Detecting Outlier Patterns with Query-based Artificially Generated Searching Conditions

Shuo Yu, Feng Xia, *Senior Member, IEEE,* Yuchen Sun, Tao Tang, Xiaoran Yan, *Member, IEEE,* and Ivan Lee, *Senior Member, IEEE*

***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.**

***Index Terms*—human behaviour, outlier detection, social computing, heterogeneous network, motif**

I. 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 [1], [2], [3]. 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 [4].

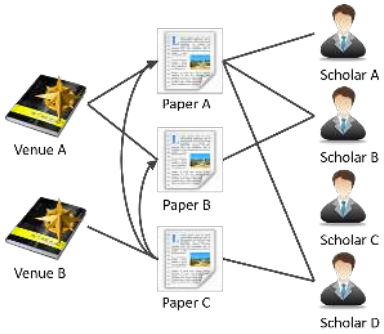


Fig. 1: An example of a bibliographic network.

In the recent past, lots of effort has been placed into the study of heterogeneous information networks and various relationships among entities. Finding unexpected events and detecting outliers from normal patterns have always been an interesting topic in various disciplines [5], [6]. 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 [7], medical diagnosis [8], social network analysis [9], fake news identification [10], public security monitoring [11] and national security [12]. 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 [13], [14], [15]. Various survey papers summarize the existing outlier detection algorithms from abundant perspectives [16], [17], [18].

Beyond node or pairs, outlier detection based on subgraphs 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 [19], [20]. For example, motifs are widely used in biological networks for the identification of functional DNA sequences and gene regulatory patterns [21], [22]. Despite the efforts of scholars from computer science and bioinformatics, motif discovery remains a computationally challenging problem, given its combinatorial nature [23], [24].

Most motifs discovery algorithms try to enumerate all subgraph structures in the network and therefore suffer from

Manuscript received October 30, 2019; date of current version January 28, 2020. *(Corresponding author: Feng Xia.)*

S. Yu, Y. Sun and T. Tang are with Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, School of Software, Dalian University of Technology, Dalian 116620, China. (e-mail: y_shuo@outlook.com; aircleven@outlook.com; tau.tang@outlook.com)

F. Xia is with School of Science, Engineering and Information Technology, Federation University Australia, Ballarat, VIC 3353, Australia. (e-mail: f.xia@ieee.org)

X. Yan is with Network Science Institute, Indiana University, 107 S. Indiana Avenue, Bloomington, IN 47405-7000. (e-mail: yan30@iu.edu)

I. Lee is with School of Information Technology and Mathematical Sciences, University of South Australia, Australia. (e-mail: i-van.lee@unisa.edu.au)

the exponential complexity as the motif size grows [25]. An alternative approach based on user queries provides more focused and efficient solutions [26], [27]. 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 [28]). 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 [22], [29]. 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.

II. 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.

A. 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 [20]. 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 [31]. 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. φ is a mapping function $\varphi : V \to A$, wherein A is the set of node types.*

In some cases, the mapping function may be $\psi : E \to R$, 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.

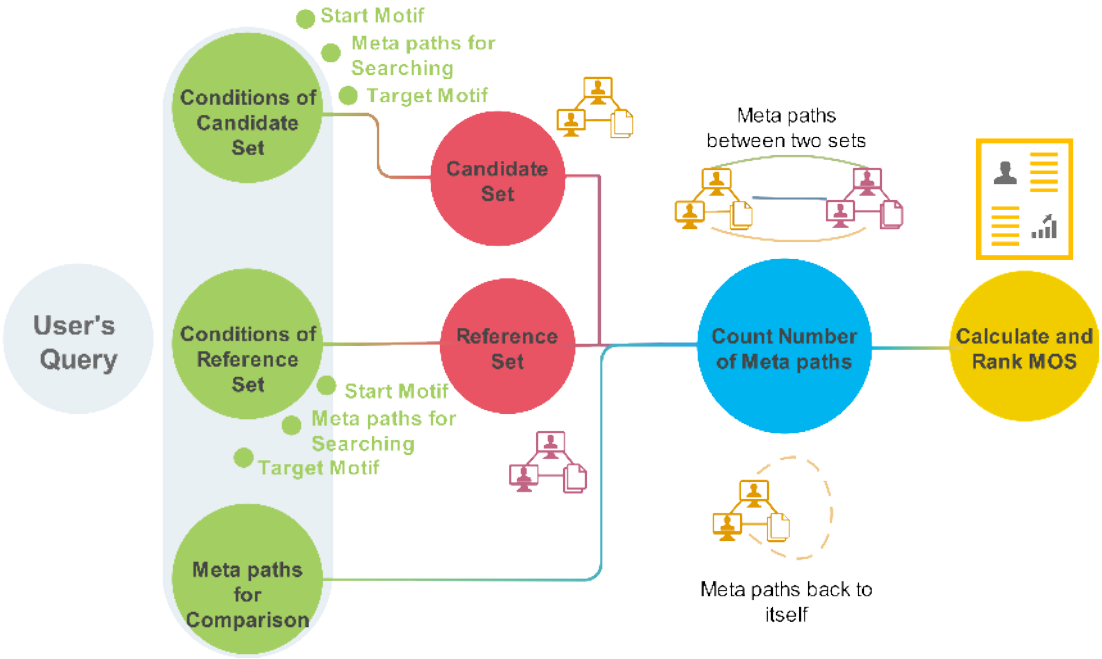


Fig. 2: The framework of finding interesting outlier motifs based on queries.

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

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 [32]. 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 $A_1 \rightarrow A_2 \rightarrow ... \rightarrow A_n$.*

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 [33], CosSim, and Normalized Connectivity [34]. These definitions are shown as follows, respectively.

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

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

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 fixed 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*

$$S_{CosSim}(x,y) = \frac{\sum_{i \in N_{xy}} (|p_{x \rightarrow i}| \times |p_{y \rightarrow i}|)}{||p_{x \rightarrow N_x}||_2 \times ||p_{y \rightarrow N_y}||_2},$$

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*

$$S_{NorCon}(x,y) = \frac{|\{p_{x\to y}|p_{x\to y} \in P\}|}{|\{p_{x\to x}|p_{x\to x} \in P\}|}$$

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

B. 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 $G = (V, E, \varphi)$ and a user's query sub-graph $M_u = (V_M, E_M, \varphi_M)$, find the candidate sub-graph set $CMS = M_{c_1}, M_{c_2}, \ldots, M_{c_n}$ based on the searching constraint condition SCC. Wherein, SCC contains the given meta path set $S_{mp} = MP_1, MP_2, \ldots, MP_n$ 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 $M_1 \in S_{out}$ 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.

C. 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 $G(V, E, \varphi)$, which states that M_o with the value of MOS is an anomaly.*

D. 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 RMS, candidate sub-graph set CMS
Ensure: An ordered list of candidate sub-graphs $Netoutlist$

```
for meta path ∈ MP do CMS.add(search results);
end for
if SCC = NULL then RMS = CMS
else
    for meta path ∈ MP do RMS.add(search results);
    end for
end if
for sub-graph m in CMS do Netoutlist.append()
end for
sort(MOSlist);
```

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 "<>" and the papers are labeled by "()". 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>-(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., <F>-<E>-(E), <A>-<B>-(A), and <C>-<D>-(B). Apparently, <C>-<D>-(B) is the start sub-graph itself. Besides, meta path 4 and 5 both lead to the same sub-graph <A>-<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.

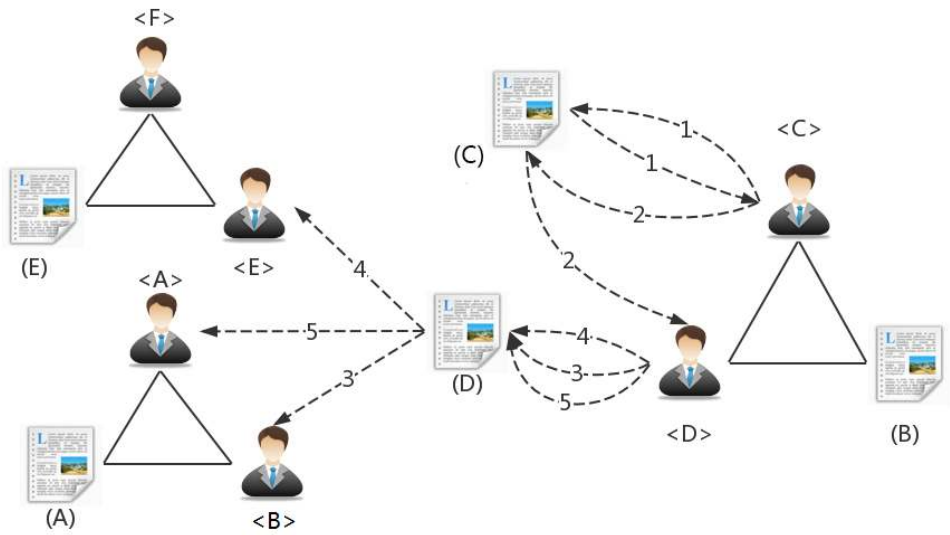


Fig. 3: A reference set and a candidate set generating process.

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) 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 <C>-<D>-(B). 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 $MP_{sym} = MP \bigodot MP^{-1}$. 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.

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.

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. Since a sub-graph may have nodes in the same categories in processing, meta paths may start from a sub-graph and return to itself. To be specific, for node similarity calculation, we only count meta paths from a node to itself. Nevertheless, for sub-graph similarity calculation, some sub-graphs may contain nodes in the same category. It leads to a situation that a symmetric meta path from a node can arrive at other nodes with the same node type. The example shown in Figure 3 has already reflected this kind of situation. Meta paths 1 and 2 both start from the start node, i.e., scholar <C>. However, meta path 1 ends with the start node <C> and meta path 2 ends with scholar <D>. Though <C> and <D> are with the same type, <C> is apparently not what we want to search for.

As a result, the number of meta paths can be represented

Algorithm 2 Reachable Nodes Detection

Require: Candidate sub-graph set CMS, reference sub-graph set RMS, meta path set MP
Ensure: Attribute node sets $N2N_C$, $N2N_R$

function GETREACHABLENODES($Sub - graphSet$)
  $N2N$ = new dictionary;
  for category c in C **do**
    initialize Lastlayer;
    initialize Currentlayer;
    for node n in each sub-graph in MS **do**
      if n in N2N[c] **then**
        continue
      end if
      Lastlayer[n] = 1;
      for node type t in mp **do**
        find the reachable node set from node n;
        update Currentlayer with paths numbers;
        Lastlayer = Currentlayer;
        initialize Currentlayer;
      end for
      $N2N$[c][n] = Lastlayer;
    end for
  end for
  return $N2N$
end function
initialize category set C = new set;
for meta path mp in MP **do**
  C.add(state node category in mp)
end for
$N2N_C$=GetReachableNodes(CMS)
$N2N_R$=GetReachableNodes(RMS)
return $N2N_C$, $N2N_R$;

Algorithm 3 Meta Paths Calculation

Require: $N2N_C$, $N2N_R$, node type set C
Ensure: two maps $A2A, A2B$

initialize $A2A, A2B$ = new set;
for node c in node type set C **do**
  for node pair (a,b) in $N2N_C[c]$ **do**
    if $N2N_C[c][a] \cap N2N_R[c][b] \neq \emptyset$ and $(a,b) \notin A2B$ **then**
      $A2B[a][b] = 0$;
      for $d \in N2N_C[c][a] \cap N2N_R[c][b]$ **do**
        $A2B[a][b] += N2N_C[c][a][d] \times N2N_R[c][b][d]$;
      end for
    end if
  end for
  for node e in $N2N_C[c]$ **do**
    if e in $N2N_C[c]$ **then**
      if $(a,e) \notin A2A$ **then**
        $A2A[a][e] = 0$;
      end if
      for $d \in N2N_C[c][a] \cap N2N_R[c][e]$ **do**
        $A2A[a][e] += N2N_C[c][a][d] \times N2N_C[c][e][d]$;
      end for
    end if
  end for
end for
return $A2A, A2B$

as shown in equation 1

$$|\{p_{m\to m}|p_{m\to m} \in P\}| = \sum_{x\in C} |\{p_{x\to x}\}| + \sum_{x,y\in C} |\{p_{x\to y}\}| \tag{1}$$

in which C is the category of x and y, $x \neq y$.

Before we calculate MOS values of candidate sub-graphs, we have to compute normalized connectivity between nodes. The process for calculating similarities is shown in Algorithm 3.

4) Ordering MOS: In the last step, we calculate MOS of each sub-graph in the candidate set, then we order them according to their MOS values. The result is an ordered list consisting of all sub-graphs in the candidate set. If the MOS of a sub-graph is low, the sub-graph is more likely to be the outlier pattern in the network.

E. Complexity Analysis

The complexity analysis can be divided into three steps. For time complexity, we firstly consider a situation that the graph is complete. There exist c types of nodes, c is a constant. The number of nodes in the graph is n, the number of nodes in each type is n/c. We define the length of search path as l_1 and the size of meta sub-graph as s. The time complexity of the first step (1) finding the candidate set of sub-graphs is $O((n/c)^{(l_1-1)} \times (n/c)^s) = O(n^{(l_1+s-1)})$. 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 $O((n/c)\times(n/c)^{l_2-1}+m^2) = O(n^{l_2}+m^2)$. We 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 and 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 $O(k^{(l_1-1)} \times k^s) = O(k^{(l_1+s)})$. 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 I.

TABLE I: Comparison of the complexity of two types of networks.

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

III. 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.

A. 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 $\binom{10000}{2} = 5 \times 10^7$. 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.

1) Case 1: We choose some typical queries to analyze experimental results. First, we use "Feng Xia (author) - Guowei 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 II.

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.

[1]http://arnetminer.org/lab-datasets/aminerdataset

TABLE II: The top 10 outlier motifs and the top 10 most similar motifs detected based on the query motif of "Feng Xia (author) - Guowei Wu (author) - authentication (term)".

Author 1	Author 2	Term	MOS
Xianjun Geng	Yun Huang	defending	58.867
Xianjun Geng	Yun Huang	against	58.867
Xianjun Geng	Yun Huang	challenge	58.867
Xianjun Geng	Yun Huang	ddos	58.867
Xianjun Geng	Andrew B. Whinston	defending	58.867
Xianjun Geng	Andrew B. Whinston	against	58.867
Xianjun Geng	Andrew B. Whinston	challenge	58.867
Xianjun Geng	Andrew B. Whinston	ddos	58.867
Yun Huang	Andrew B. Whinston	defending	58.867
Yun Huang	Andrew B. Whinston	against	58.867
⋮	⋮	⋮	⋮
Qian Zhang	Károly Farkas	more	1817.762
Marwan Krunz	Mohammad Z. Siam	beaconless	1848.091
Marwan Krunz	Mohammad Z. Siam	geographical	1848.091
Qian Zhang	Jiangchuan Liu	more	1855.639
Christos Politis	Tasos Dagiuklas	extreme	1866.679
Hsiao-Hwa Chen	Ching-Hung Yeh	802.16	1868.442
Hsiao-Hwa Chen	Tzone-I Wang	802.16	1868.442
Marwan Krunz	Mohammad Z. Siam	gains	1883.091
Qian Zhang	Xudong Wang	more	1899.95
Yuh-Shyan Chen	Yueh Min Huang	advanced	1931.042

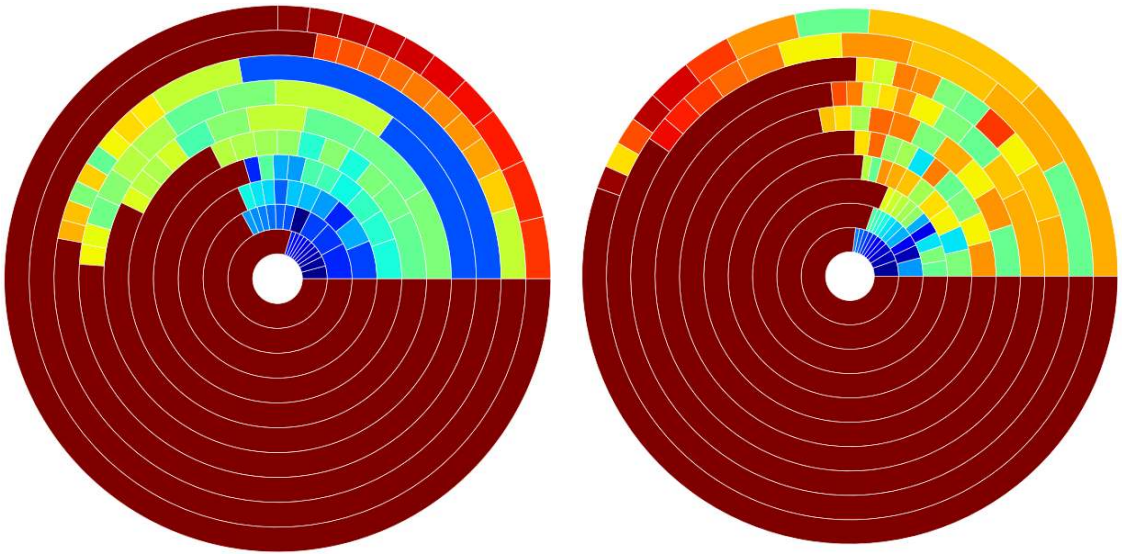
We divide the motif list into 10 groups according to MOS values. Figure 4(a) shows the distribution of the top 10 outlier terms appearing in each group. Figure 4(b) shows the distribution of the top 10 outlier authors appearing in each group.

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.

2) Case 2: In this case, the given query venue is "IEEE Transactions on Knowledge and Data Engineering". We use "Jie Tang (author) - Juanzi 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 III. 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.

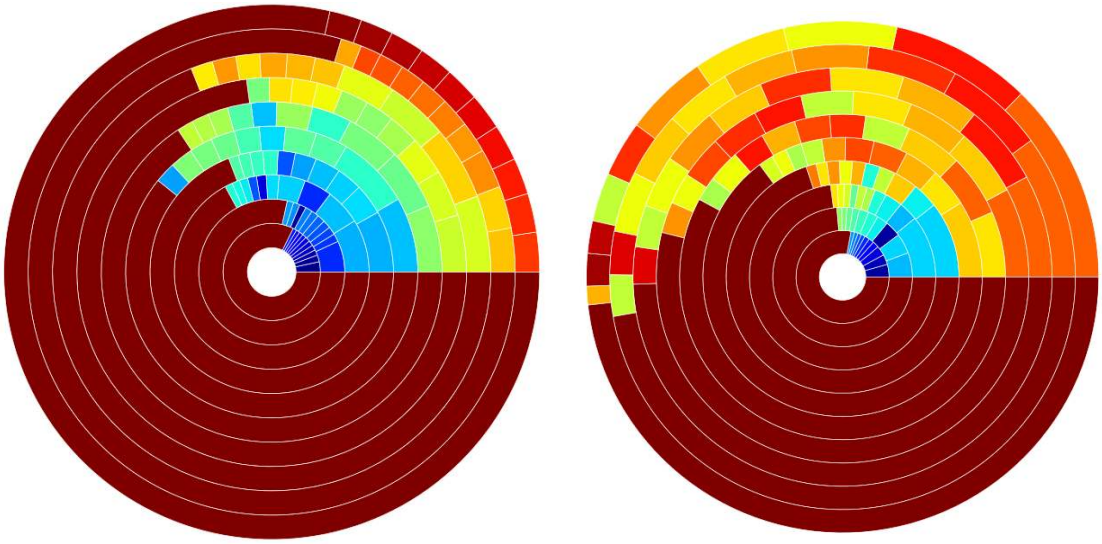
The top 10 outliers is almost irrelevant to the central semantics of the motif set. But different from the top 10 outliers, the top 10 similar motifs are not closely related to our theme of the query motif. The distribution of outlier nodes in this experiment are shown in Figure 5(a) and Figure 5(b),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

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



(a) Terms (b) Authors

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



(a) Terms (b) Authors

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

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.

IV. 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.

A. *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 IV. 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 (M-NA), 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

TABLE III: The top 10 outlier motifs and top 10 most similar motifs detected based on the query motif of "Jie Tang (author) - Juanzi Li (author) - recommendation (term)".

Author 1	Author 2	Term	MOS
Wu-Jun Li	Dit Yan yeung	mild	34.75
Wu-Jun Li	Dit Yan yeung	multiple-instance	34.75
Myunggwon G. Hwang	Chang Choi	sense	63.857
Myunggwon G. Hwang	Pan-Koo Kim	sense	63.857
Chang Choi	Pan-Koo Kim	sense	63.857
Myunggwon G. Hwang	Chang Choi	enrichment	68.538
Myunggwon G. Hwang	Pan-Koo Kim	enrichment	68.538
Chang Choi	Pan-Koo Kim	enrichment	68.538
Zhong-Yong Chen	Chen-Yuan Wu	component	84.429
A. N. Zincir-Heywood	M. I. Heywood	platforms	85.615
⋮	⋮	⋮	⋮
Wenjie Zhang	Ljiljana Brankovic	borda	7306.0625
Ke Yi	Jeffrey Jestes	nutshell	7389.433
Ke Yi	Jeffrey Jestes	concise	7389.433
Ke Yi	Divesh Srivastava	nutshell	7395.054
Ke Yi	Divesh Srivastava	concise	7395.054
Wenjie Zhang	Jianmin Wang	count	7417.208
Xiang Lian	Ke Yi	constant	7436.015
Wenjie Zhang	Jianmin Wang	consensus-based	7726.739
Wenjie Zhang	Jianmin Wang	multivalued	7726.739
Wenjie Zhang	Jianmin Wang	borda	7726.739

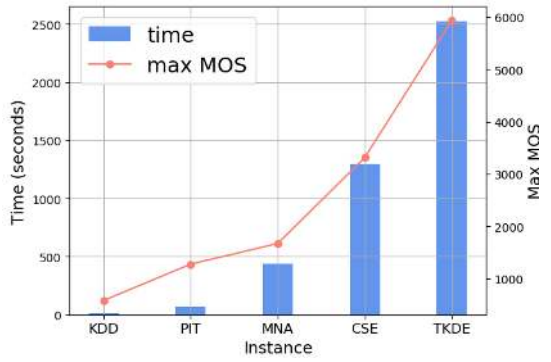


Fig. 6: The computational time and max MOS under different network scales.

part of Section III. We use "Christos Faloutsos - Hanghang Tong - diversified", "V. V. Zyablov - M. Handlery - tailbiting", "Feng Xia - Guowei Wu - authentication", "Konrad Hinsen - George K. Thiruvathukal - version" and "Jie Tang - Juanzi Li - recommendation" as the start motifs, respectively. The result of the experiment is shown in Fig. 6.

TABLE IV: 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

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

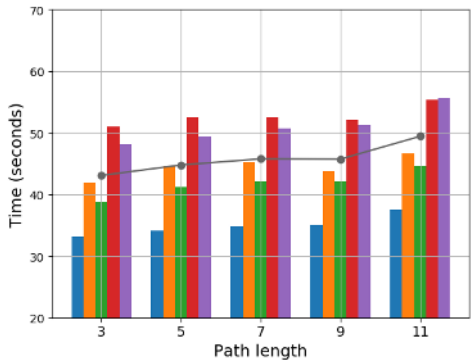


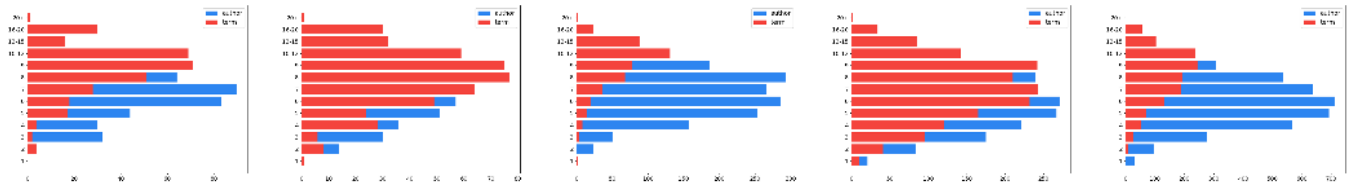
Fig. 7: The time consuming and max MOS with the increasing length of metapaths.

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.

B. 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.



(a) KDD (b) PIT (c) MNA (d) CSE (e) TKDE

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

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.

C. 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 "Feng Xia-Guowei 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 [35]. 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.

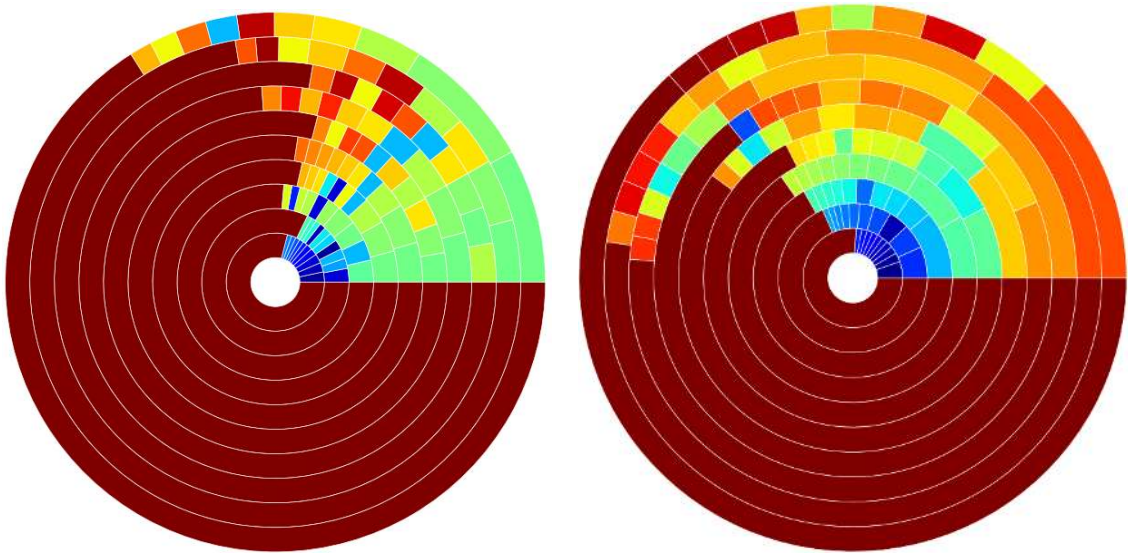
D. Similarities

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 V. Comparing with MOS, we can see that the most similar 10 motifs in Table V are completely different from those in Table II. 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 V, all of the 10 outlier motifs are with the same PathSim equal to 41.397. In Table II, these 10 outlier motifs own the same MOS values of 58.867. In Table VI, the CosSim values of these outlier motifs are all 42.038.

TABLE V: The top 10 outlier motifs and top 10 most similar motifs detected based on the query motif "Feng Xia (author) - Guowei Wu (author) - authentication (term)" and PathSim similarity.

Author 1	Author 2	Term	PathSim
Xianjun Geng	Yun Huang	defending	41.397
Xianjun Geng	Yun Huang	against	41.397
Xianjun Geng	Yun Huang	challenge	41.397
Xianjun Geng	Yun Huang	ddos	41.397
Xianjun Geng	Andrew B. Whinston	defending	41.397
Xianjun Geng	Andrew B. Whinston	against	41.397
Xianjun Geng	Andrew B. Whinston	challenge	41.397
Xianjun Geng	Andrew B. Whinston	ddos	41.397
Yun Huang	Andrew B. Whinston	defending	41.397
Yun Huang	Andrew B. Whinston	against	41.397
⋮	⋮	⋮	⋮
Hsiao-Hwa Chen	Yan Zhang	wimax	1460.127
Yan Zhang	Hsiao-Hwa Chen	resilient	1460.299
Yan Zhang	Hsiao-Hwa Chen	optimized	1469.887
Victor C. Leung	Yan Zhang	controlled	1470.980
Hsiao-Hwa Chen	Yan Zhang	inter-carrier	1471.115
Victor C. Leung	Yan Zhang	fusion	1479.168
Yan Zhang	Victor C. Leung	802.16	1482.189
Hsiao-Hwa Chen	Yan Zhang	qos-aware	1485.277
Hsiao-Hwa Chen	Yan Zhang	uplink	1485.788
Hsiao-Hwa Chen	Yan Zhang	802.16	1527.335

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



(a) Terms (b) Authors

Fig. 9: The distribution of nodes in outlier motifs. 9(a) the distribution of "term". 9(b) the distribution of "authors".

TABLE VI: The top 10 outlier motifs and top 10 most similar motifs detected based on the query motif "Feng Xia (author) - Guowei Wu (author) - authentication (term)" and the CosSim similarity.

Author 1	Author 2	Term	CosSim
Xianjun Geng	Yun Huang	defending	42.038
Xianjun Geng	Yun Huang	against	42.038
Xianjun Geng	Yun Huang	challenge	42.038
Xianjun Geng	Yun Huang	ddos	42.038
Xianjun Geng	Andrew B. Whinston	defending	42.038
Xianjun Geng	Andrew B. Whinston	against	42.038
Xianjun Geng	Andrew B. Whinston	challenge	42.038
Xianjun Geng	Andrew B. Whinston	ddos	42.038
Yun Huang	Andrew B. Whinston	defending	42.038
Yun Huang	Andrew B. Whinston	against	42.038
⋮	⋮	⋮	⋮
Hsiao-Hwa Chen	Honglin Hu	ad-hoc	1389.960
Hsiao-Hwa Chen	Chonggang Wang	ad-hoc	1404.822
Hsiao-Hwa Chen	Yueh-Min Huang	ad-hoc	1411.361
Hsiao-Hwa Chen	Ching-Hung Yeh	ad-hoc	1417.390
Hsiao-Hwa Chen	Tzone-I Wang	ad-hoc	1417.390
Hsiao-Hwa Chen	Keith Conner	ad-hoc	1421.122
Hsiao-Hwa Chen	Tinku Rasheed	ad-hoc	1421.122
Hsiao-Hwa Chen	Djamal-Eddine Meddour	ad-hoc	1429.639
Shiwen Mao	Victor C. Leung	ad-hoc	1439.300
Hsiao-Hwa Chen	Yan Zhang	ad-hoc	1496.971

TABLE VII: The top 10 outliers and top 10 most similar to given motif set using PathSim. The beginning motif is "Jie Tang (author) - Juanzi Li (author) - recommendation (term)".

Author 1	Author 2	Term	PathSim
Wu-Jun Li	Dit Yan yeung	mild	16.358
Wu-Jun Li	Dit Yan yeung	multiple-instance	16.358
Myunggwon G. Hwang	Chang Choi	sense	39.134
Myunggwon G. Hwang	Pan-Koo Kim	sense	39.134
Chang Choi	Pan-Koo Kim	sense	39.134
Myunggwon G. Hwang	Chang Choi	enrichment	39.938
Myunggwon G. Hwang	Pan-Koo Kim	enrichment	39.938
Chang Choi	Pan-Koo 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
⋮	⋮	⋮	⋮
Xuemin Lin	Yufei Tao	multivalued	5651.753
Xuemin Lin	Yufei Tao	borda	5651.753
Yufei Tao	Xuemin Lin	anonymous	5680.435
Yufei Tao	Xuemin Lin	dimensional	5689.031
Yufei Tao	Xuemin Lin	skylines	5690.555
Yufei Tao	Xuemin Lin	threshold-based	5699.910
Yufei Tao	Xuemin Lin	k	5699.910
Yufei Tao	Xuemin Lin	existentially	5700.076
Yufei Tao	Xuemin Lin	extents	5736.802
Yufei Tao	Xuemin Lin	medium	5752.686

similarities when detecting outlier motifs.

As for the most similar motifs in our result tables V and VI, 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 "Jie Tang (author) - Juanzi Li (author) - recommendation (term)". We implement our experiments with PathSim and CosSim, respectively. Table VII shows the detecting results of PathSim, and Table VIII shows that of CosSim. In Table VII, 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 III. For detecting results in Table VIII, 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

TABLE VIII: The top 10 outliers and top 10 most similar to given motif set using CosSim. The beginning motif is "Jie Tang (author) - Juanzi Li (author) - recommendation (term)".

Author 1	Author 2	Term	CosSim
Wu-Jun Li	Dit Yan yeung	mild	18.356
Wu-Jun Li	Dit Yan yeung	multiple-instance	18.356
Myunggwon G. Hwang	Chang Choi	sense	40.436
Myunggwon G. Hwang	Pan-Koo Kim	sense	40.436
Chang Choi	Pan-Koo Kim	sense	40.436
Myunggwon G. Hwang	Chang Choi	enrichment	41.740
Myunggwon G. Hwang	Pan-Koo Kim	enrichment	41.740
Chang Choi	Pan-Koo 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
⋮	⋮	⋮	⋮
Xiang Lian	Lei Chen	cost	4152.757
Xiang Lian	Lei Chen	range	4156.470
Yufei Tao	Xuemin Lin	location-based	4158.846
Yufei Tao	Xuemin Lin	medium	4168.364
Yufei Tao	Xuemin Lin	neighbor	4179.917
Yufei Tao	Xuemin Lin	nearest	4214.662
Xiang Lian	Lei Chen	neighbor	4232.639
Yufei Tao	Xuemin Lin	reverse	4236.367
Xiang Lian	Lei Chen	nearest	4265.022
Lei Chen	Xiang Lian	reverse	4278.532

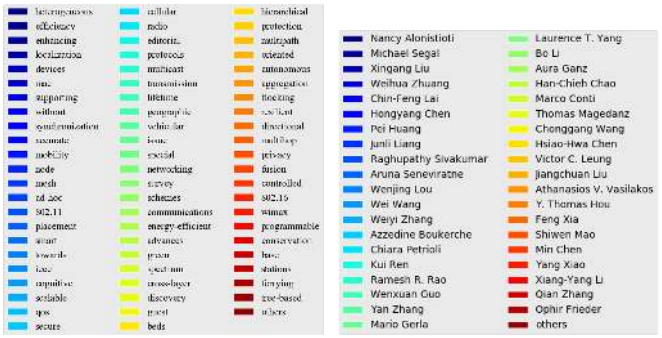
affected by different similarities. In other words, our proposed algorithm can detect outlier motifs with a stable performance. As for the top 10 most similar motifs, few nodes or edges of Table VII or Table VIII are the same comparing with MOS. This means that different similarities lead to different detecting results, but the proposed algorithm reduces or eliminates such cases. Therefore, the proposed method can achieve relative stable detecting results.

V. 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.

Appendix A
Labels for Figures

We list the labels of the figures here for references, see figs. 10 to 12.



(a) Labels for Figure 4(a) (b) Labels for Figure 4(b)

Fig. 10: The labels for Figure 4(a) and Figure 4(b).

(a) Labels for Figure 5(a) (b) Labels for Figure 5(b)

Fig. 11: The labels for Figure 5(a) and Figure 5(b).

References

[1] J. Liu, X. Kong, F. Xia, X. Bai, L. Wang, Q. Qing, and I. Lee, "Artificial intelligence in the 21st century," *IEEE Access*, vol. 6, pp. 34 403–34 421, 2018.

[2] X. Kong, F. Xia, Z. Ning, A. Rahim, Y. Cai, Z. Gao, and J. Ma, "Mobility dataset generation for vehicular social networks based on floating car data," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 3874–3886, 2018.

[3] Z. Ning, L. Liu, F. Xia, B. Jedari, I. Lee, and W. Zhang, "Cais: A copy adjustable incentive scheme in community-based socially aware networking," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 4, pp. 3406–3419, April 2017.

[4] S. Yu, F. Xia, K. Zhang, Z. Ning, J. Zhong, and C. Liu, "Team recognition in big scholarly data: Exploring collaboration intensity," in *The 3rd IEEE International Conference on Big Data Intelligence and Computing*, 2017.

[5] C. Lim and P. P. Maglio, "Data-driven understanding of smart service systems through text mining," *Service Science*, vol. 10, no. 2, pp. 154–180, 2018.

[6] P. Chattopadhyay, L. Wang, and Y.-P. Tan, "Scenario-based insider threat detection from cyber activities," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 3, pp. 660–675, 2018.

(a) Labels for Figure 5(a) (b) Labels for Figure 5(b)

Fig. 12: The labels for Figure 9(a) and Figure 9(b).

[7] R. K. Kovarasan and M. Rajkumar, "An effective intrusion detection system using flawless feature selection, outlier detection and classification," in *Progress in Advanced Computing and Intelligent Engineering*, 2019, pp. 203–213.

[8] P. Savadjiev, J. Chong, A. Dohan, M. Vakalopoulou, C. Reinhold, N. Paragios, and B. Gallix, "Demystification of ai-driven medical image interpretation: past, present and future," *European Radiology*, vol. 29, no. 3, pp. 1616–1624, 2019.

[9] R. Kaur, S. Singh, and H. Kumar, "Authorship analysis of online social media content," in *Proceedings of 2nd International Conference on Communication, Computing and Networking*, 2019, pp. 539–549.

[10] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *ACM SIGKDD Explorations Newsletter*, vol. 19, no. 1, pp. 22–36, 2017.

[11] X. Zhang, S. Yang, Y. Y. Tang, and W. Zhang, "A thermodynamics-inspired feature for anomaly detection on crowd motions in surveillance videos," *Multimedia Tools and Applications*, vol. 75, no. 14, pp. 8799–8826, 2016.

[12] J. Vaidya and C. Clifton, "Privacy-preserving outlier detection," in *Fourth IEEE International Conference on Data Mining*, 2004, pp. 233–240.

[13] M. Gupta, J. Gao, C. Aggarwal, and J. Han, "Outlier detection for temporal data," *Synthesis Lectures on Data Mining and Knowledge Discovery*, vol. 5, no. 1, pp. 1–129, 2014.

[14] M. Landauer, M. Wurzenberger, F. Skopik, G. Settanni, and P. Filzmoser, "Time series analysis: unsupervised anomaly detection beyond outlier detection," in *International Conference on Information Security Practice and Experience*, 2018, pp. 19–36.

[15] G. O. Campos, A. Zimek, J. Sander, R. J. Campello, B. Micenková, E. Schubert, I. Assent, and M. E. Houle, "On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study," *Data Mining and Knowledge Discovery*, vol. 30, no. 4, pp. 891–927, 2016.

[16] G.-J. Qi, C. C. Aggarwal, and T. S. Huang, "On clustering heterogeneous social media objects with outlier links," in *Proceedings of the 5th ACM International Conference on Web Search and Data Mining*, 2012, pp. 553–562.

[17] Y. Yu, A. Workman, J. G. Grasmick, M. A. Mooney, and A. S. Hering, "Space-time outlier identification in a large ground deformation data set," *Journal of Quality Technology*, vol. 50, no. 4, pp. 431–445, 2018.

[18] R. Domingues, M. Filippone, P. Michiardi, and J. Zouaoui, "A comparative evaluation of outlier detection algorithms: Experiments and analyses," *Pattern Recognition*, vol. 74, pp. 406–421, 2018.

[19] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, "Network motifs: simple building blocks of complex networks," *Science*, vol. 298, no. 5594, pp. 824–827, 2002.

[20] C. Shi, Y. Li, J. Zhang, Y. Sun, and P. S. Yu, "A survey of heterogeneous information network analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 1, pp. 17–37, 2017.

[21] S. Georgiev, A. P. Boyle, K. Jayasurya, X. Ding, S. Mukherjee, and U. Ohler, "Evidence-ranked motif identification," *Genome Biology*, vol. 11, no. 2, p. R19, 2010. [Online]. Available: http://genomebiology.biomedcentral.com/articles/10.1186/gb-2010-11-2-r19

[22] A. Masoudi-Nejad, F. Schreiber, and Z. Kashani, "Building blocks of biological networks: a review on major network motif discovery algorithms," *IET Systems Biology*, vol. 6, no. 5, pp. 164–174, Oct. 2012. [Online]. Available: https://digital-library.theiet.org/content/journals/10.1049/iet-syb.2011.0011

[23] N. Kashtan, S. Itzkovitz, R. Milo, and U. Alon, "Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs," *Bioinformatics*, vol. 20, no. 11, pp. 1746–1758, Jul. 2004. [Online]. Available: https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/bth163

[24] G. Ciriello and C. Guerra, "A review on models and algorithms for motif discovery in protein-protein interaction networks," *Briefings in Functional Genomics and Proteomics*, vol. 7, no. 2, pp. 147–156, Feb. 2008. [Online]. Available: https://academic.oup.com/bfg/article-lookup/doi/10.1093/bfgp/eln015

[25] S. Wernicke and F. Rasche, "Fanmod: a tool for fast network motif detection," *Bioinformatics*, vol. 22, no. 9, pp. 1152–1153, 2006.

[26] J. A. Grochow and M. Kellis, "Network Motif Discovery Using Subgraph Enumeration and Symmetry-Breaking," in *Research in Computational Molecular Biology*, T. Speed and H. Huang, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, vol. 4453, pp. 92–106. [Online]. Available: http://link.springer.com/10.1007/978-3-540-71681-5_7

[27] S. Omidi, F. Schreiber, and A. Masoudi-Nejad, "MODA: An efficient algorithm for network motif discovery in biological networks," *Genes & Genetic Systems*, vol. 84, no. 5, pp. 385–395, 2009. [Online]. Available: http://joi.jlc.jst.go.jp/JST.JSTAGE/ggs/84.385?from=CrossRef

[28] R. S. Amant and D. L. Roberts, "Natural interaction for bot detection," *IEEE Internet Computing*, vol. 20, no. 4, pp. 69–73, 2016.

[29] W. E. Schlauch and K. A. Zweig, "Influence of the null-model on motif detection," in *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 2015, pp. 514–519.

[30] Z. Sun, H. Wang, H. Wang, B. Shao, and J. Li, "Efficient subgraph matching on billion node graphs," *Proceedings of the VLDB Endowment*, vol. 5, no. 9, pp. 788–799, 2012.

[31] C. Liang, Y. Li, J. Luo, and Z. Zhang, "A novel motif-discovery algorithm to identify co-regulatory motifs in large transcription factor and microrna co-regulatory networks in human," *Bioinformatics*, vol. 31, no. 14, pp. 2348–55, 2015.

[32] Y. Li, H. U. Leong, L. Y. Man, and Z. Gong, "Quick-motif: An efficient and scalable framework for exact motif discovery," in *IEEE International Conference on Data Engineering*, 2015.

[33] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks," *Proceedings of the VLDB Endowment*, vol. 4, no. 11, pp. 992–1003, 2011.

[34] J. Kuck, H. Zhuang, X. Yan, H. Cam, and J. Han, "Query-based outlier detection in heterogeneous information networks," in *Advances in database technology: proceedings. International Conference on Extending Database Technology*, vol. 2015, 2015, p. 325.

[35] B. Shams and S. Haratizadeh, "Reliable graph-based collaborative ranking," *Information Sciences*, vol. 432, pp. 116–132, 2018.