A Graph Mining Approach for Ranking and Discovering the Interesting Frequent Subgraph Patterns

Saif Ur Rehman¹ · Kexing Liu² · Tariq Ali¹ · Asif Nawaz¹ · Simon James Fong²

Received: 29 January 2021 / Accepted: 12 July 2021
© The Author(s) 2021

Abstract

Graph mining is a well-established research field, and lately it has drawn in considerable research communities. It allows to process, analyze, and discover significant knowledge from graph data. In graph mining, one of the most challenging tasks is frequent subgraph mining (FSM). FSM consists of applying the data mining algorithms to extract interesting, unexpected, and useful graph patterns from the graphs. FSM has been applied to many domains, such as graphical data management and knowledge discovery, social network analysis, bioinformatics, and security. In this context, a large number of techniques have been suggested to deal with the graph data. These techniques can be classed into two primary categories: (i) a priori-based FSM approaches and (ii) pattern growth-based FSM approaches. In both of these categories, an extensive research work is available. However, FSM approaches are facing some challenges, including enormous numbers of frequent subgraph patterns (FSPs); no suitable mechanism for applying ranking at the appropriate level during the discovery process of the FSPs; extraction of repetitive and duplicate FSPs; user involvement in supplying the support threshold value; large number of subgraph candidate generation. Thus, the aim of this research is to make do with the challenges of enormous FSPs, avoid duplicate discovery of FSPs, and use the ranking for such patterns. Therefore, to address these challenges a new FSM framework **A R**anked **F**requent pattern-growth **E**ramework (A-RAFF) is suggested. Consequently, A-RAFF provides an efficacious answer to these challenges through the initiation of a new ranking measure called FSP-Rank. The proposed ranking measure FSP-Rank effectively reduced the duplicate and enormous frequent patterns. The effectiveness of the techniques proposed in this study is validated by extensive experimental analysis using different benchmark and synthetic graph datasets. Our experiments have consistently demonstrated the promising empirical results, thus confirming the superiority and practical feasibility of the proposed FSM framework.

Keywords Graph data · Social network · Graph mining · Transactional graphs · Frequent subgraph patterns · Ranking

1 Introduction

In the era of the connected world, our social and digital lives are confronted with the networks (or simply graphs) on a daily basis [1]. Graph-based representation of real-world problem has been proved to be very beneficial due to their improving simplicity and professional use in finding solutions to the problems [2, 3]. Graphs are generated from almost every field of today's life. Internet browsing means traversing a big network of web pages that is inter-linked via clickable (or sometimes hyper) links [4]. Online social networks such as Facebook are based on massive networks, in which different people are connected through so-called friendship links (a graph of friends) [5, 6]. Further, using mobile accessing one webpage generates a few dozen wired or wireless connections among devices in a matter of

✉ Saif Ur Rehman
saif@uair.edu.pk

Kexing Liu
yb87406@umac.mo

Tariq Ali
tairq.ali@uair.edu.pk

Asif Nawaz
Asif.nawaz@uair.edu.pk

Simon James Fong
ccfong@umac.mo

¹ University Institute of Information Technology, PMAS Arid Agriculture University, Rawalpindi, Pakistan

² University of Macau, Taipa, Macau SAR

microseconds [7]. An example of a real-world graph network is given in Fig. 1.

Keeping in view the significance of graph structure, graph mining is estimated to start since the beginning of 1994 [8]. Since then graph mining has attracted the research community and brought a revolutionary benefit to various fields including computational biology [9, 10], social network analysis [5, 7, 11], chemical data analysis [12, 13], drug discovery [14, 15], and communication networking [16, 17]. Figure 2 shows a typical graph mining process.

Frequent Subgraph Mining (FSM) is a well-studied problem in the graph mining field and boosts several real-world applications such as chemical compound analysis and classification [18, 19], text sentiment analysis, document image clustering [20], bug isolation in software [21, 22], relationship prediction [23], web content mining [24–26], social network mining [8, 27–29], fraud detection [30], email mining [31–33], and anomaly detection [34, 35]. FSM has been a focused theme in graph mining for last two decades; therefore, sufficient literature was dedicated to the field, making tremendous development [35–38]. FSM is classified into two broad classes: (1) transaction-based FSM and (2) single large graph-based.

In most of the studies [39–48], the FSM techniques are decomposed into two major categories: A priori-based and pattern growth-based FSM approaches. In the last few

decades, substantial literature was added to both of these FSM categories. These are included, gSpan [38], FSP [49], FFSM [50], FSG [51], Gaston [54], CloseGraph [52], SPIN [53], SUBDUE [55], TSP [56], FS3 [57], and so forth. For example, the following figure gives an example of FSM. In Fig. 3, a sample graph dataset is given.

If we assumed the support threshold value as 3, then the possible frequent subgraph can be discovered from Fig. 3 as shown in Fig. 4:

The major contributions of this study are fivefold:

- (i) First, the proposal of a new unifying framework **A_R**anked **E**requent pattern-growth **E**ramework (**A-RAFF**) from transaction graph datasets.
- (ii) Second, the proposal of new graph ranking measure, **FSP-Rank**, for addressing the ranking problem of FSPs.
- (iii) Third, comprehensive performance evaluation of the proposed FSM framework, **A-RAFF**, using different graph datasets.
- (iv) Fourth, computing the interestingness of the discovered FSPs.
- (v) Fifth, comparison of the **FSP-Rank** with the **GraphRank**.

The rest of the paper is organized as follows: First, we discussed the related work on FSPs' discovery approaches

Fig. 1 Facebook friends relationship [8]

Fig. 2 A typical graph mining procedure

Fig. 3 A sample graph database

Fig. 4 Sample-mined frequent subgraph patterns

in Sect. 2. Then, we present our novel FSPs ranking framework, A-RAFF, in Sect. 4, where the fundamental principle of the approach is presented. In Sect. 4, we report the experimental results obtained from the simulations of the proposed approach on various graph networks. Finally, in Sect. 5, we draw conclusions with final thoughts for future work.

2 Related Work

In this section, the literature is reviewed, which is directly related to FSM in transaction-based graph datasets. Different representative techniques from the two categories, a priori-based and pattern growth-based FSM approaches, are surveyed, emphasizing on their technique, their contributions

to the FSM domain, and limitations. Research findings are also given at the end of this section.

SPIN (SPanning tree-based maximal graph mINing) was the first approach to mix two distinctive techniques: discovery FSPs in graph dataset and discovery of frequent tree structures in forests (i.e., a collection of trees) for the designing of efficient subgraph mining techniques. SPIN discovers only the maximal frequent subgraphs (MFS) (an MFS is one which is not included in any other FSPs available in the database) of a set of big-size graphs. Huan et al. discussed that extracting the only MFS offered the following advantages in processing big graph datasets [53]: (1) it significantly minimizes the total number of discovered subgraph patterns; (2) various “pruning” schemes can be efficiently combined into the discovery process, which dramatically decrease the time required to mine graph datasets; (3) the non-MFS can be reconstructed from the MFS. SPIN algorithm performance was shown using different experiments, which presented the effectiveness of SPIN in that it exponentially minimizes the number of FSPs discovered. In massive graph datasets, discovered FSPs can become too bigger to allow an entire enumeration by realistic computational resources.

HSIGRAM is another a priori-based algorithm proposed for mining frequent subgraph called HSIGRAM. Adjacency matrix was used to represent the graphs and use iterative merging for the generation of the subgraph. There are two important mechanisms introduced by the HSIGRAM algorithm, the way adopted to enumerate the candidates and the technique used to calculate the frequency of the candidate subgraphs, which considerably influence HSIGRAM overall computational complexity. The further details can be found in [59].

MARGIN is another a priori-based FSG approach to mine the maximal FSPs. The set of maximal frequent subgraphs (MFSs) is significantly smaller than the set of FSPs [53]; therefore, we provided the scope for sufficient pruning of the exponentially massive search space. MARGIN algorithm recursively discovers the candidates by using core procedure, *ExpandCut*, and the set of maximal FSPs was then extracted by the operation of post-processing. Although the MARGIN was found computationally faster when compared with gSpan technique using benchmark graph datasets, its efficiency totally relied on the initial chosen cut and is the flaw found in the analysis of MARGIN [60].

To handle the drawbacks in the existing complete or heuristic frequent subgraph discovery approaches, GREW was proposed [59]. GREW is specially designed and developed to execute on a large graph dataset and to discover patterns corresponding to connect subgraphs that have a large number of nodes-disjoint embeddings. There are two versions of GREW: GREW-single-edge collapsing (GREW-SE) and GREW-multi-edge collapsing (GREW-ME). It can generally operate effectively on very large graphs. The graph is

represented using a sparse graph. Four different data sets were used for performance evaluation of GREW. Dynamic GREW extended FSM algorithms for time series of graphs, particularly dynamic GREW focused on the subgraph patterns which are topologically frequent within a big graph. Different experiments were performed to check the practical feasibility of the dynamic GREW; results on different real-world data validated the proposed technique for the dynamic graph scenarios. The limitation of this technique is an extra overhead in identifying dynamic patterns.

In **SEuS (Structure Extraction using Summaries)**, the authors explored the challenge of frequent structure discovery in the semi-structured data (represented using labeled directed graphs) [61]. It is a three-step procedure: in the first step (summarization), SEuS pre-processes the given graph dataset in order to output a crisp summary. The computed summary is similar to data guides and other approximate typing mechanisms for semi-structured data [62–64]. Candidate generation is the second phase of SEuS. SEuS involved an expert to iteratively look for frequently occurring structures and improve the threshold value. Although the results obtained are approximate in this phase, they are accurate enough to allow tedious structures to be filtered out. When the expert filtered potential structures based on the approximate results of the search phase, an accurate count of the number of occurrences of each possible structure is generated by the third phase, which is the counting phase. SEuS can be used with large connected graph dataset as well as graph transnational datasets.

The **gSpan (graph-based Substructure pattern)** gSpan uses DFS searching strategy and is the fastest FSM approach [38]. DFS code is used for canonical representation of the graph dataset, while the vertex of the graph is traversed using DFS. gSpan restricts the refinement generation in two different schemes. First, fragments can only be expanded at graph nodes, which lie on the rightmost path of the DFS tree. Secondly, fragment generation is allowed by the occurrence in the appearance lists. gSpan avoids duplicate candidate generation by using a canonical code (DFS code) and rightmost path extensions. Although gSpan showed its performance using different real-world graph datasets, gSpan generated a large number of FSPs which become difficult to analyze [40, 48, 55].

Earlier FSM approaches worked to discover all the possible FSPs, which resulted in a large number of FSPs. Such huge number of FSPs were difficult to explore further and were resource intensive as well. Therefore, the concept of CloseGraph was introduced [52]. A close graph is defined by “A given input graph, g , is said to be a closed graph in a given graph database such that there exists no proper super graph of this input graph g with the same support in the database as that of graph g ”. CloseGraph is an enhancement of gSpan approach [52]. Two major concepts involved

in CloseGraph approach are: first is the rightmost extensions and DFS lexicographic order of the FSP generated, and the other concept is an equivalent occurrence and early termination for those graphs which are not closed graphs, so that such patterns can be pruned [40]. In different experiments, CloseGraph showed very good results. It avoided the generation of undesired subgraph patterns during the subgraph generation phase. Also, we reduced the set of FSPs but later studies showed that the CloseGraph missed some of the FSPs, which were not closed but were useful [40, 47].

The aspire of FFSM (**Fast Frequent Subgraph Mining**) was to mine all the connected FSPs from the graph dataset [50]. FFSM follows the DFS scheme from [20, 38] and incorporated new techniques to improve the frequent subgraph discovery efficiency. In FFSM, two efficient candidate enumeration schemes were introduced. These are FFSM-Join and FFSM-Extension. In FFSM, graphs are represented using the triangular matrices. In triangular matrices, diagonals are used to store the graph vertex labels, edge labels elsewhere. Thus, the matrix code is the combination of all matrix values, from left side to right side and row by row [47]. Using the lexicographic ordering, isomorphic graph structures have the identical canonical code. In FFSM, two matrices of graph fragments are joined to result refinements, which at most present two new structures. A restricted extension operation was also required in FFSM, in which a new edge-vertex pair may only be added to the last vertex of a canonical adjacency matrix (CAM). Once the refinement is generated, FFSM permutes lines of the matrix to examine whether a resulting matrix is in canonical form or not? If the matrix is not in canonical form, it can be ignored. In experiments, results reported that FFSM outperforms gSpan. As an central deficiency of the, FFSM needs to scan the occurrence of an additional collection of subgraph which are not canonical [50].

FSP (**Frequent Substructure Pattern Mining**) is a recently presented FSM approach, which suggested an important enhancement in the PG-based FSM category [49]. Following are the potential benefits, which were introduced to PG by FSP: (1) improving the graph canonical representation and defining an association between sub-DFS trees, whose root node shares the identical parent node in the candidate subgraph structure; (2) exploring the similarities of the structure in the DFS search space and showing two one-to-one reflection between the child subgraph patterns and latter sibling subgraph patterns; (3) two different techniques were applied to mine subgraph problem in order to minimize the total number of subgraph patterns as well as graph isomorphism tests efficiently.

The GASTON (**GrAph/Sequence/Tree extractiON**) is a tool, which stores all the embedding to generate just refinements to achieve efficient subgraph isomorphism tests [54]. The core idea of Gaston was to separate many types

of structures such as path, tree, and graph. In Gaston working; initially, only those fragments are considered which are paths or trees and finally cyclic general graphs structures are considered. Thereby, this outsized portion of the mining work is done efficiently. In Gaston, only the last phase is critical as it faces the issue of *NP-completeness* of the subgraph isomorphism examination. Experiments on real datasets, Gaston performance, were very good. Since Gaston focuses on maximal FSGs, there are chances that interesting subgraphs missed from the final resultant FSPs [55].

RING is an integrated approach, which was proposed to mine the frequent representatives subgraph patterns [65]. RING is based on the distance between two graph structure for discovery of the FSPs, involving an invariant vectors' methodology is adopted during mining of subgraph pattern. Furthermore, an invariant distance (i.e., the Euclidean distance) is used between the graphs as an alternative of the edit distance or other variation of the graph distance measures. There are two important phases of RING approach: firstly, it discovers a set of random set of FSPs before adding them in different cluster of subgraphs. Then, it picks the center of the clusters (groups) as representative subgraph patterns, which are original. Secondly, it used the depth-first searching technique. RING used the special structure of indexing, which is called R-Trees. However, RING lacks the comparative analysis with the state-of-the-art FSM approaches.

In a recent study on FSM, GraphSig is proposed to discover the discriminative subgraph patterns with low frequencies, and GraphSig was proposed in [64–66]. GraphSig is a scalable approach, which extracts statistically significant patterns from a large collection of graphs. In GraphSig, using a random walk first of all graphs is converted into feature vectors. Domain knowledge was involved in selecting the important feature set from the entire set of graph features. Results reported in GraphSig showed that GraphSig is able to discover the discriminating patterns in a big graph database and still with small values of the frequencies. There is a need to decide how to decide the statistical significance rather than assuming manually, and significant graph feature selection needs to be improved further.

GraphRank is another important work in FSM field, which used the feature vector to represent a graph structure [67]. The GraphRank FSM approach is proposed to estimate the statistical significance of FSPs in a given graph dataset. GraphRank defined the statistical significance as the probability that a given graph g occurs in the graph dataset of random graphs with the value of support $\mu \geq \mu_0$, which is termed as p value of the graph g . GraphRank converts a subgraph into a feature vector and calculates the importance of each subgraph taking into account the usefulness of the presence of the equivalent vector. Furthermore, in order to acquire a probability distribution based on the support of the feature vector, GraphRank used the probability of the feature

vector in a random vector dataset based on the a priori probability of the basic items. Furthermore, the computed p value is used for the ranking of FSPs. Based on this feature vector mining, significant closed FSPs were also extracted. Their experimental outputs depicted that the GraphRank is an efficient approach and helpful for ranking of the frequent subgraphs by their statistical significance.

In addition to the above-discussed FSM extensive literature, some recent works on FSM included [57, 68, 69]. FS3 is a sampling-based FM algorithm proposed by [57] to handle the scalability issue, which arises when input graph datasets are of massive size. In [68], authors proposed LC-mine framework. The LC-mine framework is a generic and an efficient framework for FSGs' discovery maintaining the local consistency techniques. The "bias in the graph projection operator" is the core idea introduced in LC-mine.

FSM-H [70] is recently proposed distributed FSM algorithm method over a MapReduce-based framework. This framework can deal with real-world graph structured data, which may be growing in its size as well as in quantity. It works in three different phases: partitioning of data, the preparation, and the mining. In the first phase of FSM-H, input graph dataset is partitioned into k -disjoint parts, such that there are equal number of graphs in each of these partitions. Further, in parallel all the edges are removed from the graph being processed, which did not satisfy the support threshold value. In FSM-H, mining is done in the second preparation phase and the mining. FSM-H comparison with the [52] on different benchmark graph datasets showed the better performance of the FSM-H. FSM-H is evaluated with the graph datasets from real-world and few big synthetic data sets are also used. These experiments showed the efficiency of this new FSM-H FSM technique for discovering of the FSGs from massive size graphs [70].

3 A Ranked Frequent Subgraph Discovery Framework (A-RAFF)

A-RAFF is decomposed into three interlinked layers: the preprocessing layer; graph pattern mining layer; and an analytical layer, which are shown in Fig. 5. Each of the three layers focuses on different functionalities, and together they structure the conceptual framework.

A-RAFF framework works with the transaction graph datasets, with small- to medium-size graph structures. Each graph is represented using the labels of the node and set of vertices along the labels on the edges. In the subsequent section, each of the three layers describes one by one.

The fundamental function of the preprocessing layer is to prepare the graph dataset for the graph pattern mining layer. In the preprocessing layer, graph dataset is cleanup, i.e., the graphs with isolated or disconnected nodes are detected and

Fig. 5 A-RAFF—a proposed layered framework

are removed from each of the graph structure. Moreover, if the dataset contains any graph which has single or two nodes with single edges are also removed from the graph dataset. In addition to this, two major tasks of the preprocessing layer are: graph dataset feature selection and partitioning of the graph dataset.

A graph dataset may exhibit different features, but all of the features are not important [58, 61]. The feature may include node labels, edge labels, weights to the individual nodes or weights on the edges between the nodes, the direction of the edge, and so forth [84, 85]. The second major function of the preprocessing layer is to partition the graphs into smaller chunks. In the graph dataset, larger/massive graph structures are distinguished, and then, such structures are partitioned. In the literature, various graph partitioning techniques and tools are available [71–77]. In the proposed framework A-RAFF, the graph partitioning is performed using a well-known graph partitioning tool KaHIP [71]. The KaHIP tool implemented a multilevel graph partitioning algorithm, called KaFFPa (Karlsruhe Fast Flow Partitioning). The KaFFPa algorithm exploits a novel local improvement technique, which is based on max-flow and min-cut computations. Furthermore, KaFFPa used more

localized FM searches in addition to involving of a sophisticated global search strategy transferred from multi-grid linear solvers' problem [72]. Finally, the graph partitions are forwarded to the pattern mining layer.

to objects and an edge between objects depicts the similarities [78]. Very limited research work is available in ranking on graph structure data such as in search engine [75, 77]. In graph mining domain, mainly research community

Proposed Algorithm 1: A-RAFF ($\mathbb{G}\mathbb{D}$, σ , $\mathcal{RF}$)

1. **Input:** $\mathbb{G}\mathbb{D}$ = a graphs dataset of the labelled undirected graphs
 2. σ = minimum threshold
 3. **Output:** $\mathcal{RF}$, a set of ranked frequent subgraph patterns
 4. count all the features and put in the feature set “**F**”
 5. **for each** feature f in the **F**-set **do**
 6. Identify most informative graph features
 7. **repeat** until all features are checked
 8. partition the graph using KaHIP Tool based on an **F**-set [80]
 9. $\mathcal{F} \leftarrow \emptyset$ $\mathcal{F}$ denotes the frequent subgraph patterns set
 10. $\mathcal{F} \leftarrow$ discovered 1-frequent subgraphs patterns
 11. **for each** graph $g_{i_partition} \in G_i$ **do**
 12. FSPs($\mathbb{G}\mathbb{D}$, $g_{i_partition}$, σ , $\mathcal{F}$)
 13. **end**
 14. FSP-Rank($\mathcal{F}$)
 15. **return** $\mathcal{RF}$
-

The second layer in the A-RAFF framework is the mining layer. This layer is responsible for the discovery and ranking of the FSPs. This layer retrieves the preprocessed graph structures from the graph preprocessing layer as an input and discovers the frequent subgraph. Next, the discovered FSPs are ranked using the proposed frequent subgraph ranking scheme. For ranking of the FSPs, different features of the mined FSPs and the individual FSP are used. Once the FSPs are ranked, then these are passed to the next layer of the conceptual framework for exploration of the trends in the discovered FSPs.

FSP-Rank: Ranking is considered as a significant task in the graph theory. In a graph structure, vertices correspond

is interested in the relative importance of the graph nodes with the top ranked node. In [79], centrality measure was used to compute the rank of the graph node. Ranking is also playing very significant role in FSM techniques. Some work has been done in ranking the discovered FSPs [67], in which the challenge of ranking is addressed by computing the statistical significance of the frequent subgraphs. However, there is still need of applying the ranking to the FSPs for better and effective patterns in the final set, which will be used for further analysis.

Proposed Algorithm 2: FSPs ($\mathbb{G}\mathbb{D}, g_i_partition, \sigma, \mathcal{F}$)

Input: $\mathbb{G}\mathbb{D}$ = a graph dataset, $g_i_partition$ = a graph partition, and
 σ = Support threshold value

Output: $\mathcal{F}$ mined frequent subgraph patterns

```

1. if  $g_i\_partition \in \mathcal{F}$  then
2.   return
3. else
4.    $\mathcal{F} \leftarrow \mathcal{F} \cup g_i\_partition$ 
5. end
6. extend  $g_i\_partition$  by adding all edges “e”  $\in \mathbb{G}\mathbb{D}$  such that
7.    $extended\_g_i \leftarrow g_i\_partition \cup e$ 
8. foreach  $extended\_g_i$  from Line. (7) do
9.   if support ( $extended\_g_i$ )  $\geq \sigma |\mathbb{G}\mathbb{D}|$  then
      $\text{FSPs}(\mathbb{G}\mathbb{D}, extended\_g_i, \sigma, \mathcal{F})$ 
10. else
     return
11. end
12. return  $\mathcal{F}$ 

```

In the proposed A-RAFF framework, once the FSPs are discovered, the next step is to rank the extracted patterns. For FSPs ranking, the FSP-Rank algorithm is proposed. The FSP-Rank measure involves different characteristics of the FSPs and computes the rank value for each of the FSPs. Using the ranked values, $f(R_k)$ as computed from Eq. (1), of the FSPs, different duplicate pattern structures were identified. Such duplicate structures were removed from the final result set of the FSPs discovered by the proposed A-RAFF framework. Therefore, total number of FSPs are reduced in the final result set.

$$f(R_k) = (1 - \lambda) * \sum_{i=1}^n \left(W_i + \lambda * \left(\frac{D_i}{n_i} \right) \right) \quad (1)$$

where λ is a normalized factor and is calculated from Eq. (2). W represents the total of the weight of all the nodes in the i th frequent subgraph. This parameter, W , is used when the discovered FSPs are weighted. D_i denotes the degree of FSPs. The value of D_i is the total of in-degree and out-degree of i th FSP, and n represents the total number of nodes in the i th FSP. The value of $f(R_k)$ will always lie in the range ($0 \leq f(R_k) \leq 1$).

$$\lambda = \left(\frac{\sum_{i=1}^n (\text{FSG}_i)}{\sum_{i=1}^n T(V_i)} \right) \quad (2)$$

where (FSG_i) denotes the number of discovered FSPs and $T(V_i)$ represents the total number of vertices found in all of the n FSPs discovered so far. Therefore, the discovered FSPs are finally ranked based on the value computed from Eq. (1). Furthermore, an example is discussed here, to have in-depth understanding of the working on the concepts of the FSPs ranking described by Eqs. (1) and (2).

Proposed Algorithm 3: FSP-Rank(FSPs, $\mathcal{RF}$)

Input: FSPs = frequent subgraph patterns

Output: $\mathcal{RF}$, set of rank frequent subgraph patterns

Compute λ using Equation (2)

for each FSP in FSPs **do**

Score (D_i) $\leftarrow \sum_{i=0}^n \text{FSP}_i$ (in - degree + out - degree)

Compute $f(R_k)$ Using Equation (1)

$\mathcal{RF} \leftarrow \mathcal{RF} \cup \text{FSP}$

next

return $\mathcal{RF}$

Example To describe the process of computing the rank values, a simple example is given. Consider, there are following sample five FSPs mined from the graph dataset.

In Fig. 6, all the FSPs extracted from an undirected graph dataset are listed. The computation of the normalization factor λ is shown in Table 1. Firstly, the degree for each FSP is computed. Using the number of nodes in each FSP and degree score, the normalization factor is computed. The value of normalization factor is computed using Eq. (2) and is shown in column 4 of Table 1.

$$\lambda = \left(\frac{\sum_{i=1}^n (\text{FSP})}{\sum_{i=1}^n T(V_i)} \right) = (\text{Total FSPs} / \# \text{ of vertices in all the FSPs})$$

$$= 5/16 = 0.3125$$

In Table 2, the rank value for each of the FSPs is computed. Using the computed value of the term $\lambda * \left(\frac{D_i}{n_i} \right)$, from the last column of Table 2, the ranking score for each of the FSPs is presented in Table 2. In this example, the value of the factor W is used as 0, as the FSPs are discovered from the unweighted graph dataset.

After computing the rank score for each of the FSPs, the FSPs are rearranged according to their ranked score. For example, in Table 3, the FSPs are ranked according to their computed rank values using the proposed FSP-Rank measure, therefore giving the ranked FSPs.

The ranked values for the FSPs are further used to identify the duplicate FSPs. All the frequent graph patterns having the same ranked score were analyzed for possible duplicate structures. After analyzing those structures which have the same ranking score were removed from the final result set of the FSPs. These FSPs were also cross-validated (manually) to ensure that these FSPs are actually the duplicate and repetitive patterns.

Moreover, since FSPs are generated solely based on frequency, not every frequent pattern is equally significant and interesting as well [80–83]. Foregoing any difficulties in defining a measure which correctly identifies what we discover interesting, the second key dilemma is the exponentially outsized search space. That is, there are exponentially

Fig. 6 Sample-extracted FSPs

Table 1 Computation of λ using FSP characteristics

FSPs	D_i = Degree (In-Deg + Out-Deg)	n_i = number of vertices	λ = normalization factor	$\left(\frac{D_i}{n_i} \right)$	$\lambda * \left(\frac{D_i}{n_i} \right)$
FSP-1	2	2	0.3125	1	0.3125
FSP-2	4	3		1.33	0.415625
FSP-3	6	3		2	0.625
FSP-4	6	4		1.5	0.46875
FSP-5	8	4		2	0.625

Table 2 Computation of the final rank score

FSPs	$f(R_k) = (1 - \lambda) * \sum_{i=1}^n \left(W_i + \lambda * \left(\frac{D_i}{n_i} \right) \right)$
FSP-1	0.21484375
FSP-2	0.285742188
FSP-3	0.4296875
FSP-4	0.322265625
FSP -5	0.4296875

Table 3 Rank of the discovered frequent subgraphs based on FSP-rank value

Frequent subgraph	Ranking based on FSP-rank values
FSP-3	0.4296875
FSP-5	0.4296875
FSP-4	0.3222656
FSP-2	0.2857422
FSP-1	0.2148438

many potentially interesting patterns. Naively evaluating these patterns one by one and only reporting those that meet the criteria is hence infeasible for all but the most trivial of pattern languages [84]. As such, in addition to correctly identifying what is interesting, ideally an interestingness measure also defines a structured, easily traversable search space to find these patterns [84–87]. In the next section, we have used three different measures of interestingness including I_{RAE} , I_{CON} , and $I_{Variance}$ to validate the discovered interestingness of the FSPs discovered by the proposed FSM framework, A-RAFF. In our opinion, such measures were not incorporated in the FSM domain until now.

The third layer of A-RAFF is the analytical layer. The goal of the analytical layer is to effectively utilize the discovered ranked FSPs. The analytical layer is therefore focusing on the notions of expediency and meaningfulness of the ranked FSPs in the domain of the graph structure dataset. Consequently, the analysis performed in this layer can be further extended for the benefits of the business and organizations. For example, analysis of the ranked FSPs can be useful to evaluate the trends (such as extraction of the most influential person/friends in the circle of an individual or a community) found in the social sites such as Facebook, Twitter, MySpace, and Orkut.

The three layers of the conceptual framework are mutually functioning on each other. It is anticipated that the results from the preprocessing layer will provide processed graph datasets and provide the basis for the next graph pattern mining and analytical layers. The graph pattern mining layer is the methodical pursuit of activities in the framework. At the meantime, the graph pattern mining layer is acting as a bridge between preprocessing and the analytical layers.

4 Results and Analysis

To evaluate the proposed framework, A-RAFF, a series of experiments is performed. This section presents the detailed analysis and discussion of the experiments. Furthermore, the performance of the A-RAFF framework is also compared with some well-known FSM approaches. The performance analysis of the A-RAFF framework with the chosen FSM approaches is performed based on the time required to discover all the FSPs and also the number of FSPs discovered by each of the FSM approach.

4.1 Evaluation Metrics

The input parameter, support threshold parameter (σ), is used in all the FSM techniques for extraction of FSPs. The value of the σ is kept same for all the FSM approaches in each experiment. For each graph dataset, the proposed A-RAFF framework is compared with the chosen FSM techniques at $\sigma = 10\%$, 20% , 30% , 40% , and 50% . At each of the selected threshold value, we have discovered the interesting FSPs based on the ranking value. In the subsequent section, the results on each of the graph datasets and their analysis are presented.

4.2 Experimental Setup

All the experiments are executed on a 32-bit machine running the Linux operating system with 6 GB memory and 3.0 GHz Intel processor. A-RAFF is implemented using Java programming language with JDK 1.8 and NetBeans as development IDE. The proposed A-RAFF framework is compared to a number of different existing FSM approaches, such as FFSM [50], FSP[49], CloseGraph [52], gSpan [38], FSG [48], SPIN [53], and Gaston [54]. The FSM approaches, FSG, FFSM, SPIN, FSP, and CloseGraph, considered for comparison are implemented using Java language. The executables of two comparative FSM approaches are acquired from their respective authors [38, 53].

4.3 Datasets' Description

Five different graph datasets are used for assessment purposes: three real-world graph datasets and two synthetic graph datasets. Real graph datasets are included: chemical compound, AIDS antiviral screen compound, and DTP human tumor cell line screen (CANSO3SD). These graph datasets are benchmark and are extensively used by the FSM techniques for their performance evaluations [52–55, 88].

The synthetic graph generator was provided by [51]. The synthetic graph generator allows to specify the graphs number (D), average size of graph (T), the number and average

size of the seed graphs (S and I, respectively), and the number of distinctive labels. Two different synthetic graph datasets were generated from experiments. In the first synthetic graph dataset, synthetic graph dataset-1, there were total 10,000 graphs, with an average of 40 vertices (ranging from 1 to 90), and 10 different uniformly distributed labels of the nodes and edge; the second dataset, synthetic graph dataset-2, contained 30,000 graphs. In synthetic graph dataset-2, there were total 30,000 graph structure considered, with an average of 75 nodes (ranging from 1 to 125) and 14 uniformly distributed labels for the graph nodes and edge. The graph datasets considered were divided into two subsets. Eighty percentage of each graph dataset was used for training, and 20% of each graph dataset was reserved for testing of the proposed A-RAFF.

4.4 Experimental Results

An extensive series of the experiment was performed to empirically assess the A-RAFF framework performance in comparison with the other existing FSM approaches. Furthermore, A-RAFF was also compared with the FSM approaches with respect to the number of FSPs discovered. In all of the above experiments, the proposed FSM approach A-RAFF framework outperforms the other FSM approaches under consideration. We have summarized the performance evaluation in the following tables, from Tables 4, 5, 6, 7, and 8, which is giving a clearer picture of the proposed A-RAFF performance with the existing FSM techniques.

In tabular results from Tables 4, 5, 6, 7, and 8, we have shown how much time was improved by the proposed

A-RAFF framework against each existing FSM approach on the entire graph datasets used. Moreover, the best performance of the proposed A-RAFF on each graph dataset at the defined threshold value was shown in bold. The running time consumed by each existing FSM approach to extract the FSPs from the given graph dataset is provided in seconds. From the tabular results shown in Tables 4, 5, 6, 7, and 8, we can see that on most of the graph datasets (both real and synthetic datasets), the performance of the A-RAFF was very promising. Moreover, the highest performance of the A-RAFF was observed against FSG approach. In Table 4, at the threshold value of 10%, the best performance achieved by the A-RAFF was 38.04% against the FSG FSM approach. Similarly, at the threshold value of 20%, A-RAFF best performance was observed against FSG approach, and it was 75.33% on the synthetic graph dataset-1, see the results in Table 5. In some cases, the performance of gSpan, Gaston tool, and FFSM was observed better than the A-RAFF framework. For example, in Table 5, on Chemical Compound and CANSO3SD gSpan and Gaston tools perform well than the A-RAFF. A-RAFF took 11 and 12 s more than the gSpan and Gaston, respectively, to discover the FSPs the threshold value of 30%. Such trends are shown with negative values enclosed in brackets.

In addition to the computational time, we also experimented the proposed A-RAFF based on the number of FSPs reduced. The results clearly show that A-RAFF performed better in terms of discovery and reducing the final set of the FSPs. Moreover, we have observed that at low values of the support threshold parameter, A-RAFF reduced less number of FSPs.

Table 4 A summary of A-RAFF performance comparison with existing FSM approaches (Ex-FSM) on various benchmark graph datasets at support threshold value of 10%

Datasets	Chemical compound			CANSO3SD			AIDS antiviral screen compound			Synthetic graph dataset-1			Synthetic graph dataset-2		
	Runtime Ex-FSM		Time improved by A-RAFF	Runtime Ex-FSM		Time improved by A-RAFF	Runtime Ex-FSM		Time improved by A-RAFF	Runtime Ex-FSM		Time improved by A-RAFF	Runtime Ex-FSM		Time improved by A-RAFF
	Sec	%age		Sec	%age		Sec	%age		Sec	%age		Sec	%age	
gSpan	1800	200	11.11	1785	260	14.57	1700	356	20.94	441	94	21.32	1358	251	18.48
Gaston	1813	213	11.75	1855	330	17.79	1593	249	15.63	431	84	19.49	1294	187	14.45
FFSM	1901	301	15.83	2050	525	25.61	1876	532	28.36	477	130	27.25	1430	323	22.59
FSG	2015	415	20.60	2095	570	27.21	1915	571	29.82	560	213	38.04	1655	548	33.11
FSP	2041	441	21.61	1976	451	22.82	1861	517	27.78	460	113	24.57	1379	272	19.72
SPIN	2091	491	23.48	2001	476	23.79	1955	611	31.25	483	136	28.16	1396	289	20.70
CloseGraph	2100	500	23.81	1777	252	14.18	1846	502	27.19	477	130	27.25	1240	133	10.73

Runtime Ex-FSM: The running time taken by each existing FSM approach (which are listed in the first column) to extract all FSPs

Runtime is given in seconds

Sec. Column in the “Time Improved by A-RAFF” column shows the improved time in seconds, whereas %age shows the percentage of time improved. The highest improvement in terms of running time on a particular graph dataset in time improved by A-RAFF column

Table 5 A summary of A-RAFF performance comparison with existing FSM approaches (Ex-FSM) on various benchmark graph datasets at support threshold value of 20%

Datasets	Chemical compound			CANSO3SD			AIDS antiviral screen compound			Synthetic graph dataset-1			Synthetic graph dataset-2		
FSM approaches	Runtime Ex-FSM	Time improved by A-RAFF		Runtime Ex-FSM	Time improved by A-RAFF		Runtime Ex-FSM	Time improved by A-RAFF		Runtime Ex-FSM	Time improved by A-RAFF		Runtime Ex-FSM	Time improved by A-RAFF	
		Sec	%age		Sec	%age		Sec	%age		Sec	%age		Sec	%age
gSpan	115	5	4.35	143	(-11)	N/A	325	101	31.08	61	24	39.34	163	81	49.69
Gaston	98	(-12)	N/A	165	11	6.67	296	72	24.32	45	8	17.78	115	33	28.70
FFSM	175	65	37.14	171	17	9.94	413	189	45.76	38.7	1.7	4.39	96	14	14.58
FSG	189	79	41.80	320	166	51.88	427	203	47.54	150	113	75.33	246	164	66.67
FSP	121	11	9.09	167	13	7.78	316	92	29.11	42	5	11.90	113	31	27.43
SPIN	195	85	43.59	174	20	11.49	327	103	31.50	101	64	63.37	121	39	32.23
CloseGraph	143	33	23.08	168	14	8.33	321	97	30.22	87	50	57.47	94	12	12.77

Table 6 A summary of A-RAFF performance comparison with existing FSM approaches (Ex-FSM) on various benchmark graph datasets at support threshold value of 30%

Datasets	Chemical compound			CANSO3SD			AIDS antiviral screen compound			Synthetic graph dataset-1			Synthetic graph dataset-2		
FSM approaches	Runtime Ex-FSM	Time improved by A-RAFF		Runtime Ex-FSM	Time improved by A-RAFF		Runtime Ex-FSM	Time improved by A-RAFF		Runtime Ex-FSM	Time improved by A-RAFF		Runtime Ex-FSM	Time improved by A-RAFF	
		Sec	%age		Sec	%age		Sec	%age		Sec	%age		Sec	%age
gSpan	73	8	10.96	39	5	12.82	125	8	6.40	24	9.3	38.75	37	14	37.84
Gaston	65	0	0.00	41	7	17.07	103	(-14)	N/A	17.8	3.1	17.42	29	6	20.69
FFSM	75	10	13.33	57	23	40.35	143	26	18.18	12	(-2.7)	N/A	29	6	20.69
FSG	95	30	31.58	180	146	81.11	151	34	22.52	110	95.3	86.64	165	142	86.06
FSP	68	3	4.41	51	17	33.33	120	3	2.50	17.8	3.1	17.42	29	6	20.69
SPIN	95	30	31.58	41	7	17.07	147	30	20.41	57	42.3	74.21	29	6	20.69
CloseGraph	72	7	9.72	43	9	20.93	133	16	12.03	41	26.3	64.15	29	6	20.69

Table 7 A summary of A-RAFF performance comparison with existing FSM approaches (Ex-FSM) on various benchmark graph datasets at support threshold value of 40%

Datasets	Chemical compound			CANSO3SD			AIDS antiviral screen compound			Synthetic graph dataset-1			Synthetic graph dataset-2		
FSM approaches	Runtime Ex-FSM	Time improved by A-RAFF		Runtime Ex-FSM	Time improved by A-RAFF		Runtime Ex-FSM	Time improved by A-RAFF		Runtime Ex-FSM	Time improved by A-RAFF		Runtime Ex-FSM	Time improved by A-RAFF	
		Sec	%age		Sec	%age		Sec	%age		Sec	%age		Sec	%age
gSpan	57	4	7.02	24	1	4.17	39	6	15.38	17	7	41.18	27	8	29.63
Gaston	52	(-1)	N/A	28	5	17.86	36	3	8.33	12	2	16.67	20	1	5.00
FFSM	67	14	20.90	28	5	17.86	53	20	37.74	10	0	0.00	20	1	5.00
FSG	87	34	39.08	120	97	80.83	69	36	52.17	76	66	86.84	110	91	82.73
FSP	59	6	10.17	35	12	34.29	44	11	25.00	14	4	28.57	24	5	20.83
SPIN	67	14	20.90	45	22	48.89	43	10	23.26	23	13	56.52	31	12	38.71
CloseGraph	71	18	25.35	37	14	37.84	47	14	29.79	27	17	62.96	13	(-6)	N/A

Comparing FSP-Rank with Existing Ranking Measure

GraphRank: In FSM domain, to the best of our knowledge, there exists very limited work on ranking. The performance of the proposed FSP-Rank was also experimented with an existing ranking measure called GraphRank [67]. GraphRank was used for computing the statistical significance of the FSPs. GraphRank defined the statistical significance as the probability that a graph g occurs in the graph dataset of random graphs with the value of support $\mu \geq \mu_0$, which is termed as *p-value* of the graph g [67].

We have compared the proposed FSP-Rank with GraphRank measure and investigated the effectiveness of the FSP-Rank measure over the GraphRank. Using the FSP-Rank measure, the FSPs with high score of rank are considered more significant as that with low values of rank score. In contrast, GraphRank computes the rank score using the statistical significance of the FSPs [67]. Furthermore, in GraphRank, a FSP is considered more significant if it has value less than *p-value* as compared to the other FSPs. GraphRank applied its ranking scheme to the Close-FSPs; therefore, we have simulated our results based on the closed FSPs discovered and ranked by the A-RAFF. The FSPs were ranked using the significance value of 0.6 (i.e., *p-value* = 0.6). Table 11 shows the detailed results.

In Table 11, we have shown the time required by FSP-Rank and the GraphRank to rank the discovered FSPs. In most of the cases, FSP-Rank took less time to rank the discovered FSPs as compared to the GraphRank approach. For example, at $\sigma = 10$, GraphRank consumed more than 80 s to rank the discovered FSPs. Only at $\sigma = 30$, GraphRank performed better than the proposed FSP-Rank.

Measuring the Interestingness of the Discovered FSPs:

Furthermore, we have evaluated the interestingness of the FSPs discovered using A-RAFF. Interestingness measures play an important role in KDD, regardless of the kind of patterns being mined [82]. All the patterns mined are not interesting or whatever the pattern mined by data mining tools is not interesting [87]. To analyze the interestingness of a rule set, various interestingness measures are proposed and analyzed by the researchers. Interestingness measures are generally divided into two main categories of objective and subjective measures of interest [82]. An objective measure uses the raw data, and no knowledge about the user or application is needed. More objective measures are based on theories in statistics, probability, or information theory. Coverage, support, accuracy, generality, peculiarity, reliability, diversity, and conciseness depend simply on the data and patterns and consequently can be considered objective [83]. The objective measures were based on the statistical strengths or properties of the discovered patterns to measure their degree of interestingness. A subjective measure takes into account both the data and the user of these data. To

define a subjective measure, access to the user's domain or background knowledge about the data is required [82, 86]. This access can be obtained by interacting with the user during the data mining process or by explicitly signifying the user's expectations or knowledge. The measures usually determine whether a pattern is "actionable" and/or "unexpected". An unexpectedness, actionable, novel is the criteria under subjective nature.

To compute the interestingness of the discovered FSPs, we used three different objective interestingness measure, called the I_{RAE} measure, the measure is I_{CON} and $I_{Variance}$ measure [82]. The I_{RAE} , I_{CON} , and $I_{Variance}$ interestingness measures are computed using Eqs. (3), (4) and (5), respectively,

$$I_{RAE} = \frac{\sum_{i=1}^m n_i(n_i - 1)}{N(N - 1)} \quad (3)$$

$$I_{CON} = \sqrt{\frac{(\sum_{i=1}^m p_i^2) - \bar{q}}{1 - \bar{q}}} \quad (4)$$

$$I_{Variance} = \frac{\sum_{i=1}^m (p_i - \bar{q})^2}{m - 1} \quad (5)$$

where n_i shows the number of FSPs discovered by the A-RAFF; m corresponds to the total number of tests/experiments (i.e., a total number of different threshold settings, on which we have performed the experiments. In our problem setting, there are five different threshold settings; we have extracted the FSPs at 10%, 20%, 30%, 40%, and 50% threshold value); N shows an aggregate of total FSPs at all the threshold values; p_i shows the actual probability of a specific test i ; and $\bar{q} = \frac{1}{m}$ and represents the uniform probability for test i for all $(i = 1, 2, 3, \dots, m)$.

We have applied the I_{RAE} , I_{CON} , and $I_{Variance}$ interestingness measures on the FSPs to measure the discovered patterns diversity criteria. To compute the diversity criteria, we used the extracted FSPs using A-RAFF and it is given in Tables 9 and 10. The I_{RAE} , I_{CON} , and $I_{Variance}$ Interestingness measure values computed from these tabular FSP results are given in Table 12.

Table 12 shows the different interestingness measure values against each graph dataset where we have mined the FSPs. These values represent the interestingness of the FSPs discovered from each of the graph datasets at different threshold parameter settings. For example, on chemical compound graph dataset the interestingness value computed by I_{RAE} is 0.84, and a high value of I_{RAE} represents the more interestingness of the patterns. Similarly, the maximum interestingness value for the FSPs on the chemical compound graph dataset computed by the I_{CON} was 1. Overall interestingness of the discovered FSPs was shown very well

Table 8 A summary of A-RAFF performance comparison with existing FSM approaches (Ex-FSM) on various benchmark graph datasets at support threshold value of 50%

Datasets	Chemical compound			CANSO3SD			AIDS antiviral screen compound			Synthetic graph dataset-1			Synthetic graph dataset-2		
FSM approaches	Runtime Ex-FSM	Time improved by A-RAFF		Runtime Ex-FSM	Time improved by A-RAFF		Runtime Ex-FSM	Time improved by A-RAFF		Runtime Ex-FSM	Time improved by A-RAFF		Runtime Ex-FSM	Time improved by A-RAFF	
		Sec	%age		Sec	%age		Sec	%age		Sec	%age		Sec	%age
gSpan	29	8	27.59	20	6	30.00	21	2	9.52	12	6.6	55.00	14	4.5	32.14
Gaston	26	5	19.23	19	5	26.32	14	(-5)	N/A	9	3.6	40.00	13	3.5	26.92
FFSM	43	22	51.16	21	7	33.33	51	32	62.75	6	0.6	10.00	11	1.5	13.64
FSG	65	44	67.69	85	71	83.53	62	43	69.35	55.5	50.1	90.27	88	78.5	89.20
FSP	34	13	38.24	22	8	36.36	27	8	29.63	11	5.6	50.91	15	5.5	36.67
SPIN	35	14	40.00	32	18	56.25	25	6	24.00	15	9.6	64.00	19	9.5	50.00
CloseGraph	23	2	8.70	25	11	44.00	23	4	17.39	16	10.6	66.25	8	(-1.5)	N/A

Table 9 A summary of number of FSPs discovered by A-RAFF with existing FSM approaches (Ex-FSM) on various benchmark graph datasets at various support threshold value of (10% to 50%)

Datasets	Chemical compound			CANSO3SD graph dataset			AIDS antiviral screen compound		
Threshold value	Number of FSPs discovered			Number of FSPs discovered			Number of FSPs discovered		
	Ex-FSM: gSpan, Gas- ton, FFSM, FSG, FSP	Proposed A-RAFF	FSPs reduced By A-RAFF %age	Ex-FSM	Proposed A-RAFF	FSPs reduced By A-RAFF %age	Ex-FSM	Proposed A-RAFF	FSPs reduced By A-RAFF %age
10	16,555	13,258	19.91	1640	1327	19.08	12,025	10,230	14.92
20	1010	811	19.70	438	327	25.34	2600	1876	27.84
30	420	328	21.90	265	171	35.47	1215	863	28.97
40	130	99	23.84	169	119	29.58	778	621	20.17
50	55	39	29.09	85	61	28.23	356	267	25.00

*In Tables 10 and 11, we have not included CloseGraph and SPIN as both of these FSM approaches discovered the specific type of FSPs (CloseGraph FSM approach discovers the closed FSPs, while SPIN discovers the maximal FSPs), whereas A-RAFF framework is devised to discover all types of FSPs including closed and maximal FSPs

Table 10 A summary of number of FSPs discovered by A-RAFF with existing FSM approaches (Ex-FSM) on various benchmark synthetic graph datasets at various support threshold value of (10% to 50%)

Datasets	Synthetic graph dataset-1			Synthetic graph dataset-2		
Threshold value	Number of FSPs discovered			Number of FSPs discovered		
	Ex-FSM: gSpan, Gaston, FFSM, FSG, FSP	Proposed A-RAFF	FSPs reduced by A-RAFF %age	Ex-FSM	Proposed A-RAFF	FSPs reduced by A-RAFF %age
10	3035	2500	17.62	9015	7033	21.98
20	770	613	20.38	2135	1763	17.53
30	445	331	25.61	1143	904	20.90
40	280	205	26.78	776	519	33.11
50	150	107	28.66	348	231	33.04

Table 11 Comparison of proposed FSP-rank with GraphRank on chemical compound (time is given in seconds)

Threshold (σ)	Total FSPs discovered	Time required to rank the FSPs	
		GraphRank	FSP-Rank
10	360	1688	1600
20	133	174	110
30	99	55	65
40	83	60	53
50	31	29	21

Table 12 Various interestingness measures values for FSPs discovered by the A-RAFF

Graph datasets	Interestingness measure values		
	I_{RAE}	I_{CON}	$I_{Variance}$
Chemical compound	0.84	1	0.02
CANSO3SD graph dataset	0.40	0.77	0.05
AIDS antiviral screen compound	0.57	0.84	0.04
Synthetic graph dataset-1	0.48	0.78	0.05
Synthetic graph dataset-2	0.50	0.78	0.05

by the I_{CON} interestingness on all the considered graph datasets. The results described in Table 12 give the indication the FSPs generated from the proposed FSM framework A-RAFF is very diverse.

5 Conclusion

In this paper, we study the problem of ranking the frequent subgraph patterns. We highlight the importance of the problem and show that current solutions are inapplicable. Therefore, a novel solution is proposed, A-RAFF, to rank the FSPs from transaction graph databases. It is based on utilizing the preprocessing of the graph dataset to identify the which nodes of the input graphs to postpone and which subgraph nodes can give quicker and effective ranked frequent subgraph patterns. This operation on the graph dataset is executed to enhance the performance of the next phase, the graph pattern mining. Furthermore, we proposed the definition of the ranking measure, FSP-Rank, to rank the mined FSPs. Also, we have used different interestingness measures to show the importance of the discovered FSPs. Our experimental results are significantly better than those with existing approaches with regard to both the runtime and reducing the discovered FSPs. Thus, we believe that the proposed methodology can be an efficient tool for ranking

of the discovered frequent subgraph patterns. Although this research study reached to its objectives, few unavoidable weaknesses are observed, which are given as follows. First, the findings of this study are restricted on the discovery of the FSPs from medium-size graph datasets. However, this would be better if frequent subgraph discovery was performed on big graphs, as there is ever-increasing demand of handling the big graph structures. Second, as the proposed technique for frequent subgraph discovery works with transaction graph datasets, we have to convert the big social graphs into smaller one for exploration of trends. Third, although the proposed ranking scheme produced good results, however, to further improve the results, new similarity measures are needed.

Declaration

Conflict of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Kim, J., Hastak, M.: Social network analysis. *Int. J. Inf. Manag.* **38**, 86–96 (2018)
- Rehman, S.U., Asghar, S.: Online social network trend discovery using frequent subgraph mining. *Soc. Netw. Anal. Min.* **10**(1), 1–13 (2020)
- Pal, M., Samanta, S., Ghorai, G.: *Modern Trends in Fuzzy Graph Theory*, pp. 7–93. Springer, Berlin (2020)
- Reichle, R., Gaul, M., Nicklis, S., Hornung, C., Nissel, D., Schneider, S., et al.: *Navigation apparatus and method for displaying a navigation tree on a display unit*, ed. Google Patents (2017)
- Freedman, S., Jin, G.Z.: The information value of online social networks: lessons from peer-to-peer lending. *Int. J. Ind. Organ.* **51**, 185–222 (2017)
- Jalili, M., Orouskhani, Y., Asgari, M., Alipourfard, N., Perc, M.: Link prediction in multiplex online social networks. *R. Soc. Open Sci.* **4**, 160863 (2017)
- Samanta, S., Pal, M., Mahapatra, R., Das, K., Bhadoria, R.S.: A study on semi-directed graphs for social media networks. *Int. J. Comput. Intell. Syst.* **14**(1), 1034–1041 (2021)
- Atastina, I., Sitohang, B., Saptawati, G., Moertini, V.: A review of big graph mining research. In: *IOP Conference Series: Materials Science and Engineering*, p. 012065 (2017)

9. Rehman, S., Ali, T., Nawaz, A.: g-Sum: a graph summarization approach for a single large social network. In: EAI Endorsed Transactions on Scalable Information Systems (2021). <https://doi.org/10.4108/eai.23-3-2021.169073>
10. Koyutürk, M., Grama, A., Szpankowski, W.: An efficient algorithm for detecting frequent subgraphs in biological networks. *Bioinformatics* **20**, i200–i207 (2004)
11. Das, K., Samanta, S., Pal, M.: Study on centrality measures in social networks: a survey. *Soc. Netw. Anal. Min.* **8**(1), 1–11 (2018)
12. Prado, A., Plantevit, M., Robardet, C., Boulicaut, J.-F.: Mining graph topological patterns: finding covariations among vertex descriptors. *IEEE Trans. Knowl. Data Eng.* **25**, 2090–2104 (2013)
13. Yan, X., Zhu, F., Han, J., Yu, P.S.: Searching substructures with superimposed distance. In: ICDE'06. Proceedings of the 22nd International Conference on Data Engineering, 2006, pp. 88–88 (2006)
14. Csérmely, P., Korcsmáros, T., Kiss, H.J., London, G., Nussinov, R.: Structure and dynamics of molecular networks: a novel paradigm of drug discovery: a comprehensive review. *Pharmacol. Ther.* **138**, 333–408 (2013)
15. Takigawa, I., Mamitsuka, H.: Graph mining: procedure, application to drug discovery and recent advances. *Drug Discov. Today* **18**, 50–57 (2013)
16. Zhang, X., Ouyang, T., Pan, D., Si, X., Rahman, S.: Upstream pilot structure in point to multipoint orthogonal frequency division multiplexing communication system, ed. Google Patents (2016)
17. Xiao, H., Hu, Y., Yan, K., Ouyang, S.: Power allocation and relay selection for multisource multirelay cooperative vehicular networks. *IEEE Trans. Intell. Transp. Syst.* **17**, 3297–3305 (2016)
18. Deshpande, M., Kuramochi, M., Wale, N., Karypis, G.: Frequent substructure-based approaches for classifying chemical compounds. *IEEE Trans. Knowl. Data Eng.* **17**, 1036–1050 (2005)
19. Huan, J., Wang, W., Bandyopadhyay, D., Snoeyink, J., Prins, J., Tropsha, A.: Mining protein family specific residue packing patterns from protein structure graphs. In: Proceedings of the Eighth Annual International Conference on Research in Computational Molecular Biology, pp. 308–315 (2004)
20. Borgelt, C., Berthold, M.R.: Mining molecular fragments: finding relevant substructures of molecules. In: 2002 IEEE International Conference on Data Mining, 2002. ICDM 2003. Proceedings, pp. 51–58 (2002)
21. Eichinger, F., Böhm, K., Huber, M.: Mining edge-weighted call graphs to localise software bugs. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 333–348 (2008)
22. Liu, C., Yan, X., Yu, H., Han, J., Yu, P.S.: Mining behavior graphs for “backtrace” of noncrashing bugs. In: Proceedings of the 2005 SIAM International Conference on Data Mining, pp. 286–297 (2005)
23. Liu, Y., Xu, S., Duan, L.: Relationship emergence prediction in heterogeneous networks through dynamic frequent subgraph mining. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, pp. 1649–1658 (2014)
24. Panda, B., Tripathy, S.N., Sethi, N., Samantray, O.P.: A comparative study on serial and parallel web content mining. *Int. J. Adv. Netw. Appl.* **7**, 2882 (2016)
25. Algur, S.P., Bhat, P.: Web video object mining: expectation maximization and density based clustering of web video metadata objects. *Int. J. Inf. Eng. Electron. Bus.* **8**, 69 (2016)
26. Baeza-Yates, R., Boldi, P.: Web structure mining. In: *Advanced Techniques in Web Intelligence-I*. Springer, pp. 113–142 (2010)
27. Rehman, S.U., Haider, A.A., Afzal, T., Aziz, K.: Measuring the relevancy between tags and citation in social web. *Res. J. Appl. Sci. Eng. Technol.* **7**(24), 5172–5178 (2014)
28. Mahapatra, R., Samanta, S., Pal, M., Xin, Q.: Link prediction in social networks by neutrosophic graph. *Int. J. Comput. Intell. Syst.* **13**(1), 1699–1713 (2020)
29. Jiang, F., Kawagoe, K., Leung, C.K.: Big social network mining for following patterns. In: Proceedings of the Eighth International C* Conference on Computer Science & Software Engineering, pp. 28–37 (2015)
30. Scholer, P.: Detection of fraud ranking for mobile app using fuzzy logic
31. Alsmadi, I., Alhami, I.: Clustering and classification of email contents. *J. King Saud Univ.-Comput. Inf. Sci.* **27**, 46–57 (2015)
32. Tang, G., Pei, J., Luk, W.-S.: Email mining: tasks, common techniques, and tools. *Knowl. Inf. Syst.* **41**, 1–31 (2014)
33. Aery, M., Chakravarthy, S.: InfoSift: adapting graph mining techniques for text classification. In: FLAIRS Conference, pp. 277–282 (2005)
34. Shaheen, M., ur Rehman, S., Ghaffar, F.: Correlation and congruence modulo based clustering technique and its application in energy classification. In: *Sustainable Computing: Informatics and Systems*, p. 100561 (2021)
35. Palmer, C.R., Gibbons, P.B., Faloutsos, C.: ANF: a fast and scalable tool for data mining in massive graphs. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 81–90 (2002)
36. Vanetik, N., Gudes, E., Shimony, S.E.: Computing frequent graph patterns from semistructured data. In: 2002 IEEE International Conference on Data Mining, 2002. ICDM 2003. Proceedings, pp. 458–465 (2002)
37. Vanetik, N., Gudes, E.: Mining frequent labeled and partially labeled graph patterns. In: 20th International Conference on Data Engineering, 2004. Proceedings, pp. 91–102 (2004)
38. Yan, X., Han, J.: gspan: graph-based substructure pattern mining. In: 2002 IEEE International Conference on Data Mining, 2002. ICDM 2003. Proceedings, pp. 721–724 (2002)
39. Rehman, S.U., Asghar, S.: A-RAFF: a ranked frequent pattern-growth subgraph pattern discovery approach. *J. Internet Technol.* **20**(1), 257–267 (2019)
40. Patel, H.J., Prajapati, R., Panchal, M., Patel, M.: A Survey of Graph Pattern Mining Algorithm and Techniques. *Int. J. Appl. Innov. Eng. Manag. (IIAEM)* **2**, 125–129 (2013)
41. Rehman, S.U., Asghar, S., Fong, S.J.: Optimized and frequent subgraphs: how are they related? *IEEE Access* **6**, 37237–37249 (2018)
42. Dhiman, A., Jain, S.: Frequent subgraph mining algorithms for single large graphs—a brief survey. In: *International Conference on Advances in Computing, Communication, & Automation (ICACCA)* (Spring), pp. 1–6 (2016)
43. Rehman, S.U., Asghar, S., Fong, S.J.: An efficient ranking scheme for frequent subgraph patterns. In: Proceedings of the 2018 10th International Conference on Machine Learning and Computing, pp. 257–262 (2018, February)
44. Rehman, S.U., Asghar, S., Zhuang, Y., Fong, S.J.: Performance evaluation of frequent subgraph discovery techniques. *Math. Prob. Eng.* **2014**, 1–6 (2014)
45. Santhi, S., Padmaja, P.: A survey of frequent subgraph mining algorithms for uncertain graph data. *Int. Res. J. Eng. Technol. (IRJET)* **2**, 688–696 (2015)
46. Naderi, H.: A survey of frequent subtrees and subgraphs mining methods. *Int. J. Comput. Sci. Bus. Inform.* **14** (2014)
47. Jiang, C., Coenen, F., Zito, M.: A survey of frequent subgraph mining algorithms. *Knowl. Eng. Rev.* **28**, 75–105 (2013)
48. Rehman, S.U., Khan, A.U., Fong, S.J.: Graph mining: a survey of graph mining techniques. In: *Seventh International Conference on*

- Digital Information Management (ICDIM 2012). IEEE, pp. 88–92 (2012, August)
49. Han, S., Ng, W.K., Yu, Y.: Fsp: frequent substructure pattern mining. In: 2007 6th International Conference on Information, Communications & Signal Processing, pp. 1–5 (2007)
 50. Huan, J., Wang, W., Prins, J.: Efficient mining of frequent subgraphs in the presence of isomorphism. In: Third IEEE International Conference on Data Mining, 2003. ICDM 2003, pp. 549–552 (2003)
 51. Kuramochi, M., Karypis, G.: Frequent subgraph discovery. In: Proceedings IEEE International Conference on Data Mining, 2001. ICDM 2001, pp. 313–320 (2001)
 52. Yan, X., Han, J.: CloseGraph: mining closed frequent graph patterns. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 286–295 (2003)
 53. Huan, J., Wang, W., Prins, J., Yang, J.: Spin: mining maximal frequent subgraphs from graph databases. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 581–586 (2004)
 54. Nijssen, S., Kok, J.N.: The gaston tool for frequent subgraph mining. *Electron. Notes Theor. Comput. Sci.* **127**, 77–87 (2005)
 55. Holder, L.B., Cook, D.J., Djoko, S.: Substructure discovery in the SUBDUE system. In: KDD Workshop, pp. 169–180 (1994)
 56. Mumu, T.S.: Sequential pattern mining of social networks
 57. T. K. Saha and M. Al Hasan, "FS3: A sampling based method for top-k frequent subgraph mining," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 8, pp. 245–261, 2015.
 58. Hu, J., Shen, X., Shao, Y., Bystroff, C., Zaki, M.J.: Mining protein contact maps. In: Proceedings of the 2nd International Conference on Data Mining in Bioinformatics, pp. 3–10 (2002)
 59. Kuramochi, M., Karypis, G.: An efficient algorithm for discovering frequent subgraphs. *IEEE Trans. Knowl. Data Eng.* **16**, 1038–1051 (2004)
 60. Thomas, L.T., Valluri, S.R., Karlapalem, K.: Margin: Maximal frequent subgraph mining. *ACM Trans. Knowl. Disc. Data (TKDD)* **4**, 10 (2010)
 61. Ghazizadeh, S., Chawathe, S.S.: SEuS: structure extraction using summaries. In: International Conference on Discovery Science, pp. 71–85 (2002)
 62. Goldman, R., Widom, J.: Dataguides: enabling query formulation and optimization in semistructured databases. Stanford (1997)
 63. Buneman, P.: Semistructured data. In: Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, pp. 117–121 (1997)
 64. Nestorov, S., Abiteboul, S., Motwani, R.: Extracting schema from semistructured data. In: ACM SIGMOD Record, pp. 295–306 (1998)
 65. Zhang, S., Yang, J., Li, S.: Ring: an integrated method for frequent representative subgraph mining. In: ICDM'09. Ninth IEEE International Conference on Data Mining, 2009, pp. 1082–1087 (2009)
 66. Ranu, S., Singh, A.K.: Graphsig: a scalable approach to mining significant subgraphs in large graph databases. In: IEEE 25th International Conference on Data Engineering, 2009. ICDE'09, pp. 844–855 (2009)
 67. He, H., Singh, A.K.: Graphrank: statistical modeling and mining of significant subgraphs in the feature space. In: Sixth International Conference on Data Mining, 2006. ICDM'06, pp. 885–890 (2006)
 68. Douar, B., Liquiere, M., Latiri, C., Slimani, Y.: LC-mine: a framework for frequent subgraph mining with local consistency techniques. *Knowl. Inf. Syst.* **44**, 1–25 (2015)
 69. Li, R., Wang, W.: REAFUM: representative approximate frequent subgraph mining. In: Proceedings of the 2015 SIAM International Conference on Data Mining, pp. 757–765 (2015)
 70. Bhuiyan, M.A., Al-Hasan, M.: An iterative MapReduce based frequent subgraph mining algorithm. *IEEE Trans. Knowl. Data Eng.* **27**, 608–620 (2015)
 71. Sanders, P., Schulz, C.: Engineering multilevel graph partitioning algorithms. In: European Symposium on Algorithms, pp. 469–480 (2011)
 72. Sanders, P., Schulz, C.: Think locally, act globally: highly balanced graph partitioning. In: International Symposium on Experimental Algorithms, pp. 164–175 (2013)
 73. Preis, R., Diekmann, R.: PARTY—a software library for graph partitioning. In: Advances in Computational Mechanics with Parallel and Distributed Processing, pp. 63–71 (1997)
 74. Muttipati, A.S., Padmaja, P.: Analysis of large graph partitioning and frequent subgraph mining on graph data. *Int. J. Adv. Res. Comput. Sci.* **6**, 29–40 (2015)
 75. Kernighan, B.W., Lin, S.: An efficient heuristic procedure for partitioning graphs. *Bell Syst. Tech. J.* **49**, 291–307 (1970)
 76. Hendrickson, B., Leland, R.W.: A multi-level algorithm for partitioning graphs. *SC* **95**, 1–14 (1995)
 77. Bichot, C.-E., Siarry, P.: Graph Partitioning. Wiley, Hoboken (2013)
 78. Agarwal, S.: Ranking on graph data. In: Proceedings of the 23rd international conference on machine learning, pp. 25–32 (2006)
 79. Saxena, A., Gera, R., Iyengar, S.: Degree ranking using local information. arXiv preprint, [arXiv:1706.01205](https://arxiv.org/abs/1706.01205) (2017)
 80. Phan, N.Q., Huynh, H.X., Guillet, F., Gras, R.: Classifying objective interestingness measures based on the tendency of value variation. In: VIII Colloque International–VIII International Conference, ASI Analyse Statistique Implicative-Statistical Implicative Analysis Radès (Tunis)–Novembre, pp. 143–172 (2015)
 81. Selvarangam, K., Kumar, K.R.: Interestingness of measures: a statistical prospective. In: 2014 International Conference on Contemporary Computing and Informatics (IC3I), pp. 209–213 (2014)
 82. Geng, L., Hamilton, H.J.: Interestingness measures for data mining: a survey. *ACM Comput. Surv. (CSUR)* **38**, 9 (2006)
 83. Zbidi, N., Faiz, S., Limam, M.: On mining summaries by objective measures of interestingness. *Mach. Learn.* **62**, 175–198 (2006)
 84. Li, X., Zhou, H., Shimada, K., Hirasawa, K.: Analysis of various interestingness measures in class association rule mining. *SICE J. Control Meas. Syst. Integr.* **4**, 295–304 (2011)
 85. Hilderman, R.J., Hamilton, H.J.: Applying objective interestingness measures in data mining systems. In: European Conference on Principles of Data Mining and Knowledge Discovery, pp. 432–439 (2000)
 86. Hussain, F., Liu, H., Suzuki, E., Lu, H.: Exception rule mining with a relative interestingness measure. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 86–97 (2000)
 87. Kuramochi, M., Karypis, G.: Finding frequent patterns in a large sparse graph. *Data Min. Knowl. Disc.* **11**, 243–271 (2005)
 88. Rehman, S.U., Asghar, S., Fong, S.: Anefficient ranking scheme for frequent subgraph patterns. In: Proceedings of the 2018 10th international conference on machinelearning and computing, pp. 257–262 (2018)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.