Automated Discovery of the Ranked Interesting Frequent Subgraph Patterns Using a Graph Mining Approach

Saif ur Rehman, Tariq Ali, Asif Nawaz, and Sohail Asghar

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

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. Graph mining has been highly motivated by the enormous number of applications. Such applications include Chemoinformatics, Bioinformatics, and societal networks. 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) Apriori-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, use the ranking for such patterns. Therefore, to address these challenges a new FSM framework A RAnked Frequent pattern-growth Framework (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, based on the characteristics of the FSPs, 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. Finally, our experiments using real and synthetic graph datasets have consistently demonstrated promising empirical results, thus confirming the superiority and practical feasibility of the proposed FSM framework.

Keywords
Social Network, Social Graph, Graph Data, Graph Mining, Graph Summarization, Graph Partition, Reconstruction Error, Big Graph.

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

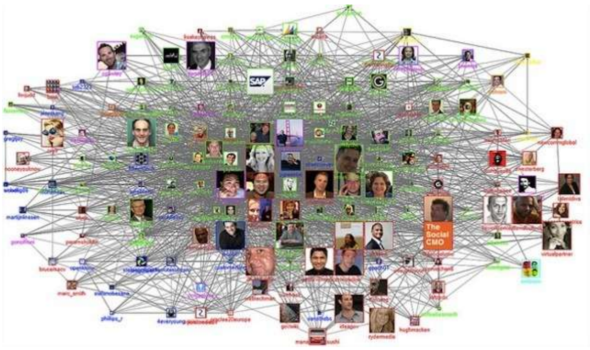


Figure 1: Facebook Friends Relationship [8]

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 benefits to various fields including; computational biology [9-10], social network analysis [5, 11], chemical data analysis [12, 13], drug discovery [14, 15] and communication networking [16, 17]. Figure 2 shows a typical graph mining process.

Frequent **S**ubgraph **M**ining (FSM) is a well-studied problem in the graph mining field and boosts several real-

Manuscript received February 5, 2026
Manuscript revised February 20, 2026
https://doi.org/10.22937/IJCSNS.2026.26.2.1

world application 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]. As Frequent Subgraph Mining (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].

In most of the literatures [39-46, 47, 48], FSM techniques are decomposed into two major categories: Apriori 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. Next, in Section 2 various state-of-the-art FSM techniques, in Apriori-based and pattern-growth based are described.

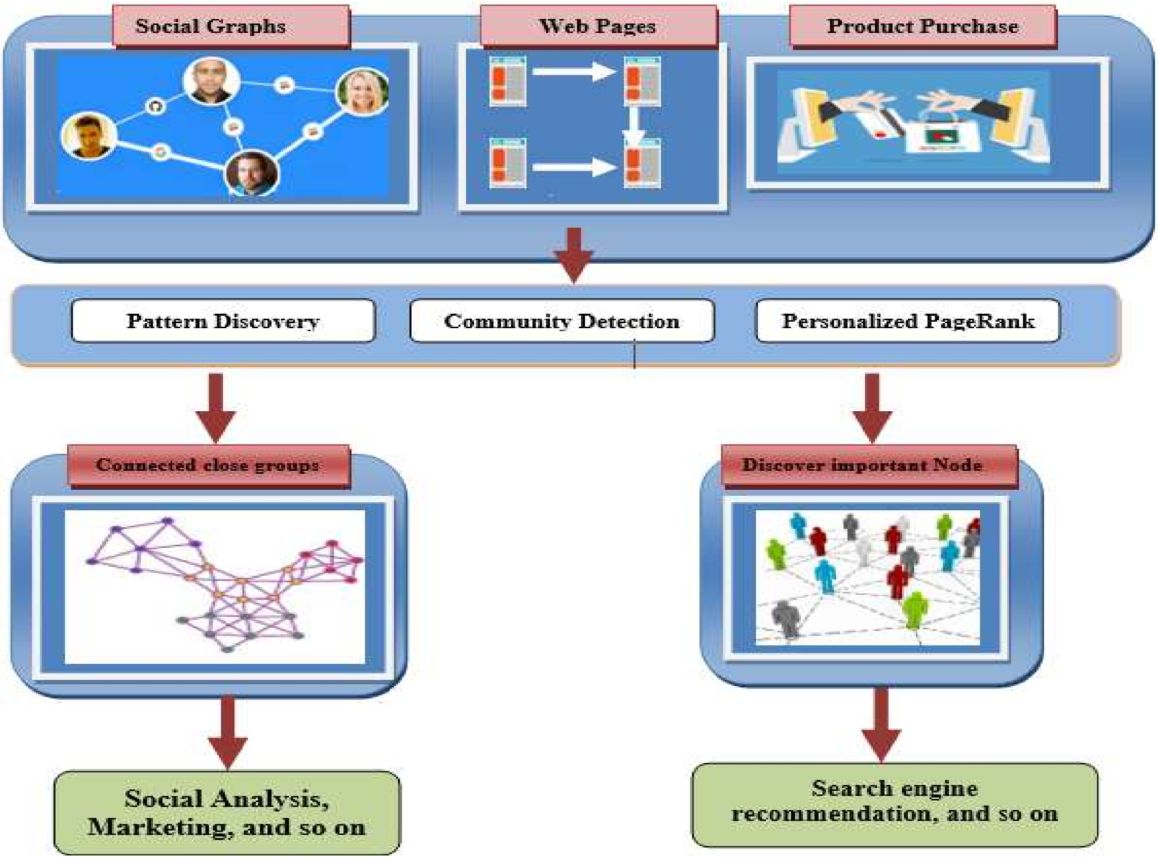


Figure 2. A Typical Graph Mining Procedure

For example, following figure gives an example of FSM. In Figure 3 a sample graph dataset is given.

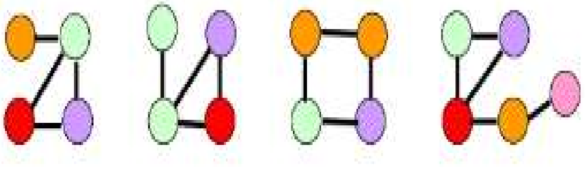


Figure 3: A Sample Graph Database

If we assumed the support threshold value as 3, then the possible frequent subgraph than can be discovered from Figure 3 are shown in Figure 4 below:

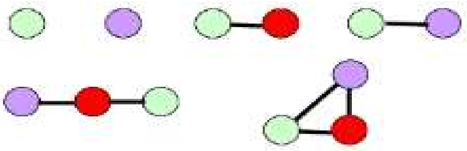


Figure 4: Sample Mined Frequent Subgraph Patterns

The primary objective of this study is to recommend a unifying framework, to reduce the enormous number of FSPs and avoid the generation of duplicate FSPs. The secondary objective is to offer a substantial comparative analysis of a number of the FSM techniques with A-RAFF. The major contribution of this study may thus be summarized as follows:

(i) The proposal of a new unifying framework A RAnked Frequent pattern-growth Framework (A-RAFF) from transaction graph datasets. A-RAFF proved to be very handy in reducing the enormous frequent subgraph patterns.
(ii) The proposal of new graph ranking measure, FSP-Rank, for addressing the ranking problem of FSPs. The proposed FSP-Rank is used as a pre-processing step in the frequent subgraph pattern discovery procedure.
(iii) Finally, comprehensive performance evaluation of the proposed FSM framework, A-RAFF, using four benchmark graph datasets. The proposed A-RAFF framework outperforms the gSpan, FSP, FFSM, and some other FSM method.
(iv) İnvestigating on how to compute the interestingness of the discovered FSPs.
(v) Comparison of the FSP-Rank with the GraphRank (Which is proposed for FSPs ranking)

The remaining paper is organized as follows The related work on FSPs discovery approaches has been discussed in Section 2. We present our novel FSPs ranking framework, A-RAFF, in Section 4, where the fundamental principle of the approach is presented. In Section 4, we report the experimental results obtained from the simulations of the proposed approach on various graph networks. Finally, in Section 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 Frequent Subgraph Mining (FSM) in transaction-based graph datasets. Different representative techniques from the two categories, Apriori-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.

The SPIN (**SPanning tree based maximal graph mINing)** was the first approach to mix two distinctive techniques: discovery frequent subgraph patterns 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 a 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 (4) In some applications such as extracting the structure motifs in a group of homology proteins [58], MFS are the subgraph patterns of most significance as they encode the maximal structure commonalities within the group. SPIN algorithm performance was shown using different experiments, which presented the effectiveness of SPIN in that it exponentially minimize the number of frequent subgraph patterns discovered. Also, SPIN showed very good scalability to big graph datasets. In massive graph datasets, discovered frequent subgraph patterns can become too bigger to allow an entire enumeration by realistic computational resources.

MARGIN is another Apriori-based FSG approach to mine the maximal frequent subgraph pattern. The set of Maximal Frequent Subgraphs (MFS) is significantly smaller than the set of frequent subgraphs [53], therefore, provided the scope for sufficient pruning of the exponentially massive search space. MARGIN is based on the observation of those FSGs which are potential maximal are included in k-subgraphs set which is frequent having $(k+1)$ -infrequent super-graphs. MARGIN algorithm recursively discovers the candidates by using core procedure, *ExpandCut*, and the set of maximal frequent subgraph patterns 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, but 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 were proposed [59]. GREW is specially designed and developed to execute on a large graph datasets 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 frequent subgraph mining algorithms for time series of graphs, particularly, dynamic GREW focused on the subgraph patterns which are topologically frequent within a big graph. The adjacency matrix graph representation was adopted the GREW. The limitation of this technique is an extra overhead in identifying dynamic patterns. Moreover, the authors claimed that their framework support to integrate the existing subgraph mining algorithms in order to make them handle dynamic graphs. Different experiments were performed to check practical feasibility of the dynamic GREW , results on different real-world data validated the proposed technique for the dynamic graph scenarios .

In SeuS (**Structure Extraction using Summaries)**, 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]. SEuS can be used with large connected graph dataset as well as graph transnational datasets. Furthermore, SEuS defeated the inherent computational complexity of the problem by using a summarized data structure to trim the search space and to supply interactive feedback.

The gSpan **(graph-based Substructure pattern)** gSpan uses DFS searching strategy and is the fastest PG based FSM approach [38]. DFS-Code is used for canonical representation of the graph dataset, while the vertex of the graph are traversed using DFS. The concatenation of graph edge representations in an order is called the graph DFS-Code. 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, but gSpan generated a large number of FSPs which become difficult to analyze [40, 55, 48].

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, in which real-word benchmark graph datasets were used, CloseGraph showed very good results. It avoided the generation of undesired subgraph patterns during the subgraph generation phase. Also, 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]. FFSM adopted the maximal code to explore isomorphism tests by keeping an embedding set for each FSP, thus circumvent the subgraph isomorphism testing cost [47]. FFSM performance was assessed using different benchmark as well as synthetic graph datasets. In experiments, results reported that FFSM outperform 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 appraoch, that suggested an important enhancement in the PG based FSM category[49]. Following are the potential benefits, which were introduced to PG by FSP; (1) improved the graph canonical representation and defined 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 showed 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 examine. Experiments on real datasets, Gaston performance was very good. Since Gaston focuses on maximal FSGs, so 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 centre of the clusters (groups) as representative subgraph patterns 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.

İn a recent study on FSM, GraphSig is proposed to discover the discriminative subgraph patterns with low frequencies, GraphSig was proposed in[646. GraphSig is a scalable approach, which extract statistically significant patterns from a large collection of graphs. GraphSig is able to discover the discriminating patterns in a big graph databases and still with small values of the frequencies. There is need to decide how to decide the statistical significance rather than assuming manually and significant graph feature selection need to be improved further.

GraphRank[67] 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 occur 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 calculate 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. Also, the problem of graph feature vector mining was addressed in GraphRank. 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 [70], 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. They proposed two arc consistency-based instances for the framework of LC- mine. FGMAC and AC-miner are two arc consistency-based instances for frequent subgraph discovery, details can be found in [70].

FSM-H [71] is recently proposed distributed frequent subgraph mining 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 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 partition. 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 [71].

3. A RAnked Frequent Subgraph Discovery Framework (A-RAFF)

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

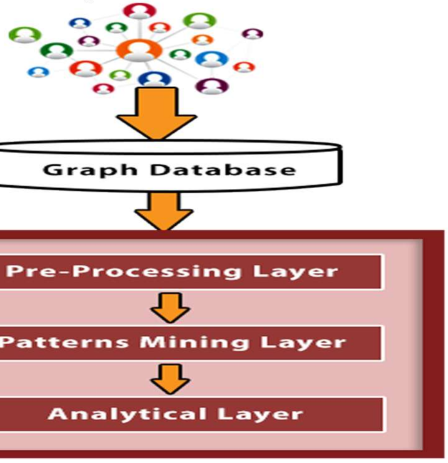


Figure 5: A-RAFF-A Proposed Layered Framework

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 with the labels on the edges. In the subsequent section, each of the three layers is described one by one. The fundamental aim of the pre-processing layer is to prepare the graph dataset for the graph pattern mining layer. In pre-processing layer, graph dataset is clean-up, i.e., the graphs with isolated or disconnected nodes are detected and 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 pre-processing layer are: graph dataset features selection and partitioning of the graph dataset.

Proposed Algorithm 1: A-RAFF ($\mathbb{GD}, \sigma, \mathcal{RF}$)

```
**Input:** $\mathbb{GD} =$ a graphs dataset of the labelled undirected graphs
σ= minimum threshold
**Output:** $\mathcal{RF}$ , a set of ranked frequent subgraph patterns
count all the features and put in the feature set **"F "**
**for each** feature ***f*** in the **F**-set **do**
    Identify most informative graph features
r**epeat** until all features are checked
partition the graph using KaHIP Tool based on an **F**-set [80]
$\mathcal{F} \leftarrow \emptyset$ $\mathcal{F}$ denotes the frequent subgraph patterns set
$\mathcal{F}$ ←discovered 1-frequent subgraphs patterns
**for each** graph $g_{i}\_partition \in G_{i}$ **do**
        FSPs($\mathbb{GD}, g_{i}\_partition, \sigma, \mathcal{F}$)
**end**
FSP-Rank($\mathcal{F}$)
**return** $\boldsymbol{\mathcal{RF}}$
```

Figure 6: The Proposed Algorithm for A-RAFF

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 [85, 186]. The second major function of the pre-processing 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 literature, various graphs partitioning techniques and tools are available [72-178]. In the proposed framework A-RAFF, the graph partitioning is performed using a well-known graph partitioning tool KaHIP [72]. 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 strategies transferred from multi-grid linear solvers problem [73]. Finally, the graph partitions are forwarded to the pattern mining layer.

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 pre-processed graph structures from the graph pre-processing 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 to objects and an edge between objects depict the similarities [79]. Very limited research work is available in ranking on graph structure data such as in search engine [76, 78]. In graph mining domain, mainly research community is interested in the relative importance of the graph nodes with the top ranked node. In [80], 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 ($\mathbb{GD}, g_i\_partition, \sigma, \mathcal{F}$)
Input: $\mathbb{GD}$ = 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" $\in \mathbb{GD}$ such that
7.  $extended\_g_i \leftarrow g_i\_partition \cup e$
8.  foreach $extended\_g_i$ from Line. (7) do
9.          if support $(extended\_g_i) \geq \sigma\ |\mathbb{GD}|$ then
                FSPs($\mathbb{GD}, extended\_g_i, \sigma, \mathcal{F}$)
10. else
                return
11. end
12. return $\mathcal{F}$
```

Figure 7: Propsoed Algorithm for FSPs Discovery

In the proposed A-RAFF framework, once the FSPs are discovered, the next step is to rank the extracted patterns. For FSPs ranking, a new ranking measure, called FSP-Rank, 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 value 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. The proposed ranking algorithm, FSP-Rank, is presented in Figure 8.

The rank value for the FSP is computed using the following equation,

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

```
Proposed Algorithm 5.3: FSP-Rank(FSPs, $\mathcal{RF}$)
    Input: FSPs = frequent subgraph patterns
    Output: $\mathcal{RF}$, set of rank frequent subgraph patterns
    Compute $\boldsymbol{\lambda}$ using Equation (2)
    for each FSP in FSPs do
      Score ($D_i$) ← $\sum_{i=0}^{n} FSP_i$ (in − degree + out − degree)
    Compute $f(R_k)$ using Equation (1)
                    $\mathcal{RF} \leftarrow \mathcal{RF} \cup$ FSP
    next
    return $\mathcal{RF}$
```

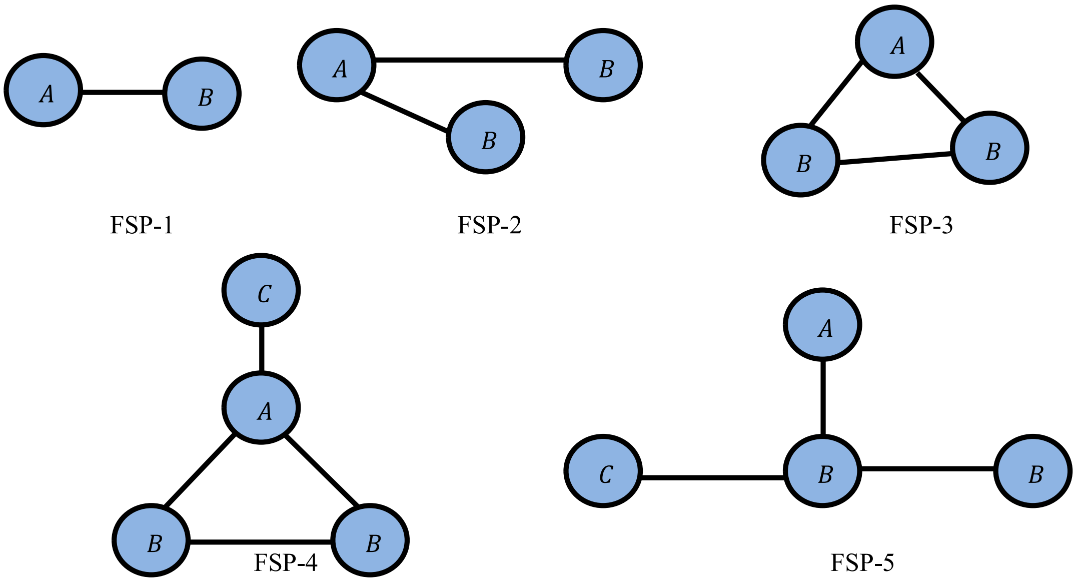
Figure 8: Proposed Ranking Algorithm: FSP-Rank

Where, λ is a normalized factor and is calculated from the equation 5.3. W represents the total of the weight of the 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 $(0 \leq f(R_k) \leq 1)$. The value of the normalization factor, λ, is computed using the equation as follows,

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

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 equation 1. Furthermore, an example is discussed here, to have in depth understanding of the working on the concepts of the FSPs ranking described by the equations 1 and 2.

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,



FSP-1

FSP-2

FSP-3

FSP-4

FSP-5

Figure 9: Sample Extracted FSPs

In Figure 9, 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 FSPs and degree score, the normalization factor is computed. The value of normalization factor is computed using above Equation 2, and is shown in column 4 of Table 1.

$\lambda = \left(\frac{\sum_{i=1}^{n}(FSP)}{\sum_{i=1}^{n}T(V_i)}\right)$ = (Total FSPs / # of vertices in all the FSPs) = 5/16 = 0.3125

Table 1: Computation of $\boldsymbol{\lambda}$Using FSPs Characteristics

FSPs	$\mathbf{D_i}$ = Degree (In-Deg + Out-Deg)	n_i = number of Vertices	$\boldsymbol{\lambda}$ = Normalization Factor	$\left(\frac{\boldsymbol{D_i}}{\boldsymbol{n_i}}\right)$	$\boldsymbol{\lambda} * \left(\frac{\boldsymbol{D_i}}{\boldsymbol{n_i}}\right)$
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

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.

Table 2: Computation of the Final Rank Score

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

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.

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

The ranked values for the FSPs are further used to identify the duplicate frequent subgraph patterns. All the frequent graph patterns having the same ranked score were analysed 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 frequent patterns are generated solely based on frequency, not every frequent pattern is equally significant and interesting as well [81-84]. Foregoing any difficulties in defining a measure which correctly identifies what we discover interesting, the second key dilemma is the exponentially outsized search space.

Table 4: 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

That is, there are exponentially 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 [85]. 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 [85-88]. In the next section, we have used three different measures of intrestingness including I_{RAE}, I_{CON} and I_{Variance} to validate the discovered intrestingness of the FSPs discovered by the proposed FSM framework, A-RAFF. In our opinion, such measures were not incorporated in the FSM domain upto 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 pre-processing 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 pre-processing 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. 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 **σ** = 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 [55, 52, 53, 54, 132].

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. 80% 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 experiments, A-RAFF outperformed the other FSM approaches under consideration. We have summarized the performance evaluation in the following tables, from Table 4 to Table 8, which is giving a clearer picture of the proposed A-RAFF performance with the existing FSM techniques. In tabular results from Table 4 to Table 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 Table 4 to Table 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 again 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 tool perform well than the A-RAFF. A-RAFF took 11 and 12 seconds more than the gSpan and Gaston respectively to discover the FSPs at 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.

5. 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 was discussed in details in Section 2, Related Work). GraphRank defined the statistical significance as, the probability that a graph g occur in the graph dataset of random graphs with the value of support$\boldsymbol{\mu} \geq \boldsymbol{\mu_0}$, which is termed as $\boldsymbol{p - value}$ of the graph g [67].

We have compared the proposed FSP-Rank with GraphRank measure and investigated the effectiveness of FSP-Rank measure over the GraphRank. In the proposed ranking measure, FSP-Rank, we have used the different characteristics of the mined frequent subgraph patterns including the degree values of the nodes, total nodes in a specific FSP, weight of the nodes and edges. Using the proposed 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 compare 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 proposed A-RAFF framework and FSP-Rank respectively. In Table 11, first the FSPs were extracted using the proposed A-RAFF framework and then these extracted FSPs were ranked using the proposed ranking algorithm FSP-Rank. Next, using the CloseGraph FSM technique [52], all the Closed-FSPs were extracted and then using the GraphRank, the closed-FSPs were ranked based on their statistical significance. The closed-FSPs were ranked using the significance value of 0.6 (i.e., $p - value$ = 0.6). Table 11 shows the detailed results.

Table 5: 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

In Table 5, we have shown the time required by FSP-Rank and the GraphRank to rank the discovered closed-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 seconds to rank the discovered FSPs. Only at $\sigma = 30$, GraphRank performed better than the proposed FSP-Rank.

Measuring the Interestingness of the discovered FSPs: Finally, we have evaluated the interestingness of the FSPs discovered using the proposed FSM framework A-RAFF. Interestingness measures play an important role in KDD, regardless of the kind of patterns being mined [203]. All the patterns mined are not interesting or whatever the pattern mined by data mining tools are not interesting [208]. To analyze the interestingness of a rule set, various interestingness measures (IS Measures) are proposed and analyzed by the researchers. IS measures are generally divided into two main categories of objective and subjective measures of interest [203, 209-211]. An objective measure uses the raw data and no knowledge about the user or application is needed. Most 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 [204, 212]. The objective measures 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 [203, 207, 213]. 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 if a pattern is "actionable" and/or "unexpected". An unexpectedness, actionable, novel are criteria under subjective nature.

In this study, we have evaluated the interestingness of the discovered FSPs using the objective measure of interestingness. To compute the interestingness of the discovered FSPs we used two different objective interestingness measure, called the I_{RAE} measure, originally proposed by Rae and Taylor in [214] and second measure is $\mathrm{I_{CON}}$ proposed by Egghe and Rousseau in [215]. We selected these three measures of interestingness as these are simple to compute and relevant to the work presented in this paper. The I_{RAE} , $\mathrm{I_{CON}}$ and $\mathrm{I_{Variance}}$ interestingness measures are computed using the following equation 3, 4 and 5 respectively,

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

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

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

Where, n_i shows the number of FSPs discovered by the A-RAFF framework; 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 5 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 $(i = 1, 2, 3, .., m)$.

We have applied the I_{RAE} , $\mathrm{I_{CON}}$ and $\mathrm{I_{Variance}}$ interestingness measures on the FSPs extracted using A-RAFF which were shown in Table 9 and 10. These measures have been applied to the discovered FSPs in order to measure the discovered patterns diversity criteria. The I_{RAE} , $\mathrm{I_{CON}}$ and $\mathrm{I_{Variance}}$ Interestingness measure values computed from these tabular FSPs results are given in Table 12.

Table 6: 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

Table 6 shows the different interestingness measure value against each graph dataset from 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, 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 by the I_{CON} interestingness on all the considered graph datasets. The results described in Table 13 give the indication the FSPs generated from the proposed FSM framework A-RAFF are very diverse.

6. Conclusion and Future Work

The proposed conceptual and architectural framework called A RAnked Frequent pattern-growth Framework (A-RAFF) was discussed. The three layers of the conceptual framework: graph pre-processing layer; graph pattern mining layer and analytical layer were described. Furthermore, the subcomponents of the architectural framework along with the proposed algorithms were also presented. The FSPs ranking scheme and its significance were also highlighted in this chapter. Further, the proposed A-RAFF framework is experimentally investigated using a diverse set of real-world benchmark and synthetic graph datasets. We also compared the FSP-Rank with the existing ranking measure GraphRank. Different experiments show the prominence of the A-RAFF, as depicted in Table 4 through Table 8 with respect to performance comparisons. Maximum time reduced by A-RAFF for extraction of FSPs is 611 (Sec.) on the AIDS Antiviral Screen Compound, Table 4. An exhaustive analysis of A-RAFF with other FSM approaches with respect to number of FSPs extracted given in Table 9 and 10, clearly highlighting that A-RAFF has significantly reduced the final set of FSPs. Finally, we have used different interestingness measures to show the importance of the discovered FSPs. Although this research study reached to its objectives, there were few unavoidable weaknesses are observed which are given as follows. First, the findings of this study are restricted on the discovery of the frequent subgraph patterns 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, so 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. The objective of the researchers is to deliver the practical solutions and improvement of frequent subgraph patterns discovery. This study is also an effort to produce the practical solution of enormous frequent subgraph patterns discovered. However, the improvements are never ending task in research and following are the few future guidelines that would be helpful to further enhance the research. There is lots of room to enhance the ranking scheme introduced in this work.

References

[1] J. Kim and M. Hastak, "Social network analysis," *International Journal of Information Management: The Journal for Information Professionals,* vol. 38, pp. 86-96, 2018.

[2] S. S. Sonawane and P. A. Kulkarni, "Graph based representation and analysis of text document: A survey of techniques," *International Journal of Computer Applications,* vol. 96, 2014.

[3] T. Maugey, A. Ortega, and P. Frossard, "Graph-based representation for multiview image geometry," *IEEE Transactions on Image Processing,* vol. 24, pp. 1573-1586, 2015.

[4] R. Reichle, M. Gaul, S. Nicklis, C. Hornung, D. Nissel, S. Schneider*, et al.*, "Navigation apparatus and method for displaying a navigation tree on a display unit," ed: Google Patents, 2017.

[5] S. Freedman and G. Z. Jin, "The information value of online social networks: lessons from peer-to-peer lending," *International Journal of Industrial Organization,* vol. 51, pp. 185-222, 2017.

[6] M. Jalili, Y. Orouskhani, M. Asgari, N. Alipourfard, and M. Perc, "Link prediction in multiplex online social networks," *Royal Society open science,* vol. 4, p. 160863, 2017.

[7] S. Agrebi and J. Jallais, "Explain the intention to use smartphones for mobile shopping," *Journal of Retailing and Consumer Services,* vol. 22, pp. 16-23, 2015.

[8] I. Atastina, B. Sitohang, G. Saptawati, and V. Moertini, "A Review of Big Graph Mining Research," in *IOP Conference Series: Materials Science and Engineering*, 2017, p. 012065.

[9] I. Atastina, B. Sitohang, G. Saptawati, and V. Moertini, "A Review of Big Graph Mining Research," in *IOP Conference Series: Materials Science and Engineering*, 2017, p. 012065.

[10] M. Koyutürk, A. Grama, and W. Szpankowski, "An efficient algorithm for detecting frequent subgraphs in biological networks," *Bioinformatics,* vol. 20, pp. i200-i207, 2004.

[11] S. A. Moosavi, M. Jalali, N. Misaghian, S. Shamshirband, and M. H. Anisi, "Community detection in social networks using user frequent pattern mining," *Knowledge and Information Systems,* vol. 51, pp. 159-186, 2017.

[12] A. Prado, M. Plantevit, C. Robardet, and J.-F. Boulicaut, "Mining graph topological patterns: Finding covariations among vertex descriptors," *IEEE Transactions on Knowledge and Data Engineering,* vol. 25, pp. 2090-2104, 2013.

[13] X. Yan, F. Zhu, J. Han, and P. S. Yu, "Searching substructures with superimposed distance," in *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*, 2006, pp. 88-88.

[14] P. Csermely, T. Korcsmáros, H. J. Kiss, G. London, and R. Nussinov, "Structure and dynamics of molecular networks: a novel paradigm of drug discovery: a comprehensive review," *Pharmacology & therapeutics,* vol. 138, pp. 333-408, 2013.

[15] I. Takigawa and H. Mamitsuka, "Graph mining: procedure, application to drug discovery and recent advances," *Drug discovery today,* vol. 18, pp. 50-57, 2013.

[16] X. Zhang, T. Ouyang, D. Pan, X. Si, and S. Rahman, "Upstream pilot structure in point to multipoint orthogonal frequency division multiplexing communication system," ed: Google Patents, 2016.

[17] H. Xiao, Y. Hu, K. Yan, and S. Ouyang, "Power Allocation and Relay Selection for Multisource Multirelay Cooperative Vehicular Networks," *IEEE Transactions on Intelligent Transportation Systems,* vol. 17, pp. 3297-3305, 2016.

[18] M. Deshpande, M. Kuramochi, N. Wale, and G. Karypis, "Frequent substructure-based approaches for classifying chemical compounds," *IEEE Transactions on Knowledge and Data Engineering,* vol. 17, pp. 1036-1050, 2005.

[19] J. Huan, W. Wang, D. Bandyopadhyay, J. Snoeyink, J. Prins, and A. Tropsha, "Mining protein family specific residue packing patterns from protein structure graphs," in *Proceedings of the eighth annual international conference on Resaerch in computational molecular biology*, 2004, pp. 308-315.

[20] C. Borgelt and M. R. Berthold, "Mining molecular fragments: Finding relevant substructures of molecules," in *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, 2002, pp. 51-58.

[21] F. Eichinger, K. Böhm, and M. Huber, "Mining edge-weighted call graphs to localise software bugs," in *Joint European Conference on Machine*

Learning and Knowledge Discovery in Databases, 2008, pp. 333-348.

[22] C. Liu, X. Yan, H. Yu, J. Han, and P. S. Yu, "Mining behavior graphs for “backtrace” of noncrashing bugs," in *Proceedings of the 2005 SIAM International Conference on Data Mining*, 2005, pp. 286-297.

[23] Y. Liu, S. Xu, and L. Duan, "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*, 2014, pp. 1649-1658.

[24] B. Panda, S. N. Tripathy, N. Sethi, and O. P. Samantray, "A comparative study on serial and parallel web content mining," *International Journal of Advanced Networking and Applications,* vol. 7, p. 2882, 2016.

[25] S. P. Algur and P. Bhat, "Web Video Object Mining: Expectation Maximization and Density Based Clustering of Web Video Metadata Objects," *International Journal of Information Engineering and Electronic Business,* vol. 8, p. 69, 2016.

[26] R. Baeza-Yates and P. Boldi, "Web structure mining," in *Advanced Techniques in Web Intelligence-I*, ed: Springer, 2010, pp. 113-142.

[27] H.-H. Shuai, C.-Y. Shen, D.-N. Yang, Y.-F. Lan, W.-C. Lee, P. S. Yu*, et al.*, "Mining online social data for detecting social network mental disorders," in *Proceedings of the 25th International Conference on World Wide Web*, 2016, pp. 275-285.

[28] S. N. Chakradeo, R. M. Abraham, B. A. Rani, and R. Manjula, "Data mining: Building social network," *Indian Journal of Science and Technology,* vol. 8, pp. 212-216, 2015.

[29] F. Jiang, K. Kawagoe, and C. K. Leung, "Big Social Network Mining for Following Patterns," in *Proceedings of the Eighth International C* Conference on Computer Science & Software Engineering*, 2015, pp. 28-37.

[30] P. SCHOLER, "Detection of Fraud Ranking for Mobile App Using Fuzzy Logic."

[31] I. Alsmadi and I. Alhami, "Clustering and classification of email contents," *Journal of King Saud University-Computer and Information Sciences,* vol. 27, pp. 46-57, 2015.

[32] G. Tang, J. Pei, and W.-S. Luk, "Email mining: tasks, common techniques, and tools," *Knowledge and Information Systems,* vol. 41, pp. 1-31, 2014.

[33] M. Aery and S. Chakravarthy, "InfoSift: Adapting Graph Mining Techniques for Text Classification," in *FLAIRS Conference*, 2005, pp. 277-282.

[34] W. Eberle and L. Holder, "Discovering structural anomalies in graph-based data," in *Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on*, 2007, pp. 393-398.

[35] C. R. Palmer, P. B. Gibbons, and C. Faloutsos, "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*, 2002, pp. 81-90.

[36] N. Vanetik, E. Gudes, and S. E. Shimony, "Computing frequent graph patterns from semistructured data," in *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, 2002, pp. 458-465.

[37] N. Vanetik and E. Gudes, "Mining frequent labeled and partially labeled graph patterns," in *Data Engineering, 2004. Proceedings. 20th International Conference on*, 2004, pp. 91-102.

[38] X. Yan and J. Han, "gspan: Graph-based substructure pattern mining," in *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, 2002, pp. 721-724.

[39] T. Ramraj and R. Prabhakar, "Frequent Subgraph Mining Algorithms–A Survey," *Procedia Computer Science,* vol. 47, pp. 197-204, 2015.

[40] H. J. Patel, R. Prajapati, M. Panchal, and M. Patel, "A Survey of Graph Pattern Mining Algorithm and Techniques," *International Journal of Application or Innovation in Engineering & Management (IJAIEM),* vol. 2, 2013.

[41] K. Lakshmi, "Frequent subgraph mining algorithms--a survey and framework for classification," 2012.

[42] C. Jiang, F. Coenen, and M. Zito, "A survey of frequent subgraph mining algorithms," *The Knowledge Engineering Review,* vol. 28, pp. 75-105, 2013.

[43] A. Dhiman and S. Jain, "Frequent subgraph mining algorithms for single large graphs—A brief survey," in *Advances in Computing, Communication, & Automation (ICACCA)(Spring), International Conference on*, 2016, pp. 1-6.

[44] S. Thomas and J. J. Nair, "A survey on extracting frequent subgraphs," in *Advances in Computing, Communications and Informatics (ICACCI), 2016 International Conference on*, 2016, pp. 2290-2295.

[45] S. Santhi and P. Padmaja, "A survey of frequent subgraph mining algorithms for uncertain graph

data," *Int. Res. J. Eng. Technol.(IRJET),* vol. 2, pp. 688-696, 2015.

[46] H. Naderi, "A Survey of Frequent Subtrees and Subgraphs Mining Methods," *International Journal of Computer Science and Business Informatics,* vol. 14, 2014.

[47] C. Jiang, F. Coenen, and M. Zito, "A survey of frequent subgraph mining algorithms," *The Knowledge Engineering Review,* vol. 28, pp. 75-105, 2013.

[48] V. Krishna, N. R. Suri, and G. Athithan, "A comparative survey of algorithms for frequent subgraph discovery," *Current Science,* pp. 190-198, 2011.

[49] S. Han, W. K. Ng, and Y. Yu, "Fsp: Frequent substructure pattern mining," in *Information, Communications & Signal Processing, 2007 6th International Conference on*, 2007, pp. 1-5.

[50] J. Huan, W. Wang, and J. Prins, "Efficient mining of frequent subgraphs in the presence of isomorphism," in *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, 2003, pp. 549-552.

[51] M. Kuramochi and G. Karypis, "Frequent subgraph discovery," in *Data Mining, 2001. ICDM 2001, Proceedings IEEE international conference on*, 2001, pp. 313-320.

[52] X. Yan and J. Han, "CloseGraph: mining closed frequent graph patterns," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003, pp. 286-295.

[53] J. Huan, W. Wang, J. Prins, and J. Yang, "Spin: mining maximal frequent subgraphs from graph databases," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004, pp. 581-586.

[54] S. Nijssen and J. N. Kok, "The gaston tool for frequent subgraph mining," *Electronic Notes in Theoretical Computer Science,* vol. 127, pp. 77-87, 2005.

[55] L. B. Holder, D. J. Cook, and S. Djoko, "Substucture Discovery in the SUBDUE System," in *KDD workshop*, 1994, pp. 169-180.

[56] T. S. Mumu, "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] J. Hu, X. Shen, Y. Shao, C. Bystroff, and M. J. Zaki, "Mining protein contact maps," in *Proceedings of the 2nd International Conference on Data Mining in Bioinformatics*, 2002, pp. 3-10.

[59] M. Kuramochi and G. Karypis, "An efficient algorithm for discovering frequent subgraphs," *IEEE Transactions on Knowledge and Data Engineering,* vol. 16, pp. 1038-1051, 2004.

[60] L. T. Thomas, S. R. Valluri, and K. Karlapalem, "Margin: Maximal frequent subgraph mining," *ACM Transactions on Knowledge Discovery from Data (TKDD),* vol. 4, p. 10, 2010

[61] S. Ghazizadeh and S. S. Chawathe, "SEuS: Structure extraction using summaries," in *International Conference on Discovery Science*, 2002, pp. 71-85.

[62] R. Goldman and J. Widom, "Dataguides: Enabling query formulation and optimization in semistructured databases," Stanford1997.

[63] P. Buneman, "Semistructured data," in *Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, 1997, pp. 117-121.

[64] S. Nestorov, S. Abiteboul, and R. Motwani, "Extracting schema from semistructured data," in *ACM SIGMOD Record*, 1998, pp. 295-306.

[65] S. Zhang, J. Yang, and S. Li, "Ring: An integrated method for frequent representative subgraph mining," in *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on*, 2009, pp. 1082-1087.

[66] S. Ranu and A. K. Singh, "Graphsig: A scalable approach to mining significant subgraphs in large graph databases," in *Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on*, 2009, pp. 844-855.

[67] H. He and A. K. Singh, "Graphrank: Statistical modeling and mining of significant subgraphs in the feature space," in *Data Mining, 2006. ICDM'06. Sixth International Conference on*, 2006, pp. 885-890.

[68] B. Douar, M. Liquiere, C. Latiri, and Y. Slimani, "LC-mine: a framework for frequent subgraph mining with local consistency techniques," *Knowledge and Information Systems,* vol. 44, pp. 1-25, 2015.

[69] R. Li and W. Wang, "REAFUM: Representative Approximate Frequent Subgraph Mining," in *Proceedings of the 2015 SIAM International Conference on Data Mining*, 2015, pp. 757-765.

[70] B. Douar, M. Liquiere, C. Latiri, and Y. Slimani, "LC-mine: a framework for frequent subgraph mining with local consistency techniques,"

Knowledge and Information Systems, vol. 44, pp. 1-25, 2015.

[71] M. A. Bhuiyan and M. Al Hasan, "An iterative MapReduce based frequent subgraph mining algorithm," *IEEE Transactions on Knowledge and Data Engineering,* vol. 27, pp. 608-620, 2015.

[72] P. Sanders and C. Schulz, "Engineering multilevel graph partitioning algorithms," in *European Symposium on Algorithms*, 2011, pp. 469-480.

[73] P. Sanders and C. Schulz, "Think locally, act globally: Highly balanced graph partitioning," in *International Symposium on Experimental Algorithms*, 2013, pp. 164-175.

[74] R. Preis and R. Diekmann, "PARTY-a software library for graph partitioning," *Advances in Computational Mechanics with Parallel and Distributed Processing,* pp. 63-71, 1997.

[75] A. S. Muttipati and P. Padmaja, "Analysis of Large Graph Partitioning and Frequent Subgraph Mining on Graph Data," *International Journal of Advanced Research in Computer Science,* vol. 6, 2015.

[76] B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *The Bell system technical journal,* vol. 49, pp. 291-307, 1970.

[77] B. Hendrickson and R. W. Leland, "A Multi-Level Algorithm For Partitioning Graphs," *SC,* vol. 95, pp. 1-14, 1995.

[78] C.-E. Bichot and P. Siarry, *Graph partitioning*: John Wiley & Sons, 2013.

[79] S. Agarwal, "Ranking on graph data," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 25-32.

[80] A. Saxena, R. Gera, and S. Iyengar, "Degree Ranking Using Local Information," *arXiv preprint arXiv:1706.01205,* 2017.

[81] N. Q. Phan, H. X. Huynh, F. Guillet, and R. Gras, "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 (Tunisie)-Novembre*, 2015, pp. 143-172.

[82] K. Selvarangam and K. R. Kumar, "Interestingness of measures: A statistical prospective," in *Contemporary Computing and Informatics (IC3I), 2014 International Conference on*, 2014, pp. 209-213.

[83] L. Geng and H. J. Hamilton, "Interestingness measures for data mining: A survey," *ACM Computing Surveys (CSUR),* vol. 38, p. 9, 2006.

[84] N. Zbidi, S. Faiz, and M. Limam, "On mining summaries by objective measures of interestingness," *Machine learning,* vol. 62, pp. 175-198, 2006.

[85] X. Li, H. ZHOU, K. SHIMADA, and K. HIRASAWA, "Analysis of various interestingness measures in class association rule mining," *SICE Journal of Control, Measurement, and System Integration,* vol. 4, pp. 295-304, 2011.

[86] R. J. Hilderman and H. J. Hamilton, "Applying objective interestingness measures in data mining systems," in *European Conference on Principles of Data Mining and Knowledge Discovery*, 2000, pp. 432-439.

[87] F. Hussain, H. Liu, E. Suzuki, and H. Lu, "Exception rule mining with a relative interestingness measure," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2000, pp. 86-97.

[88] M. Kuramochi and G. Karypis, "Finding frequent patterns in a large sparse graph," *Data mining and knowledge discovery,* vol. 11, pp. 243-271, 2005.