduction of suspicious users

With some prior knowledge of identified fraudsters in different scenarios, some users who behave similarly to these fraudsters can be found as suspicious users by human experts. The introduction of these suspicious users to seed nodes could provide more information for risky community detection. A set of comparative experiments with/without seed suspicious users are carried out on the Taobao transaction dataset. The results reported in Table 9 suggest that information of suspicious users can help to have more identified users and find more high-risk users waiting to be further investigated by human experts. Moreover, the accuracy of catching real fraudsters within detected risky communities has been increased by over 10%. Therefore, considering the information of suspicious users benefits risky community detection in e-commerce scenarios.

4.7 Patterns of risky communities

Even if risky communities known as fraud gangs are detected with a high accuracy, the fraud patterns organized within are

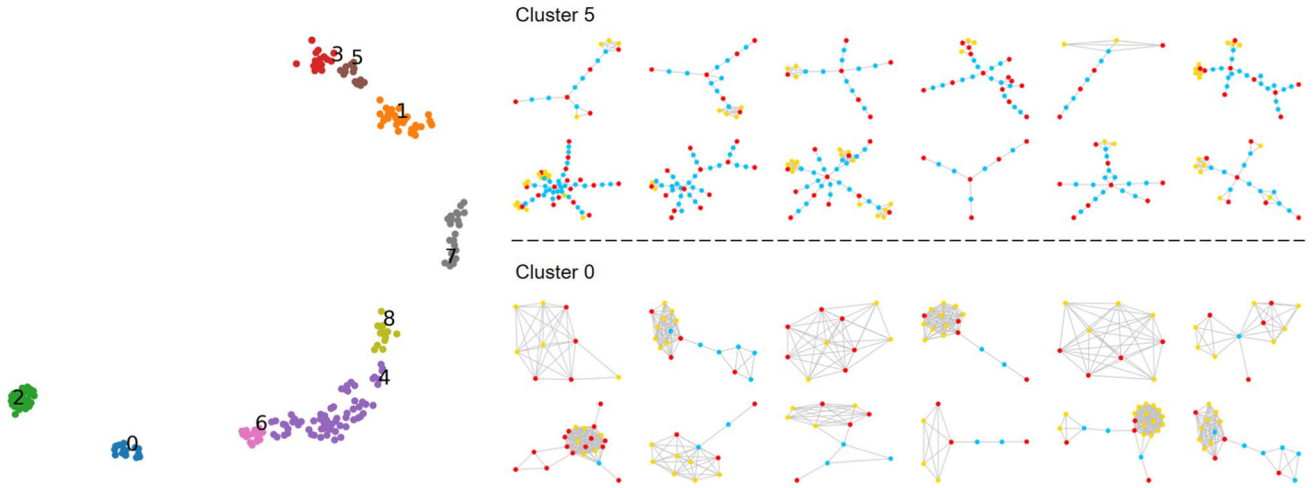

Fig. 11 Visualization of the risky community clustering result on the telecom fraud dataset. The subfigure in the left column demonstrates that risky communities are well divided into nine clusters. Risky communities belonging to cluster 5 and cluster 0 are displayed in the right column for fraud pattern analysis. In each community, nodes in red represent the seed fraudsters, and nodes in blue or yellow represent unidentified users waiting for further check

sometimes difficult to explain and understand. Here we take telecom fraud as an example to look into the fraud patterns of risky communities detected by our platform EriskCom. We first employ a recent graph-level representation learning method [67] using graph auto-encoder model [34] to learn the representations of risky communities based on subgraphs sampled by PageRank-based random walk and then discover clusters of communities by DBSCAN [18], a classical density-based clustering algorithm. Risky communities are divided into nine clusters, as visualized in Fig. 11. Here we take two clusters, i.e., cluster 5 and cluster 0, to study their fraud patterns. For cluster 5, identified fraudsters are not connected directly but connected by other hub users, which means the fraudsters could transfer money through hub accounts to hide from being discovered. For cluster 0, members are more densely connected in the main part of the risky communities, which demonstrates that money transfer frequently occurs among the majority of members.

5 Related work

Following the principle that anomalies are the minority, graph-based anomaly detection aims to uncover the abnormal objects, i.e., nodes, edges, or subgraphs that are rare and significantly differ from the majority of the reference objects in graphs [3,48]. Benefiting from the rapid development of graph-based anomaly detection approaches, various types of frauds in finance, e-commerce, security, and many other scenarios, such as credit card fraud [39], review/opinion spam [57], rumors [28], and network intrusion [16], have been studied. Our work focuses on the detection of risky communities, so we review related works on anomalous subgraph detection, community detection, and community search which are closely associated with our task.

5.1 Anomalous subgraph detection

In social networks, fraudsters create fake identities and randomly interact with innocent users. [61] searched for subgraphs regarding the collaborative malicious activities modeled by their defined Random Link Attacks (RLA) property. [2] focused on the patterns that egonets (i.e., 1-hop neighborhood) in a social network obey, and detected anomalous subgraphs of various types such as near-clique and near-star. Inspired from the signal processing area, spectral norms of graph's modularity matrix from community detection [54] have been used to detect small, highly anomalous subgraphs [50,51]. Dense subgraph detection has also been applied to some scenarios where fraudsters usually add a lot of connections to create locally dense regions in the whole graph [26,32]. Recently, local graph clustering has been applied to a system for small-scale fraud communities that fraud accounts form [47]. However, the above works study various patterns of anomalous subgraphs but do not consider the semantic features on nodes observed in real-world scenarios.

One of the early studies on attributed graph anomaly detection [55] follows the main idea that subgraphs with frequent substructures [65,69] are less abnormal than those with few frequent substructures. Based on the SUBDUE system [11] which compresses the networks with frequent substructures to get the candidate subgraphs, subgraphs formed by less compression are anomalous for they contain fewer frequent substructures. Later studies consider numeric

attributes [14] and activities of node/edge including insertion, deletion, and label modification [17]. Different from the aforementioned approaches, [23] indicated that the existence of low-probability edges and the absence of high-probability edges can be used to score the anomaly of subgraphs. In the setting of bipartite graphs that usually model relationships (e.g., review and purchasing) between users and products, semantic features such as keywords and ratings, could be modeled as a tensor for further dense subgraph detection [31,60]. Unlike previous works that only consider shallow topology structure, [66] considered deep topology structure to embed nodes into a latent space where the suspicious users in the same fraud group are as close as possible while normal users distribute uniformly, so that fraud groups can be easily discovered by density-based methods.

5.2 Community detection

The concept of community originates in the field of network science. In a network, communities are subgraphs within which nodes share dense connections between each other while between which nodes share sparse connections [63]. Conventional community detection methods mainly consider the network topology when partitioning the whole graph into communities. Hierarchical clustering studies hierarchical organization of communities in networks by a divisive [22] or agglomerative [52] way. Stochastic block models are widely applied to study overlapping communities by assigning nodes into communities based on probabilities of likelihood. Random walks running dynamical processes [59] and label propagation algorithm following a message passing mechanism [56] are the most popular community detection methods based on dynamics. Spectral clustering utilizes the eigenvectors of network topological matrices to uncover communities [53]. As modularity [54] was proposed to evaluate community detection results, optimization methods [6,44,45] have been developed for community detection by modularity maximization. According to the latest investigation [46,63], researchers are keen on applying deep learning techniques to community detection for their advantage of handling nonlinear topological structure and semantic information from rich sources. However, these community detection approaches that divide the entire graph into communities do not tell whether a detected community is risky or not. Furthermore, they cannot exclude the normal users less related to risky communities and identify core members of risky communities, while eRiskCom can handle the two concerns in its modules as detailed in Sect. 3.

5.3 Community search

Community search is another popular task different from community detection. The goal of community search is to seek high-quality communities based on query request. Given a node, it aims to find a community that contains the node and satisfies the properties of connectivity and cohesiveness. Following a recent survey [20], community search approaches can be studied by five categories according to adopted cohesiveness metrics. (1) The k-core based approaches aim to find the largest connected subgraph(s) in which each node has at least k neighbors [5]. [62] proposed a greedy global search algorithm, which follows the peeling framework of computing the densest subgraphs and removes nodes iteratively. [4,13] tried to reduce the size of returned communities with a smaller size. For large graphs, [64] constructed a small-weighted spanning tree connecting query nodes, and then expanded it to a k-core in a hierarchical fashion. [19] designed a CL-tree index structure which takes a linear time cost to uncover the connected k-core regarding the keywords held by a query node. (2) The k-truss based approaches target searching the largest subgraph(s) in which every edge is contained in at least k-2 triangles [10], i.e., computing the k-truss community or finding the closest communities. [29] proposed an index that supports the k-truss community query in the optimal time. To further improve the efficiency, [1] introduced another index which preserves the k-truss cohesiveness and the triangle connectivity in the triangle-connected truss communities. The closest model [30] is suitable to discover the closest communities each of which contains multiple query nodes rather than one single query node by [1,29]. (3) The k-clique based approaches are dedicated to discovering completely connected subgraphs of k nodes in which any two of nodes are connected by edge. Faced with the overwhelming number of k-cliques communities in real life, [12] focused on searching for communities containing a given query node and further relaxed the k-clique constraints in practice. During the search process, nodes can be pruned by considering the acquaintance constraint and social radius constraint [71]. Besides cohesiveness, the influence of community was also considered in [41]. (4) The k-ECC (k-edge-connected component)-based approaches aim to search for a subgraph still connected after removing any k-1 edges [21]. [7] computed the maximum Steiner Maximum-Connected Subgraph (SMCS), i.e., a subgraph with maximum edge connectivity, containing a set of query nodes. The extremely large and complex maximum SMCS makes it difficult to be feasible for real-world applications. To address this issue, [27] studied the minimal SMCS problem. (5) Other metrics-based approaches expand the community/subgraph from query node(s) with a specific scoring function such as local modularity [9], query biased density [68,70], personalized PageRank [25,37], and local spectra [43].

6 Conclusions and future work

We propose eRiskCom, a platform for e-commerce risky community detection. Taking in observed fraudsters and suspicious users as seeds, eRiskCom uncovers risky communities containing seeds and unobserved potential fraudsters. Four major modules make use of the information of seeds and relationships between them via graph modeling. We study the impact of diverse edge weight assignment strategies on the distribution of seed fraudsters in detected risky communities, and a fraud-based strategy based on adjacency of fraudsters which helps catch more fraudsters in risky communities is employed in eRiskCom. In terms of graph partition, Louvain performs better than LPA for Louvain can detect more complete communities taking in more seed fraudsters and other users who need further investigation. The pruning operation helps filter out users less related to seed fraudsters and seed suspicious users. Besides, the introduction of seed suspicious users in eRiskCom could provide additional information to help detect risky communities. In e-commerce scenarios, eRiskCom can detect risky communities containing more unidentified users and discover unobserved fraudsters more accurately, and the patterns of detected risky communities are easy to understand.

With the help of the platform, regulators take measures for targeted control, which enforces that fraudsters may interact more frequently with non-fraudsters to camouflage. eRiskCom will be developed to detect hidden fraudsters and uncover more complex patterns of fraud groups. Since the majority of victims do not report through financial platforms, we cannot always acquire prior knowledge of risky communities. To this end, we will develop the capacity of eRiskCom to uncover those communities without any fraudsters or suspicious users observed. As fraud groups in the real world are changing over time, eRiskCom will also be developed to track the evolution of risky communities. To exploit the patterns of risky communities detected by eRiskCom to guide the graph modeling for further improvement, a quantitative characterization [15] of fraud groups will be provided. While utilizing richer user information not considered in this work, future efforts should look into the fairness issue regarding sensitive demographic features [49] without marginalizing at-risk groups.

Acknowledgements This work was supported by the ARC DECRA Project (No. DE200100964). Dr. Li and Dr. Wu are the corresponding authors.

References

1. Akbas, E., Zhao, P.: Truss-based community search: a truss-equivalence based indexing approach. Proc. VLDB Endow. **10**(11), 1298–1309 (2017)
2. Akoglu, L., McGlohon, M., Faloutsos, C.: OddBall: spotting anomalies in weighted graphs. In: PAKDD, pp. 410–421 (2010)
3. Akoglu, L., Tong, H., Koutra, D.: Graph based anomaly detection and description: a survey. DMKD **29**, 626–688 (2015)
4. Barbieri, N., Bonchi, F., Galimberti, E., Gullo, F.: Efficient and effective community searchs. DMKD **29**, 1406–1433 (2015)
5. Batagelj, V., Zaversnik, M.: An O(m) algorithm for cores decomposition of networks. arXiv preprint arXiv:cs/0310049 (2003)
6. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. J. Stat. Mech. **2008**(10), P10008 (2008)
7. Chang, L., Lin, X., Qin, L., Yu, J.X., Zhang, W.: Index-based optimal algorithms for computing steiner components with maximum connectivity. In: SIGMOD, pp. 459–474 (2015)
8. Chen, Y.L., Kuo, M.H., Wu, S.Y., Tang, K.: Discovering recency, frequency, and monetary (RFM) sequential patterns from customers' purchasing data. Electron. Commer. Rec. Appl. **8**(5), 241–251 (2009)
9. Clauset, A.: Finding local community structure in networks. Phys. Rev. E **72**, 026132 (2005)
10. Cohen, J.: Trusses: cohesive subgraphs for social network analysis. Technical report, National Security Agency (2008)
11. Cook, D.J., Holder, L.B.: Graph-based data mining. IEEE Intell. Syst. Appl. **15**(2), 32–41 (2000)
12. Cui, W., Xiao, Y., Wang, H., Lu, Y., Wang, W.: Online search of overlapping communities. In: SIGMOD, pp. 277–288 (2013)
13. Cui, W., Xiao, Y., Wang, H., Wang, W.: Local search of communities in large graphs. In: SIGMOD, pp. 991–1002 (2014)
14. Davis, M., Liu, W., Miller, P., Redpath, G.: Detecting anomalies in graphs with numeric labels. In: CIKM, pp. 1197–1202 (2011)
15. Derzsy, N., Majumdar, S., Malik, R.: An interpretable graph-based mapping of trustworthy machine learning research. In: Complex Networks XII, pp. 73–85 (2021)
16. Ding, Q., Katenka, N., Barford, P., Kolaczyk, E., Crovella, M.: Intrusion as (anti)social communication: characterization and detection. In: KDD, pp. 886–894 (2012)
17. Eberle, W., Holder, L.: Discovering structural anomalies in graph-based data. In: ICDMW, pp. 393–398 (2007)
18. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD, pp. 226–231 (1996)
19. Fang, Y., Cheng, R., Chen, Y., Luo, S., Hu, J.: Effective and efficient attributed community search. VLDB J. **26**, 803–828 (2017)
20. Fang, Y., Huang, X., Qin, L., Zhang, Y., Cheng, R., Lin, X.: A survey of community search over big graphs. VLDB J. **29**, 353–392 (2020)
21. Gibbons, A.: Algorithmic Graph Theory. Cambridge University Press, Cambridge (1985)
22. Girvan, M., Newman, M.E.: Community structure in social and biological networks. PNAS **99**(12), 7821–7826 (2002)
23. Gupta, M., Mallya, A., Roy, S., Cho, J.H.D., Han, J.: Local learning for mining outlier subgraphs from network datasets. In: SDM, pp. 73–81 (2014)
24. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: NIPS, pp. 1024–1034 (2017)
25. Hollocou, A., Bonald, T., Lelarge, M.: Multiple local community detection. SIGMETRICS Perform. Eval. Rev. **45**(3), 76–83 (2018)
26. Hooi, B., Shin, K., Song, H.A., Beutel, A., Shah, N., Faloutsos, C.: Graph-based fraud detection in the face of camouflage. ACM TKDD **11**(4) (2017)
27. Hu, J., Wu, X., Cheng, R., Luo, S., Fang, Y.: On minimal steiner maximum-connected subgraph queries. IEEE TKDE **29**(11), 2455–2469 (2017)
28. Huang, Q., Zhou, C., Wu, J., Wang, M., Wang, B.: Deep structure learning for rumor detection on twitter. In: IJCNN, pp. 1–8 (2019)

29. Huang, X., Cheng, H., Qin, L., Tian, W., Yu, J.X.: Querying k-truss community in large and dynamic graphs. In: SIGMOD, pp. 1311–1322 (2014)
30. Huang, X., Lakshmanan, L.V.S., Yu, J.X., Cheng, H.: Approximate closest community search in networks. Proc. VLDB Endow. **9**(4), 276–287 (2015)
31. Jiang, M., Beutel, A., Cui, P., Hooi, B., Yang, S., Faloutsos, C.: A general suspiciousness metric for dense blocks in multimodal data. In: ICDM, pp. 781–786 (2015)
32. Jiang, M., Cui, P., Beutel, A., Faloutsos, C., Yang, S.: Inferring strange behavior from connectivity pattern in social networks. In: PAKDD, pp. 126–138 (2014)
33. Jurgovsky, J., Granitzer, M., Ziegler, K., Calabretto, S., Portier, P.E., He-Guelton, L., Caelen, O.: Sequence classification for credit-card fraud detection. Expert Syst. Appl. **100**, 234–245 (2018)
34. Kipf, T.N., Welling, M.: Variational graph auto-encoders. In: Bayesian Deep Learning Workshop, NIPS (2016)
35. Kirlidog, M., Asuk, C.: A fraud detection approach with data mining in health insurance. Procedia Soc. Behav. Sci. **62**, 989–994 (2012)
36. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. J. ACM **46**(5), 604–632 (1999)
37. Kloumann, I.M., Kleinberg, J.M.: Community membership identification from small seed sets. In: KDD, pp. 1366–1375 (2014)
38. Kruskal, J.B.: On the shortest spanning subtree of a graph and the traveling salesman problem. Proc. Am. Math. Soc. **7**(1), 48–50 (1956)
39. Lebichot, B., Braun, F., Caelen, O., Saerens, M.: A graph-based, semi-supervised, credit card fraud detection system. In: Complex Networks, pp. 721–733 (2016)
40. Lempel, R., Moran, S.: SALSA: the stochastic approach for link-structure analysis. ACM TOIS **19**(2), 131–160 (2001)
41. Li, J., Wang, X., Deng, K., Yang, X., Sellis, T., Yu, J.X.: Most influential community search over large social networks. In: ICDE, pp. 871–882 (2017)
42. Li, X., Liu, S., Li, Z., Han, X., Shi, C., Hooi, B., Huang, H., Cheng, X.: Flowscope: spotting money laundering based on graphs. In: AAAI, pp. 4731–4738 (2020)
43. Li, Y., He, K., Bindel, D., Hopcroft, J.E.: Uncovering the small community structure in large networks: a local spectral approach. In: WWW, pp. 658–668 (2015)
44. Liu, F., Wu, J., Xue, S., Zhou, C., Yang, J., Sheng, Q.: Detecting the evolving community structure in dynamic social networks. World Wide Web **23**, 715–733 (2020)
45. Liu, F., Wu, J., Zhou, C., Yang, J.: Evolutionary community detection in dynamic social networks. In: IJCNN, pp. 1–7 (2019)
46. Liu, F., Xue, S., Wu, J., Zhou, C., Hu, W., Paris, C., Nepal, S., Yang, J., Yu, P.S.: Deep learning for community detection: progress, challenges and opportunities. In: IJCAI, pp. 4981–4987 (2020)
47. Ma, J., Zhang, D., Wang, Y., Zhang, Y., Pozdnoukhov, A.: GraphRAD: a graph-based risky account detection system. In: MLG (2018)
48. Ma, X., Wu, J., Xue, S., Yang, J., Zhou, C., Sheng, Q.Z., Xiong, H., Akoglu, L.: A comprehensive survey on graph anomaly detection with deep learning. IEEE TKDE (2021)
49. Majumdar, S.: Fairness, explainability, privacy, and robustness for trustworthy algorithmic decision making. In: S. Basak, M. Vračko (eds.) Big Data Analytics in Chemoinformatics and Bioinformatics. Elsevier (2022)
50. Miller, B.A., Beard, M.S., Wolfe, P.J., Bliss, N.T.: A spectral framework for anomalous subgraph detection. IEEE TSP **63**(16), 4191–4206 (2015)
51. Miller, B.A., Bliss, N.T., Wolfe, P.J.: Subgraph detection using eigenvector L1 norms. In: NIPS, pp. 1633–1641 (2010)
52. Newman, M.E.J.: Fast algorithm for detecting community structure in networks. Phys. Rev. E **69**, 066133 (2004)
53. Newman, M.E.J.: Spectral methods for community detection and graph partitioning. Phys. Rev. E **88**, 042822 (2013)
54. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E **69**, 026113 (2004)
55. Noble, C.C., Cook, D.J.: Graph-based anomaly detection. In: KDD, pp. 631–636 (2003)
56. Raghavan, U.N., Albert, R., Kumara, S.: Near linear time algorithm to detect community structures in large-scale networks. Phys. Rev. E **76**, 036106 (2007)
57. Rayana, S., Akoglu, L.: Collective opinion spam detection: bridging review networks and metadata. In: KDD, pp. 98–994 (2015)
58. Reichardt, J., Bornholdt, S.: Statistical mechanics of community detection. Phys. Rev. E **74**, 016110 (2006)
59. Rosvall, M., Bergstrom, C.T.: Maps of random walks on complex networks reveal community structure. PNAS **105**(4), 1118–1123 (2008)
60. Shin, K., Hooi, B., Faloutsos, C.: Fast, accurate, and flexible algorithms for dense subtensor mining. ACM TKDD **12**(3) (2018)
61. Shrivastava, N., Majumder, A., Rastogi, R.: Mining (social) network graphs to detect random link attacks. In: ICDE, pp. 486–495 (2008)
62. Sozio, M., Gionis, A.: The community-search problem and how to plan a successful cocktail party. In: KDD, pp. 939–948 (2010)
63. Su, X., Xue, S., Liu, F., Wu, J., Yang, J., Zhou, C., Hu, W., Paris, C., Nepal, S., Jin, D., Sheng, Q.Z., Yu, P.S.: A comprehensive survey on community detection with deep learning. arXiv preprint arXiv:2105.12584 (2021)
64. Sun, L., Huang, X., Li, R., Choi, B., Xu, J.: Index-based intimate-core community search in large weighted graphs. IEEE TKDE (2020)
65. Sun, Q., Li, J., Peng, H., Wu, J., Ning, Y., Yu, P.S., He, L.: SUGAR: subgraph neural network with reinforcement pooling and self-supervised mutual information mechanism. In: WWW, pp. 2081–2091 (2021)
66. Wang, H., Zhou, C., Wu, J., Dang, W., Zhu, X., Wang, J.: Deep structure learning for fraud detection. In: ICDM, pp. 567–576 (2018)
67. Wang, L., Zong, B., Ma, Q., Cheng, W., Ni, J., Yu, W., Liu, Y., Song, D., Chen, H., Fu, Y.: Inductive and unsupervised representation learning on graph structured objects. In: ICLR (2020)
68. Wang, Z., Wang, W., Wang, C., Gu, X., Li, B., Meng, D.: Community focusing: yet another query-dependent community detection. In: AAAI, pp. 329–337 (2019)
69. Wu, J., Zhu, X., Zhang, C., Yu, P.S.: Bag constrained structure pattern mining for multi-graph classification. IEEE TKDE **26**(10), 2382–2396 (2014)
70. Wu, Y., Jin, R., Li, J., Zhang, X.: Robust local community detection: On free rider effect and its elimination. Proc. VLDB Endow. **8**(7), 798–809 (2015)
71. Yang, D.N., Chen, Y.L., Lee, W.C., Chen, M.S.: On social-temporal group query with acquaintance constraint. Proc. VLDB Endow. **4**(6), 397–408 (2011)
72. Yang, J., Leskovec, J.: Defining and evaluating network communities based on ground-truth. Knowl. Inf. Syst. **42**, 181–213 (2015)
73. Zhang, G., Zhao, L., Huang, J., Wu, J., Zhou, C., Yang, J.: eFraudCom: an e-commerce fraud detection system via competitive graph neural networks. ACM TOIS (2021)
74. Zhang, Y., Bian, J., Zhu, W.: Trust fraud: a crucial challenge for China's e-commerce market. Electron. Commer. Rec. Appl. **12**(5), 29–308 (2013)