Detecting Outlier Patterns with Query-based Artificially Generated Searching Conditions

Shuo Yu  
School of Software  
Dalian University of Technology  
Dalian, China  
y_shuo@outlook.com

Feng Xia  
School of Engineering, IT and Physical Sciences  
Federation University Australia  
Ballarat, Australia  
f.xia@ieee.org

Yuchen Sun  
School of Software  
Dalian University of Technology  
Dalian, China  
aircleven@outlook.com

Tao Tang  
School of Software  
Dalian University of Technology  
Dalian, China

Xiaoran Yan  
Network Science Institute  
Indiana University  
Ballarat  
e-mail: yan30@iu.edu

Ivan Lee  
School of Information Technology and Mathematical Sciences  
University of South Australia  
Australia  
ivan.lee@unisa.edu.au

ABSTRACT
In the age of social computing, finding interesting network patterns or motifs is significant and critical for various areas such as decision intelligence, intrusion detection, medical diagnosis, social network analysis, fake news identification, national security, etc. However, sub-graph matching remains a computationally challenging problem, let alone identifying special motifs among them. This is especially the case in large heterogeneous real-world networks. In this work, we propose an efficient solution for discovering and ranking human behavior patterns based on network motifs by exploring a user’s query in an intelligent way. Our method takes advantage of the semantics provided by a user’s query, which in turn provides the mathematical constraint that is crucial for faster detection. We propose an approach to generate query conditions based on the user’s query. In particular, we use meta paths between nodes to define target patterns as well as their similarities, leading to efficient motif discovery and ranking at the same time. The proposed method is examined on a real-world academic network, using different similarity measures between the nodes. The experiment result demonstrates that our method can identify interesting motifs, and is robust to the choice of similarity measures.

KEYWORDS
human behavior, outlier detection, social computing, heterogeneous network, motif

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference’17, July 2017, Washington, DC, USA  
© 2022 Association for Computing Machinery.  
ACM ISBN 978-1-4503-xxxx-xx/YY/MM. . . $15.00  
https://doi.org/10.1145/nnnnnnn.mmmmm

1 INTRODUCTION
Networks have been extensively utilized to study the interactions among entities in many fields like social computing. In particular, heterogeneous information networks can model complex, large-scale data sets by introducing multiple node and edge types, leading to more detailed and realistic applications in the real world [11, 18, 21]. Heterogeneous edges can be used to capture relationships among different types of nodes, and edge weights are used to evaluate the strengths of relationships. For example, Figure 1 shows a bibliographic network consisting of authors, papers, and venues. In this heterogeneous information network, paper authorship, paper citations, and venue publication relationships are captured at the same time. Additional co-authorship edges can be added to capture collaborative/social relationships between authors, and the collaborative strength can be evaluated by collaboration times [33].

In the recent past, lots of effort has been placed into the study of heterogeneous information networks and various relationships

Figure 1: An example of a bibliographic network.
among entities. Finding unexpected events and detecting outliers from normal patterns have always been an interesting topic in various disciplines [3, 17]. It can reveal hidden patterns, which may also guide policymakers to make better decisions. These outliers may carry significant information in abundant fields that include, but not limited to, intrusion detection [12], medical diagnosis [24], social network analysis [10], fake news identification [28], public security monitoring [35] and national security [31]. While widely applied to high-dimensional data, uncertain data, streaming data, network data, time-series data, etc., studies about outlier detection mainly focus on individually abnormal node or outlier pairs [2, 8, 14]. Various survey papers summarize the existing outlier detection algorithms from abundant perspectives [5, 23, 34].

Beyond node or pairs, outlier detection based on sub-graphs or motifs aims to identify important local structures of networks. Defined by a particular pattern of nodes and edges that is statistically significant, motifs may provide a deep insight into the network’s functional building blocks [20, 27]. For example, motifs are widely used in biological networks for the identification of functional DNA sequences and gene regulatory patterns [6, 19]. Despite the efforts of scholars from computer science and bioinformatics, motif discovery remains a computationally challenging problem, given its combinatorial nature [4, 9].

Most motifs discovery algorithms try to enumerate all sub-graph structures in the network and therefore suffer from the exponential complexity as the motif size grows [32]. An alternative approach based on user queries provides more focused and efficient solutions [7, 22]. In the context of heterogeneous information networks, motif queries can be further specified with node and edge types, as well as searching meta paths to enable faster and more precise local detections. For example, many users reviewing one commodity on an online shopping website may be a common pattern. But if many users with similar IDs post the same content in a short period, these users may be of particular interest (e.g., Internet bots [1]). The search can be further narrow down by imposing constraints on commodity and user types.

Traditional motif discovery methods rely on statistical null models to find special motifs among the sub-structures and identify outliers [19, 25]. Here we propose a motif similarity measure based on their meta path connections, which can also be specified in a user’s query. For example, in a query of an academic collaboration network, we start with a motif consists of two authors and their co-authored physics paper. If there exist a motif contains two authors and a paper in a discipline that significantly differs from physics (e.g., social science, arts, etc.), it is unlikely that the two motifs will be strongly connected by the query meta path set (author-paper-author and paper-author-paper), especially compared with other multi-authored physics papers. Since the query contains both types of constrained motifs and meta paths, our algorithm can be further optimized for motif discovery and ranking at the same time. Outlier motifs refer to those motifs lying in the searching results but with least similarities comparing to the query motif. For example, we are aiming to find two authors coauthored one paper in physics, and the searching results contain authors coauthored mathematics, chemistry, and art. Then the motif with art may be regarded as an outlier motif because others are all theoretical science.

The high-level procedure of motif discovery is shown in Figure 2. First of all, users should first define one or more target motifs to start the query. These target motifs are regarded as ”motif reference” when comparing sub-graphs. Then we use meta paths (also specified by the users) to discover sub-graphs that are related to the target motifs. These discovered sub-graphs are ”candidate motifs”. After comparing the similarities between candidates and references, the candidate motifs with lower similarities are regarded as outliers or the final output motifs. All of the formal definitions are shown in Section 2.1.

We propose the similarity measure called MOS (Motif Outlier Score) to evaluate the similarity between different candidate motifs. According to user’s queries, guided by meta paths, we can efficiently calculate MOS in the candidate motif set based on the statistics gathered during the sub-graph matching process [30]. Finally, the interesting motifs can be recognized according to the ranked scores list. Original contributions in this paper are outlined as follows.

- We propose an efficient and flexible motif discovery algorithm by taking advantage of sub-graph and meta path queries (i.e., human behaviour), leading to an intelligent motif searching framework that is applicable to a wide range of social computing applications.
- The target motif similarity is defined based on node similarities. Interesting human behavior patterns have been discovered by applying our algorithm to real-world heterogeneous information networks.
- We provide empirical evidence that our framework is robust to the choice of similarity measures, including PathSim, CosSim, and MOS.

## 2 FINDING OUTLIER MOTIFS BASED ON QUERIES

Herein, we introduce how to find outlier motifs based on users’ queries. We firstly give some related definitions. Then we introduce our proposed outlier detection method. To clearly illustrate our method, we explain it in four steps. First, we achieve motifs based on queries. Second, we count the meta paths. Then, we calculate the similarity of node pairs. Finally, we order the MOS of each motif. When we complete the four steps, we can find the outlier motifs based on queries.

As shown in Figure 2, the User’s query contains concepts like target motif and search paths and we use an instance to explain them. If an administer wants to find unusual structures of “two friends that like the same genre of music” around friends of two friends like rock on a streaming media platform. We called entities of the triplet as motif (i.e., Alice and Bob like rock, Alice-Bob-rock is a motif), and target motif is an abstract concept that constrain node types and structures of found motifs. We find their friends by the “user-user” type paths. Besides, we have to define a standard to evaluate the strange level of these motifs, if the administer wants to compare the motifs around the start motif using motifs generated by another motif (i.e., Cindy and David like blues). The motifs used for comparison form the reference set. We count the number of “user-genre-user” paths each motif in the candidate set with all motifs in the reference set. If a motif has fewer paths connecting with the reference set, it is more likely to be an outlier.
2.1 Motifs, Meta path, and Similarity

Motifs are generally considered as building blocks in various kinds of networks, including information networks, transportation networks, social networks, and so on [27]. There have been various studies verifying that motifs occupy important positions in networks. Network motifs refer to the small structures frequently appearing in the information network [16]. Herein, we formally give the definitions of information network, network motif, meta path, and other related concepts.

**Definition 1.** Information network: An information network can be written as a graph $G(V, E, \varphi)$. $V$ is the node set of the graph. $E$ is the edge set of the graph. $\varphi$ is a mapping function $\varphi : V \rightarrow A$, wherein $A$ is the set of node types. In some cases, the mapping function may be $\psi : E \rightarrow R$, where $R$ is the set of edge types. Apparently, the information network $G(V, E, \varphi)$ is homogeneous when $|A| = 1$. If $|A| > 1$, then the information network $G(V, E, \varphi)$ is heterogeneous.

The formal definition of motifs is given in Definition 2.

**Definition 2.** Network motif: The network motif refers to a sub-network structure $M \in G$, wherein $M$ appears in the network for $k > \beta (\beta > 0)$ times. Generally, $\beta$ refers to the times that $M$ appears in $G$’s corresponding random network $\tilde{G}$.

Motifs appear in a triangle structure in many situations. Some higher-order motifs are also significant but always cause higher computational complexity, especially in large-scale networks [15]. Properties of nodes and motifs in an information network can be measured in many ways. It is difficult to judge whether a certain motif is an outlier or not without any restrictions. Therefore, in this study, we constrain outlier motifs in the range of a motif set. With the motif set, the scope of finding outlier motifs can be settled. An outlier motif set is generated based on the query motif and the meta path. Herein, the meta path refers to a path connected by different types of nodes. Since there only exists one node type in homogeneous networks, meta path only exists in heterogeneous networks. The formal definition of a meta path is given as follows.

**Definition 3.** Meta path: $A$ is the node type set. A meta path is a directed path with fixed length and node types in the path. An example of a meta path can be denoted as $A_1 \rightarrow A_2 \rightarrow \ldots \rightarrow A_n$. Meta paths in heterogeneous information networks always reflect certain practical meanings. Take the meta path “user-genre-user” in the music community network as an example, this meta path can be used to describe the users who are interested in the
same type of music. Different meta paths represent different practical meanings. If the sequence of node types in a meta path is centrally symmetric, then this meta path is denoted as a symmetric meta path.

Meta path, especially symmetric meta path, can also be applied to measure similarities between different nodes. Generally, two nodes are recognized as similar if they have many meta paths. There are many ways to describe the similarity between two nodes, such as value, topology, etc. Herein, we introduce three kinds of node similarity evaluation metrics, including PathSim [29], CosSim, and Normalized Connectivity [13]. These definitions are shown as follows, respectively.

**Definition 4. PathSim:** The PathSim of two nodes in the same type is denoted as

$$S_{PathSim}(x, y) = \frac{2 \times |\{p_{x \rightarrow y} \mid p_{x \rightarrow y} \in P\}|}{|\{p_{x \rightarrow x} \mid p_{x \rightarrow x} \in P\}| + |\{p_{y \rightarrow y} \mid p_{y \rightarrow y} \in P\}|},$$

wherein, $P$ is the set of fixed symmetric meta paths, and $p_{x \rightarrow y}$ is one of the symmetric meta paths between $x$ and $y$.

PathSim is used to measure the similarity between two nodes with symmetric meta paths. Besides PathSim, CosSim (cosine similarity) can also be applied to measure the similarity between two nodes. The formal definition is given as follows.

**Definition 5. CosSim:** The cosine similarity of two nodes $x$ and $y$ is defined as

$$S_{CosSim}(x, y) = \sum_{i \in N_{x,y}} \left( \left| \{p_{i \rightarrow i} \mid p_{i \rightarrow i} \in P\} \right| \times \left| \{p_{y \rightarrow i} \mid p_{y \rightarrow i} \in P\} \right| \right),$$

wherein, $N$ is the node set, which is reachable from node $i$ via meta path $p$.

Moreover, scholars have also proposed normalized connectivity to evaluate the similarity between two nodes. Normalized connectivity is used to measure the similarity between two nodes with a fixed meta path, as shown in Definition 6.

**Definition 6. The normalized connectivity of two objects in the same type is defined as**

$$S_{NormCon}(x, y) = \frac{\left| \{p_{x \rightarrow y} \mid p_{x \rightarrow y} \in P\} \right|}{\left| \{p_{x \rightarrow x} \mid p_{x \rightarrow x} \in P\} \right|},$$

wherein, $P$ is the set of fixed symmetric meta paths, and $p_{x \rightarrow y}$ is one of the symmetric meta paths between $x$ and $y$.

### 2.2 Problem Definition

It should be noted that although our algorithm is used for motifs, it can be applied to sub-graphs in different sizes. In this section, we use a sub-graph to represent small graph structures. In other sections, we mainly use motifs.

For a given heterogeneous information network $G = (V, E, \phi)$ and a user’s query sub-graph $M_u = (V_M, E_M, \phi_M)$, find the candidate sub-graph set $CMS = M_1, M_2, \ldots, M_n$ based on the searching constraint condition SCC. Wherein, SCC contains the given meta path set $S_M = MP_1, MP_2, \ldots, MP_n$ and the start nodes for searching. The outlier sub-graph detection problem aims to find an outlier sub-graph set $S_{out} = M_1, M_2, \ldots, M_n$ satisfying that the sub-graph $M_i \in S_{out}$ owns the outlier score $MOS(i)$ that is beyond the default range.

Herein, $M_u$ should contain the complete information, including node types as well as the connections between different nodes. SCC contains two parts, i.e., meta paths and starting nodes. Meta paths determine what the searching routines are, and the starting nodes determine where the search is originated.

### 2.3 Outlier Sub-graph

The measures mentioned above are used to evaluate the similarity between nodes. However, the similarity of sub-graphs still needs to be studied. Herein, we define MOS based on the outlier scores of nodes in the sub-graph, as shown in Definition 7. MOS is a function of two sets: the reference set $S_R$ and the candidate set $x$. Sub-graphs in the reference set are used as references. Users give reference set as one of the query constraints. It can also be generated based on the query sub-graph. Sub-graphs in the candidate set are those that need to be compared with sub-graphs in the reference set. Therefore, MOS reflects the outlier degree of sub-graphs comparing with the reference set.

**Definition 7. MOS:** A value to represent the similarity between sub-graphs, which is defined as

$$\Omega(x, S_R) = \sum_{y \in S_R} s(x, y),$$

wherein, $S_R$ is the reference set and $s$ is a well-defined similarity.

**Definition 8. Outlier sub-graph:** Suppose $M_u$ is a given sub-graph and $M = \{M_1, M_2, \ldots, M_n\}$ is the sub-graph set generated based on $M_u$. A sub-graph $M_o$ is regarded as an outlier sub-graph in the information network $G(V, E, \phi)$, which states that $M_o$ with the value of MOS is an anomaly.

### 2.4 Outlier Sub-graphs Detection

The overall process of outlier sub-graph detection is shown in Algorithm 1. Based on the user’s query sub-graph, we first preprocess the data and set the search conditions. Then we generate the candidate sub-graph set and the reference sub-graph set, which are determined by the user’s query sub-graph and meta path. Next, we calculate the MOS values of the sub-graphs in the candidate set. Finally, we order the sub-graphs according to their MOS values.

**Algorithm 1 Outlier Sub-graphs Detection**

**Require:** Target sub-graph $M$, searching constraint condition SCC, meta path set $MP$; reference sub-graph set $CMS$, candidate sub-graph set $CMS$

**Ensure:** An ordered list of candidate sub-graphs $NetOutlist$

1. for meta path $\in MP$ do CMS.add(search results);
2. end for
3. if SCC = NULL then RMS = CMS
4. else
5. for meta path $\in MP$ do RMS.add(search results);
6. end for
7. end if
8. for sub-graph $m$ in CMS do NetOutlist.append()
9. end for
10. sort(MOSlist);
2.4.1 Generating Requirements Based on Queries. User query is particularly important that affects the performance of outlier sub-graphs detection. Generally, a detailed query leads to better detection results. In this work, the user query contains two mandatory parts, including at least one sub-graph and a set of meta paths. A sub-graph is judged to be an outlier sub-graph when its MOS value is beyond the reasonable range. As we have illustrated above, the MOS value is calculated based on the candidate set and reference set. Hence the reference set is vital in the whole outlier sub-graph detection process. The reference set can be provided by the user, or it can be generated based on one or more sub-graphs and meta paths given by the user.

The generation process of the reference set is the same as that of the candidate set. Figure 3 shows an example of the process. It is a heterogeneous network that consists of authors and papers. Wherein, the authors are labeled by “(D)-” and the papers are labeled by “(A)”. In Figure 3, the start sub-graph is “<C>-<D>-<B>” and the type of meta path is given as “author-paper-author”. Then from the start sub-graph, we can find two sub-graphs guide by 5 different meta paths, “<C>-<B>-<C>, <C>-<B>-<D>, <D>-<D>-<B>, <D>-<E> and <D>-<D>-<A>.” The number labels each path. It can be seen that the 5 meta paths lead us to find three sub-graphs, i.e., “<C>-<E>-<E>-<A>-<B>-<A>-<D>-<B>, and <C>-<D>-<B>-<A>. Apparently, “<C>-<D>-<B>-<A>” is the start sub-graph itself. Besides, meta path 4 and 5 both lead to the same sub-graph “<A>-<B>-<B>-<B>-<A>”.

With the search conditions of a query given in advance, it is easy to find out the nodes of specified classes. However, there exists a significant problem due to the ordered search. That is, how to avoid the same sub-graph appears many times in the result set?

For every node in a network, we keep the neighbor information classified by node type. We traverse a node by checking whether it is in the map from node instance to positions of meta sub-graph. The first problem can be solved by Depth-First Search (DFS). When generating possible sub-graphs, add the sub-graph into the result set if it (the sub-graph or its isomorphisms) is not found.

2.4.2 Enumerating Meta Paths. This procedure is to enumerate the number of meta paths between certain pairs of sub-graphs. As shown in Figure 3, there exist two meta paths between sub-graphs “<A>-<B>-<A>” and “<A>-<B>-<B>-<A>”. In real-world networks, there can be multiple meta paths with a much more complicated situation. To avoid repetitive sub-graph detection, we discover a way to effectively enumerate the meta paths between sub-graphs. That is to enumerate the meta paths between nodes.

Specifically, to get the number of meta paths between any candidate sub-graph and reference set, we first construct a reference node set and a candidate node set, respectively. It should be noted that there could be multiple meta paths used in enumeration, which means that all categories that appeared in the meta sub-graph can be the start node of a meta path. They can also be assigned with different weights. The default weight of each path is 1.

The two nodes connected by a symmetric meta path are generally share higher similarities. This is because a symmetric meta path always connects two nodes of the same type. In this work, we use the number of symmetric meta paths to evaluate the similarity between nodes. For two nodes in a network, the more symmetric meta paths they have, the more similar they are that viewed from different nodes. For two sub-graphs in the network, more symmetric meta paths between nodes at the same positions in sub-graphs, we consider that sub-graphs are more similar in the corresponding positions. If all the nodes at the same positions in two sub-graphs are similar, then we can assume the two sub-graphs are similar. For a given meta path MP, the symmetric meta path of MP is written as $MP_{sym} = MP \circ MP^{-1}$. The detailed procedures can be seen in Algorithm 2. We find the reachable node set for each node in the meta path, and enumerate the number of paths to the reachable node set. After updating the Lastlayer set, Currentlayer is cleared up.

Algorithm 2 Reachable Nodes Detection

Require: Candidate sub-graph set CMS, reference sub-graph set RMS, meta path set MP
Ensure: Attribute node sets $N2NC$, $N2NR$

1: function GetReachableNodes(Sub - graphSet)
2:     $N2N = new$ dictionary;
3:     for category $c$ in Sub do
4:         initialize Lastlayer;
5:         initialize Currentlayer;
6:         for node $n$ in each sub-graph in MS do
7:             if $n$ in $N2N[c]$ then
8:                 continue
9:             end if
10:            Lastlayer[$n$] = 1;
11:            for node type $t$ in $mp$ do
12:                find the reachable node set from node $n$;
13:                update Currentlayer with paths numbers;
14:                Lastlayer = Currentlayer;
15:                initialize Currentlayer;
16:            end for
17:            $N2N[c][n] = Lastlayer$;
18:        end for
19:     end function

20: $N2NC$ = GetReachableNodes(CMS)
21: $N2NR$ = GetReachableNodes(RMS)
22: return $N2NC; N2NR$

Herein, we propose a novel search procedure called bidirectional search. Generally, the default search process is a directed search procedure. However, for a symmetric meta path we employed in this work, we use bidirectional search to reduce the complexity. We implement our search process beginning from both the candidate node set and the reference node set. Since the search process is bidirectional, we only search for the half-length of the meta path. This can ensure that the ending nodes are with the same node type.

2.4.3 Calculating Similarity. We have introduced three ways to calculate the similarity between two nodes. However, each of them cannot be applied to calculate the sub-graphs similarity directly.
The time complexity of the first step (1) finding the candidate set of sub-graphs is $O((n/c)^{(l_1+1) × (n/c)^2}) = O(n^{l_1+2})$. In the second step (2), assume the size of the candidate set is $m$ and the length of half of the given symmetric path is $l_2$. We use dictionaries to save found nodes and number of paths to them, and the complexity of counting edges is $O((n/c) × (n/c)^{(l-1)/2} + m^2) = O(n^{l_2} + m^2)$. We

---

**Algorithm 3 Meta Paths Calculation**

**Require:** $N2N_C$, $N2N_R$, node type set $C$

**Ensure:** two maps $A2A$, $A2B$

1. initialize $A2A$, $A2B = $ new set;
2. for node $c$ in node type set $C$ do
3.   for node pair $(a,b)$ in $N2N_C[c]$ do
4.     if $N2N_C[c][a] ∩ N2N_R[c][b] ≠ ∅$ and $(a,b) ∉ A2B$
5.        $A2B[a][b] = 0$;
6.        for $d ∈ N2N_C[c][a] ∩ N2N_R[c][b]$ do
7.           $A2B[a][b] += N2N_C[c][a][d] × N2N_R[c][b][d]$;
8.        end for
9.    end if
10.   end for
11. end for
12. end for
13. if $(a,e) ∉ A2A$ then
14.   $A2A[a][e] = 0$;
15. end if
16. for $d ∈ N2N_C[c][a] ∩ N2N_R[c][e]$ do
17.   $A2A[a][e] += N2N_C[c][a][d] × N2N_R[c][e][d]$;
18. end for
19. end if
20. end for
21. end for
22. return $A2A$, $A2B$
sort sub-graphs according to their MOS in the last step (3), whose complexity is \(O(m \log(m))\).

However, networks, in reality, are often sparse. As a result, we use average node degree to compute the complexity of our method. Assume the average node degree is \(k\), and \(k \ll n\), then the time complexity of the first step (1) is \(O(k^{(l_1+1)} \times k^4) = O(k^{(l_1+5)})\). Assume the nodes of candidate set are \(n_c\), the complexity of the second step (2) is \(O(n_c \times k^{l_2-1} + m^2) = O(n_c \times k^{l_2} + m^2)\). The last step (3), we sort the similarities of sub-graphs in the candidate set, the complexity in this step is \(O(m \log(m))\). We list the two complexity in Table 1.

### Table 1: Comparison of the complexity of two types of networks.

| Complete graph | Average degree=k graph |
|----------------|------------------------|
| \(O(n^{l_1+5-1})\) | \(O(k^{(l_1+5)})\) |
| \(O(n^2 + m^2)\) | \(O(n_c \times k^{l_2} + m^2)\) |
| \(O(m \log(m))\) | \(O(m \log(m))\) |

### 3 INTERESTING OUTLIER MOTIFS IN REAL-WORLD DATA SETS

In this section, to verify the effectiveness of the proposed method, we employ it in a real-world data set. Herein, we mainly apply the method in academic networks.

#### 3.1 Data Set and Experimental Settings

We employ part of data in Aminer\(^1\) to construct a heterogeneous information network. It contains information including index, title, venue, and other terms of paper. However, not all papers have complete information. We extract 2,092,356 papers and 1,571,933 authors.

When we construct this heterogeneous network, we use three node types, including venues, authors, and terms. The node type of "paper" is not involved in the construction of the heterogeneous network. This is because the node type "paper" is the indirect relationship between authors and terms. Terms are extracted from titles of paper, and authors are connected with certain terms when titles of their published papers contain these terms. The types of edges contain "author - author" (co-authorship), "author - term" (via a paper), "term - term" (appearing in the same title), "author - venue" (publication), and "term - venue" (publication).

Interestingly, by the above mentioned preprocessing procedure, nodes with super high degree are generally irrelevant to our target. For example, the venue node with the largest degree is "IEEE Transactions on Information Theory". The degree is 11,227. If we search motifs consisting of this node, the complexity of the search is more than \((\frac{1000}{2})^{5} = 5 \times 10^7\). The term node with the largest degree is "of", which appears for 698,767 times. Although high degree nodes in the network mean high frequency, these nodes are insignificant, especially when we want to distinguish outlier motifs. The common feature of motifs is obvious, but finding out underlying relations without any help is a difficult task. In terms, words with high frequency are some usual words like "of", "in", "the", etc. Therefore, we preprocess them by threshold filtering, and those appear for more than 5,000 times are filtered.

#### 3.1.1 Case 1

We choose some typical queries to analyze experimental results. First, we use "F. Xia (author) - G. Wu (author) - authentication (term)" from "Mobile Networks and Applications" as query motif, "author - author - term" as target motif, the motif set searched by "author - term - author" as candidate set and reference set, ("author - term - author", "term - author - term") as the meta path set for MOS calculation. The result set consists of 11,219 motifs. We show the top 10 motifs, which are most similar to the motif set and least similar to the motif set in Table 2.

### Table 2: The top 10 outlier motifs and the top 10 most similar motifs detected based on the query motif of "F. Xia (author) - G. Wu (author) - authentication (term)."

| Author 1         | Author 2         | Term         | MOS   |
|------------------|------------------|--------------|-------|
| X. Geng          | Y. Huang         | defending   | 58.867|
| X. Geng          | Y. Huang         | against      | 58.867|
| X. Geng          | Y. Huang         | challenge    | 58.867|
| X. Geng          | Y. Huang         | ddos         | 58.867|
| X. Geng          | A. B. Whinston   | defending   | 58.867|
| X. Geng          | A. B. Whinston   | against      | 58.867|
| X. Geng          | A. B. Whinston   | challenge    | 58.867|
| X. Geng          | A. B. Whinston   | ddos         | 58.867|
| Y. Huang         | A. B. Whinston   | defending   | 58.867|
| Y. Huang         | A. B. Whinston   | against      | 58.867|
| Q. Zhang         | K. Farkas        | more         | 1817.762|
| M. Krunz         | M. Z. Siam       | beaconless   | 1848.091|
| M. Krunz         | M. Z. Siam       | geographical | 1848.091|
| Q. Zhang         | J. Liu           | more         | 1855.639|
| C. Politis       | T. Dagiuklas     | extreme      | 1866.679|
| H. Chen          | C. Yeh           | 802.16       | 1868.442|
| H. Chen          | T. Wang          | 802.16       | 1868.442|
| M. Krunz         | M. Z. Siam       | gains        | 1883.091|
| Q. Zhang         | X. Wang          | more         | 1899.95 |
| Y. Chen          | Y. Huang         | advanced     | 1931.042|

Although irrelevant words are removed, we still find some useful information - "802.16" is more relative to "authentication" than "ddos". The more interesting thing is that if the motifs are extracted from the same paper, the MOS of motifs tends to be similar - they are not the same, like a motif appearing in two papers. We give more information in Figure 4.

We divide the motif list into 10 groups according to MOS values. Figure 4(a) shows the distribution of the top 10 outlier terms appearing in each group. Figure 4(b) shows the distribution of the top 10 outlier authors appearing in each group.

To be specific, the different color stands for the MOS value between motifs. The more the color appear in blue, the corresponding...
term is closer to the outlier. In the old wireless network standard, 802.11 is less relative than 802.16. Research in 802.16 is highly associated to authentication. Moreover, "authentication" is used frequently in network security, so we can see terms like privacy are highly related to the motif set. If we take network security as the central semantic of the motif set, it is obvious the MOS represents domains related to the central semantic, while others are components of the network.

3.1.2 Case 2. In this case, the given query venue is "IEEE Transactions on Knowledge and Data Engineering". We use "J. Tang (author) - J. Li (author) - recommendation (term)" as a query motif and "author - author - term" as target motif type. The motif set searched by "author - term - author" is regarded as the candidate set and the reference set. We use a set of "author - term - author", "term - author - term" as the meta path set for MOS calculation. The result consists of 26,782 motifs. We list 20 motifs in Table 3. The 20 motifs include 10 motifs, which are most similar to the motif set and another 10 motifs, which are least similar to the motif set.

The top 10 outliers is almost irrelevant to the central semantics of the motif set. But different from the top 10 outliers, the top 10 similar motifs are not closely related to our theme of the query motif. The distribution of outlier nodes in this experiment are shown in Figure 5(a) and Figure 5(b), respectively.

![Figure 4](image1.png)
(a) Terms
(b) Authors

**Figure 4:** The distribution of nodes in outlier motifs. 4(a) the distribution of "terms". 4(b) the distribution of "authors".

![Figure 5](image2.png)
(a) Terms
(b) Authors

**Figure 5:** The distribution of nodes in outlier motifs. 5(a) the distribution of "term". 5(b) the distribution of "authors".

Most of the words appearing in Figure 5(a) are related to recommendation. However, it seems that recommendation is not related to the motif sets. If we focus on the distribution of motifs containing recommendation in the result, there only exists a few parts in the result. Another surprising conclusion is that the nodes with excessive degree will affect our result since they will generate more motifs. Despite the simple preprocess, the results of the experiment are ambiguous. In the next section, we will use a well-defined data set to verify our method and assess its performance.

### 4 DISCUSSION

Herein, we discuss several critical problems that are crucial to experimental results. First, we show the efficiency of our algorithm under different conditions. Second, we discuss how to choose a candidate set. Then, we discuss how the meta paths affect outlier motif detection. Finally, we discuss the experimental results when using different similarities to detect outlier motifs.

#### 4.1 Efficiency under different conditions

Herein, we conduct several experiments to explore the efficiency of the algorithm under different conditions. We extract 5 different venues and list their basic information in Table 4. The full venue names are Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), Problems of Information Transmission (PIT), Mobile Networks and Applications (MNA), Computing in Science and Engineering (CSE), and IEEE Transactions on Knowledge and Data Engineering (TKDE). Other query conditions are the same as Case 2 in the first part of Section 3. We use "C. Faloutsos - H. Tong - diversified", "V. V. Zyablov - M. Handlery - tailbiting", "F. Xia - G. Wu - authentication", "K. Hinsen - G. K. Thiruvathukal - version" and "J. Tang - J. Li - recommendation" as the start motifs, respectively. The result of the experiment is shown in Fig. 6.
Table 4: The basic information of different venues.

|                  | KDD | PIT | MNA | CSE | TKDE |
|------------------|-----|-----|-----|-----|------|
| published papers | 178 | 447 | 740 | 1,338 | 2,601 |
| edge number      | 8,760 | 16,280 | 34,860 | 38,896 | 91,801 |
| author number    | 535 | 381 | 2,002 | 2,335 | 4,999 |
| term number      | 721 | 1,224 | 1,710 | 2,962 | 3,943 |

Figure 6: The computational time and max MOS under different network scales.

It can be seen that the computational time grows with the increase of edge numbers. Meanwhile, we also randomly choose 5 motifs as the start motifs in each network. Then we compute the average running time with different lengths of symmetric paths. The result is shown in Figure 7, wherein there exist five groups. Each group contains five bars, and each bar corresponds to a meta path with a certain length. The line shows the average computational time of each group. All of metapaths start with “author” and end with “author”. That is, a metapath with length of 3 is “author-term-author” and with length of 5 is “author-term-author-term-author”, and so on. Interestingly, with the increasing length of meta paths, the computational time grows with a slight ascending trend. This indicates that the computational time is robust to the length of meta paths. Combining with the basic information of networks, the computational time grows with the increase of average network degree. Variations in computational times among different meta paths are observed, which is due to different query conditions: the more specific the query conditions are, the less time the algorithm consumes. A fuzzy query condition will lead to more searching results in the candidate set, which will lead to longer computational time. In general, the computational time of the proposed algorithm mainly depends on the average network degree and the query conditions.

Since different scales of networks will have an impact on computational time, we specifically discuss the relationships between node degree distribution and computational time. The statistics of node degree distribution is shown in Figure 8. It can be seen that the node degree distribution influences on the computational time. When the networks are in smaller scales (KDD, PIT, MNA), degree distribution has little impact on computational time. CSE and TKDE are in larger scales, while the computational time of CSE is much less than that of TKDE. This is because the author node degree in CSE network is much less, meanwhile the peak node of term node degree appears earlier than that of TKDE.

4.2 Candidate Set

In this work, we generate the candidate set based on the reference set and the given meta path set. The motifs in the candidate set are relevant to the query motif and the motifs in the reference set. The outlier motifs refer to the irrelevant ones according to ground truth that appear relevant in the query results. The generation of the candidate set can collect query-relevant motifs because we use symmetric meta paths. The symmetric meta path we employed can reflect the similarity between the two connected nodes. Therefore, the motifs we generated in the candidate set are generally relevant to the query motif or the motifs in the reference set.

The number of meta paths can reflect the irrelevant relationships between outlier motifs and the query motif or the motifs in the reference set. If two motifs own a large number of meta paths, then the two motifs are of closer properties as well as structural similarities. Therefore, the outlier motifs are generally irrelevant to the query motif or the motifs in the reference set.

4.3 Meta Path

Herein, we discuss the validness of the meta path. Although we can use different meta paths for calculating MOS, the important problem is which type of paths should we use to get a better performance? In heterogeneous information networks, different meta paths have different semantic meanings, like “author - paper - author” means co-authorship between two authors. As a result, they will show different properties. We present the different paths for computing, which leads to different distributions in the result. Moreover, it also illustrates that more meta paths will get the exact relative order. But we cannot use excessive meta paths in the set for computing. In real life, different meta paths have different semantics. According to the meaning we are concerned about, we use different meta paths to find motifs and calculate the similarity between motifs. Herein, we implement a contrast experiment, and the experimental results are shown in Figure 9. We also use “F. Xia-G. Wu-authentication” as a
start motif. Meanwhile, we only use one meta path (i.e., “author-term-author”) to detect outlier motifs. We achieve totally different detecting results. Comparing with two meta paths (i.e., “author-term-author” and “term-author-term”), one meta path gains more fuzzy searching results. The reason behind may be that the searching results are generally influenced by a certain node within the start motif when employing only one meta path. Apparently, more meta path can improve the searching accuracy but decline the consuming time in the meantime.

There are also some works on the reliability of meta path [26]. However, the conclusion is not comprehensive for all types of heterogeneous networks. We also need to find a common standard for various heterogeneous networks. On the other hand, an effective method is also vital for finding fitting paths in calculating similarity.

### 4.4 Similarities

We discuss the influences of finding outlier motifs according to different similarities. Herein, we detect the outlier motifs based on PathSim and CosSim, respectively. The results of PathSim are shown in Table 5. Comparing with MOS, we can see that the most similar 10 motifs in Table 5 are completely different from those in Table 2. However, the top 10 outlier motifs are exactly the same, even with the same ranking order. Moreover, each motif listed in the top 10 outlier motifs owns the same similarity distribution. That is, though different ways calculate the similarity values, outlier motifs with the same MOS value are detected in same orders. For example, in Table 5, all of the 10 outlier motifs are with the same PathSim equal to 41.397. In Table 2, these 10 outlier motifs own the same MOS values of 58.867. In Table 6, the CosSim values of these outlier motifs are all 42.038.

CosSim reflects the similarity between two nodes from the perspective of the cosine value between two vector angles in a vector space. Herein, we list the detecting results in Table 6. Comparing with NetOut and PathSim, CosSim yields identical detecting results. There exists another interesting phenomenon that the top 10 outlier motifs have the same similarity values, including MOS, PathSim, and CosSim. It seems that different similarities barely influence on the outlier motif detection results. This phenomenon indicates

| Author 1 | Author 2 | Term      | PathSim |
|---------|---------|-----------|---------|
| X. Geng | Y. Huang| defending | 41.397  |
| X. Geng | Y. Huang| against   | 41.397  |
| X. Geng | Y. Huang| challenge | 41.397  |
| X. Geng | Y. Huang| ddos      | 41.397  |
| X. Geng | A. B. Whinston| defending | 41.397  |
| X. Geng | A. B. Whinston| against | 41.397  |
| X. Geng | A. B. Whinston| challenge | 41.397  |
| X. Geng | A. B. Whinston| ddos | 41.397  |
| Y. Huang | A. B. Whinston| defending | 41.397  |
| Y. Huang | A. B. Whinston| against | 41.397  |
| ...     | ...     | ...       | ...     |
| H. Chen | Y. Zhang | wimax     | 1460.127|
| Y. Zhang | H. Chen | resilient | 1460.299|
| Y. Zhang | H. Chen | optimized | 1469.887|
| V. C. Leung | Y. Zhang | controlled | 1470.980|
| H. Chen | Y. Zhang | inter-carrier | 1471.115|
| V. C. Leung | Y. Zhang | fusion | 1479.168|
| Y. Zhang | V. C. Leung | 802.16 | 1482.189|
| H. Chen | Y. Zhang | qos-aware | 1485.277|
| H. Chen | Y. Zhang | uplink | 1485.788|
| H. Chen | Y. Zhang | 802.16 | 1527.335|

Figure 8: The distribution of node degree in KDD, PIT, MNA, CSE, and TKDE.

Figure 9: The distribution of nodes in outlier motifs. 9(a) the distribution of “term”, 9(b) the distribution of “authors”.
Table 6: The top 10 outlier motifs and top 10 most similar motifs detected based on the query motif "Feng Xia (author) - G. Wu (author) - authentication (term)" and the CosSim similarity.

| Author 1 | Author 2 | Term       | CosSim |
|----------|----------|------------|--------|
| X. Geng  | Y. Huang | defending  | 42.038 |
| X. Geng  | Y. Huang | against    | 42.038 |
| X. Geng  | Y. Huang | challenge  | 42.038 |
| X. Geng  | Y. Huang | ddos       | 42.038 |
| X. Geng  | A. B. Whinston | defending | 42.038 |
| X. Geng  | A. B. Whinston | against  | 42.038 |
| X. Geng  | A. B. Whinston | challenge  | 42.038 |
| X. Geng  | A. B. Whinston | ddos   | 42.038 |
| Y. Huang | A. B. Whinston | defending | 42.038 |
| Y. Huang | A. B. Whinston | against  | 42.038 |
| :       | :        | :         | :     |
| H. Chen  | H. Hu    | ad-hoc     | 1389.960 |
| H. Chen  | C. Wang  | ad-hoc     | 1404.822 |
| H. Chen  | Y. Huang | ad-hoc     | 1411.361 |
| H. Chen  | C. Yeh   | ad-hoc     | 1417.390 |
| H. Chen  | T. Wang  | ad-hoc     | 1421.122 |
| H. Chen  | T. Rasheed | ad-hoc | 1421.122 |
| H. Chen  | D. Meddour | ad-hoc  | 1429.639 |
| S. Mao   | V. C. Leung | ad-hoc   | 1439.300 |
| H. Chen  | Y. Zhang | ad-hoc     | 1496.971 |

Table 7: The top 10 outliers and top 10 most similar to given motif set using PathSim. The beginning motif is "J. Tang (author) - J. Li (author) - recommendation (term)".

| Author 1 | Author 2 | Term       | PathSim |
|----------|----------|------------|---------|
| W. Li    | D. yeung | mild       | 16.358  |
| W. Li    | D. yeung | multiple-instance | 16.358 |
| M. G. Hwang | C. Choi | sense      | 39.134  |
| M. G. Hwang | P. Kim  | sense      | 39.134  |
| C. Choi  | P. Kim   | sense      | 39.134  |
| M. G. Hwang | C. Choi | enrichment | 39.938  |
| M. G. Hwang | P. Kim  | enrichment | 39.938  |
| C. Choi  | P. Kim   | enrichment | 39.938  |
| A. N. Zincir-Heywood | M. I. Heywood | object-orientated | 46.706 |
| A. N. Zincir-Heywood | C. R. Chatwin | object-orientated | 46.706 |

Table 8: The top 10 outliers and top 10 most similar to given motif set using CosSim. The beginning motif is "J. Tang (author) - J. Li (author) - recommendation (term)".

| Author 1 | Author 2 | Term       | CosSim  |
|----------|----------|------------|---------|
| W. Li    | D. yeung | mild       | 18.356  |
| W. Li    | D. yeung | multiple-instance | 18.356 |
| M. G. Hwang | C. Choi | sense      | 40.436  |
| M. G. Hwang | P. Kim  | sense      | 40.436  |
| C. Choi  | P. Kim   | sense      | 40.436  |
| M. G. Hwang | C. Choi | enrichment | 41.740  |
| M. G. Hwang | P. Kim  | enrichment | 41.740  |
| C. Choi  | P. Kim   | enrichment | 41.740  |
| A. N. Zincir-Heywood | M. I. Heywood | object-orientated | 48.951 |
| A. N. Zincir-Heywood | C. R. Chatwin | object-orientated | 48.951 |

that our proposed algorithm performs well in detecting outlier motifs. Furthermore, the proposed algorithm is insensitive to different similarities when detecting outlier motifs.

As for the most similar motifs in our result tables 5 and 6, there exist some differences. None of the top 10 most similar motifs are the same in the three tables. Some of the nodes or edges within the motifs may be the same, but the whole motifs are not. The top 10 most similar motifs are different may be caused by different similarity metrics.

To verify the similarity’s influence on the detection results, we implement another two experiments for Case 2 in an academic network, which starts from motif "J. Tang (author) - J. Li (author) - recommendation (term)". We implement our experiments with PathSim and CosSim, respectively. Table 7 shows the detecting results of PathSim, and Table 8 shows that of CosSim. In Table 7, the list of the top 10 outlier motifs is similar but not the same as the experimental results using MOS. Generally, the top 8 outlier motifs detected based on PathSim are identical as that of MOS. The other 2 outlier motifs are quite different from that in Table 3. For detecting results in Table 8, there also exist the same top 8 outlier motifs.

Similar to Case 1, the similarity values of the top 10 outliers appear with the same ranking orders. Among all the detecting results based on these three similarity metrics, the first top 2 outlier motifs own the same value. This illustrates that the proposed algorithm is a general method that will not be affected by different similarities. In other words, our proposed algorithm can detect outlier motifs with a stable performance. As for the top 10 most similar motifs, few nodes or edges of Table 7 or Table 8 are the same comparing
with MOS. This means that different similarities lead to different detecting results, but the proposed algorithm reduces or eliminates such cases. Therefore, the proposed method can achieve relative stable detecting results.

5 CONCLUSION

In this work, we have examined outlier motifs, an interesting and critical issue in social computing. We have proposed an efficient algorithm for finding outlier motifs in heterogeneous information networks. By exploring the user’s query and constrained conditions (i.e., human behaviour), we calculate MOS of each motif in the candidate motif set and sort the MOS values in the ascending order. We set the standard for MOS in query, which is the structure of a motif similar to a reference motif set. The proposed algorithm contains four steps: obtaining motifs meeting query conditions, counting meta paths between nodes, calculating MOS between a motif in the candidate motif set and the whole reference motif set, and ordering MOS of each motif in the candidate motif set. We verify our algorithm on two information networks from the real-world academic network and discuss the experimental results of both networks. Interesting outlier motifs are also found in these networks. Furthermore, we also discuss the details about how to choose the candidate set, meta path, etc. Our work sheds light on finding interesting outlier motifs in large-scale heterogeneous networks and provides a new way of outlier detection for human behavior patterns.

REFERENCES

[1] Robert St Amant and David L Roberts. 2016. Natural Interaction for Bot Detection. IEEE Internet Computing, 20 (4), 2016, 69–73.
[2] Guilherme O’Campos, Arthur Zimek, Jörg Sander, Ricardo JGB Campello, Barbora Micenková, Erich Schubert, Iria Assent, and Michael E Houle. 2016. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. Data Mining and Knowledge Discovery 30, 4 (2016), 891–927.
[3] Pratik Chattopadhyay, Ligo Wang, and Yap-Feng Tan. 2018. Scenario-based insider threat detection from cyber activities. IEEE Transactions on Computational Social Systems 5, 3 (2018), 660–675.
[4] G. Ciriello and C. Guerra. 2008. A review on models and algorithms for motif discovery in protein–protein interaction networks. Briefings in Functional Genomics 7, 2 (Feb. 2008), 147–156. https://doi.org/10.1093/bfgp/elj015
[5] Rémi Domingues, Maurizio Filippone, Pietro Michiardi, and Jihane Zouaoui. 2018. A comparative evaluation of outlier detection algorithms: Experiments and analyses. Pattern Recognition 74 (2018), 406–421.
[6] Sveny Georgiev, Alan P Boyle, Karthik Jayasurya, Xuan Ding, Sayan Mukherjee, and Uwe Ohler. 2010. Evidence-ranked motif identification. Pattern Recognition 43, 11 (2010), 2348–55.
[7] Joshua A. Grochow and Manolis Kellis. 2007. Network Motif Discovery Using Subgraph Enumeration and Symmetry-Breaking. In Research in Computational Molecular Biology, Terry Speed and Haiyan Huang (Eds.). Vol. 4453. Springer Berlin Heidelberg, Berlin, Heidelberg, 92–106. https://doi.org/10.1007/978-3-540-71681-5_7
[8] Manish Gupta, Jing Gao, Charu Aggarwal, and Jiawei Han. 2014. Outlier detection for temporal data. Synthesis Lectures on Data Mining and Knowledge Discovery 5, 1 (2014), 1–129.
[9] N. Kashtan, S. Itzkovitz, R. Milo, and U. Alon. 2004. Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs. Bioinformatics 20, 11 (July 2004), 1746–1758. https://doi.org/10.1093/bioinformatics/btt163
[10] Ravneet Kaur, Sarjeet Singh, and Harish Kumar. 2019. Authorship Analysis of Network Motif Discovery Using Adjustable Incentive Scheme in Community-Based Socially Aware Networking. IET Communications 19, 1 (2017), 17–37.
[11] Roland Leung, Jie Li, Xiaoyuan Meng, Hong-nian Qiao, S. Y. Cheung, and S. S. Y. Lai. 2012. Efficient subgraph matching on billion node graphs. Proceedings of the VLDB Endowment 5, 11 (2012), 992–1003.
[12] Xiao Huang, Hongzhi Wang, Haixun Wang, Bin Shao, and Jianzhong Li. 2012. Efficient subgraph matching on billion node graphs. Proceedings of the VLDB Endowment 5, 9 (2012), 788–799.
[13] Jasdeep Vaidya and Chris Clifton. 2004. Privacy-preserving outlier detection. In Fourth IEEE International Conference on Data Mining. 233–240.
[14] Sebastian Wernicke and Florian Rasche. 2006. FANMOD: a tool for fast network motif detection. Bioinformatics 22, 9 (2006), 1152–1153.
[15] Shuo Yu, Feng Xia, Kamyar Zhang, Zhaolong Ning, Jesusfery Zhou, and Chengfei Zhang. 2017. Team Recognition in Big Scholarly Data: Exploring Collaboration Intensity. In The 3rd IEEE International Conference on Big Data Intelligence and Computing.
[16] Youjiao Yu, Austin Workman, Jacob J Grassick, Michael A Mooney, and Amanda S Hering. 2018. Space-time outlier identification in a large ground deformation data set. Journal of Quality Technology 50, 4 (2018), 431–445.
[17] Xinfeng Zhang, Su Yang, Yuan Yan Tang, and Weishan Zhang. 2016. A thermodynamics-inspired feature for anomaly detection on crowd motions in surveillance videos. Multimedia Tools and Applications 75, 14 (2016), 8799–8826.