An Enhancement of Grami Based on Threshold Policy for Pattern Big Graphs

Shriya Sahu, Meenu Chawla, Nilay Khare

Abstract: Pattern Mining is the key mechanism to manage large scale data element. Frequent subgraph mining (FSM) considers isomorphism which is a subprocess of pattern mining is a well-studied problem in the data mining. Graphs are considered as a standard structure in many domains such as protein-protein interaction network in biological networks, wired or wireless interconnection networks, web data, etc. FSM is the task of finding all frequent subgraphs from a given database i.e. a single big graph or database of many graphs, whose support is greater than the given threshold value. Many databases consider small graphs for solving complex problems. The classification of graph depends upon the application requirement. A good mining architecture may prevent a lot of memory and time. This paper follows the Grami structure for the analysis of frequent subgraph mining and also introduces the 20% threshold policy for the enhancement of the directed pattern graphs. The constraint satisfaction problem (CSP) has been discussed and analyzed using the Grami approach. The proposed model is compared to Grami on twitter dataset based on the evaluation of time and memory consumed. The proposed algorithm shows an improvement of 3-4 % for both the parameters. The results show that the performance of Grami approach has been improved which shows a 6.6% reduction in time and 21% improvement in memory consumption using the proposed approach.

Keywords: Big graphs, Grami,Pattern Mining, Subgraph Mining

I. INTRODUCTION

There is a complex relationship among objects which is determined using direct and undirected graphs. Such relation is used for various applications such as web analysis, text retrieval, bioinformatics, chemical, social networks, and computer vision. The issues can be modeled using directed and un-directed graphs[1]. Frequent subgraph mining is a well-studied problem in graphs which is solved using directed graphs.Mining Frequent Pattern (MFP) plays a crucial role in various applications such as modeling profiles, graph classification, index selection, graph clustering, and designing database [2].There is a great use of directed graphs in MFP.

Revised Manuscript Received on January 30, 2020.
* Correspondence Author

Shriya Sahu*, Department of Computer Science & Engineering, Maulana Azad National Institute of Technology, Bhopal (M.P.) India. E-mail: s.shriya88@gmail.com

Dr. Meenu Chawla, Department of Computer Science & Engineering, Maulana Azad National Institute of Technology, Bhopal (M.P.) India. E-mail: meenuchawlamanit@gmail.com

Dr. Nilay Khare, Department of Computer Science & Engineering, Maulana Azad National Institute of Technology, Bhopal (M.P.) India. E-mail: nilay.khare@rediffmail.com

The main objective of frequent mining is to determine subgraphs whose appearances exceeds a certain threshold level. The obtained level also applicable to determine the edge values for the uncertain graphs [3]. The obtained value is useful in many real-life problems. For instance, Chao defines a set of functions which is performed by the protein association network to assign a label to the node [4]. The graphs are prepared from the proteins which are updated for adding new proteins and their connections. It is critical for the biologists to determine the functionality of node added new, without any experimental results. The addition of new protein is accurately managed by mining subgraphs having similar interactions. Saha et.al. prepares a model to determine the interface region of the complex structure of proteins and complex patterns have been mined using frequent subgraphs [5].

Consider a collaborative graph depicted in Figure 1. Typically, in such graphs, the collaboration is shown among the authors having the same work using the directed graphs. The more interested subgraphs can be made by reducing the threshold level. The reduction in level is attained to acquire interdisciplinary collaborations. Moreover, there is a surge in qualified intermediate outcomes of the mining process and expensive computations are intensified. For instance, a state of art method for directed graph diminishes after whole day consuming 190GB as an input graph having 100K nodes and 1M edges. Therefore, it is very crucial to develop an efficient mining algorithm with low frequency based threshold value. There are two settings considered in existing literature: transactional and direct graphs. In the former case, small graphs are considered to represent different transactions, such small graphs are created using the database. The transactions are denoted by ∂, which is defined by the user for a threshold. A frequent subgraph indicates τ transactions. In this paper, directed graphs are focussed in which a single large graph is considered having at least ∂ aspects in the directed graph. Such an informative context has been used in various applications, like Protein-Protein Interaction (PPI)and social networks [6]. These social networks can be modelled by a directed and undirected graph. The directed graphs which are appropriately labelled explicit more information than undirected graphs, especially for the chemical compounds [7].

The social networks developed by a directed graph, in which nodes indicates individuals, and the edge between nodes indicates the relationship between the individuals. Directed graphs using MFP present meaningful information. The generalization view of directed graph setting is a transactional one, in which nodes and edges are connected within a graph.

The detection of frequent subgraphs in a directed graph is difficult because a large number of identical subgraphs may overlap each other, which is complicated in the case of directed graphs. The complexity level is directly proportional to the exponential factor of the graph size, which is computationally demanding.

The number of repetitions per unit time (Frequency) of a subgraph (S) in a directed graph (D) is evaluated using the isomorphism of subgraph (S) in D [8]. There is an exact match of a subgraph in the directed graph using edges, nodes, and labels. Let us consider an example in Fig. 1, in which subgraphs having three isomorphism's. The best method to extract frequent subgraphs in D isa Grow and Store (GS) technique which is processed by the following steps: -

1. The nodes appearing ∂ times must be detectedand store these nodes with their appearance.
2. The frequent subgraphs are constructed by extending the stored nodes, determine the frequency of these nodes and then store the newly generated appearance of frequent subgraphs.
3. When no more frequent subgraphs detected, then stop the process. Otherwise, repeat step 2 for modeling frequent subgraphs.

The variations of the GS method is used by the already developed methods like SIGRAM. The stored appearances of the node are taken as an advantage by these methods to determine the frequency. The main obstruction of such methods is the generation of all aspects of each subgraph and storing them. The number of occurring such appearances depends upon the size and properties of the subgraph and the graphs. It is difficult to model the grow and store solutions practically because the calculation of appearances is prohibitively too large to compute and store. In this paper, GRAMI (Graph Mining) approach is used for developing a novel framework which addresses the problem of Frequent Subgraph (FS) mining.The GRAMI approach is different from other methods like grow and store method. In this framework, the templates of FS are stored without its appearances on the directed graph. This method limits the cons of GS method, which allows the developed GRAMI approach to generate Low-Frequency Threshold (LFT) value and a large number of graphs were mined. Moreover, the frequency is evaluated by using the developed approach. The developed approach used as a satisfaction problem for the constraints. The number of appearances generated at each iteration by solving the constrained problem. This frequency is evaluated using the appearances, which are sufficient and all the remaining appearances are eliminated. The process is repeated number of times until no more FS found.

The exponential factor in the constrained problem is worst to solve. More specifically, the large graphs in real life applications are supported by employing a heuristic GRAMI approach. In addition, optimization algorithms are also used to revamp the performance. A novel optimization algorithm is developed using the GRAMI approach.

1. There is apossibility of acquiring a large search space using the optimization approach.
2. The fast appearances are prioritized and slow searches are postponed.
3. Special graph structures and types is an advantageous factor.

The exhaustive appearances are eliminated using the developed optimization approach. The directed graphs and LFT are supported by the proposed GRAMI method than existing methods. Let us consider an example in which frequent patterns are computed having 100K nodes and 1M edges that it is difficult for the GS method while GRAMI takes only 15 minutes. In this paper, we propose the working model, which is based on GRAMI approach.

For example, Fig.1 depicts the GRAMI approach which considers $m_5\overset{\cdots}{-}m_8\underline{10}m_9$ matched to Z_1 using m5 (labeled DB) which is indirectly connected to m8 (labeled AI). The set of constraints are defined by the user which is both semantic and structural. The search space is limited and undesirable matches are pruned by the constraints. An approximate version AGRAMI, which is a final extension, the subgraph frequencies are approximated. In addition, some frequent subgraphs such as false negatives may be missed by the approximation method. But, returned results do not have false positives as these are not approximated. Superficially, AGRAMI and GRAMI for directed graphs can be implemented to both multiple and single labels. The nodes and edges are described with their weights as depicted in fig. 1. If we summarised, then our main objectives are to propose a novel framework in which frequent subgraphs are mined in a directed graph.

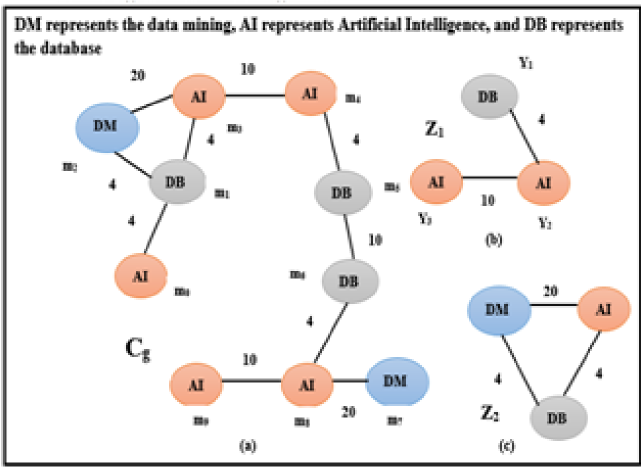
DM represents the data mining, AI represents Artificial Intelligence, and DB represents the database
DM
AI
DB
20
10
4
m_1
m_2
m_3
m_4
m_5
m_6
m_7
m_8
m_9
C_g
Z_1
Z_2
Y_1
Y_2
Y_3
(a)
(b)
(c)

Fig. 1. (a) A collaborative graph C_g in which nodes represents the authors which are labeled with their fieldwork and edges indicates the co-authorship labeled with their co-author papers. (b) Subgraph Z_1 (c) Subgraph Z_2.

A novel idea is implemented using the GRAMI approach in which appearances are refrained by calculating and storing intermediate outcomes. The constrained problem for satisfaction is used to determine the frequency of the subgraphs. A heuristic approach with novel optimization is implemented which revamps the performance of the GRAMI approach.

This is done by exploring the search space using specially directed graphs, some search spaces are pruned and some are postponed. The frequent patterns are mined by developing the variants of GRAMI. In addition, modern applications required a more powerful and robust version of matching.

In this paper, CSP is considered and GRAMI approach has been followed with its semantic and structural constraints. The extended version AGRAMI, which is an approximate version which generates results with no more false positives. Experimentally, the performance of the proposed approach is evaluated and computed that its magnitude is 3 orders of faster than already developed approaches in real life applications. The paper is organized as follows.

Section 2 demonstrates the problem formulation. Section 3 represents the GRAMI and its formulation. Section 4 discusses the result and finally concluded in section 5.

II. PRELIMINARIES

A graph C_g = (T, D, K) comprises of a set of nodes N, and set of edges E with labeling function L which assigns labels to N and E is defined. A graph Z = (T_Z, D_Z, K_Z) is a subgraph of a directed graph C_g= (T, D, K) if $T_Z \subseteq$ T, $D_Z \subseteq$ D, $K_Z \subseteq$ Kand K_Z(y) = K(y) for all y ∈ $T_Z \cup D_Z$. Figure 1 (a) describes an example of a collaborative graph. The labels of node depict author's fieldwork (DB), and edges label depicts the number of co-authored papers. To simplify, the proposed approach is applied and illustrate the examples of a directed graph, in which each node is labeled single times. More specifically, the proposed method also support undirected graphs in which each node is labeled multiple times.

Definition 1:- Let us consider an example with Z = (T_Z, D_Z, K_Z) be a subgraph of a graph C_g= (T, D, K). The isomorphism subgraph of Z to C_g is a relative (injected) function f: $T_Z \rightarrow$T which satisfies (a) K_Z(y) = K (f(y)) for all nodes y ∈T_Z, and (b) (f (m), f (y)) ∈ D and K_Z (m, y) = K(f(m),f(y)) for all edges (m,y) ∈ D_Z.

Apparently, a subgraph isomorphism mapped from T_Z to T such that each edge in D in which single edge is mapped in D_Z and vice versa. The nodes and edges labels are preserved by this mapping. For instance, Figure 1 depicts the subgraph Z_1 (y_1 $\underline{4}$ $y_2\underline{10}y_3$) has three isomorphisms with respect to graph C_g, named as m_1 $\underline{4}$ m_3 $\underline{10}m_{4,}$ $m_5\underline{4}$ $m_4\underline{10}m_3$, and $m_6\underline{4}m_8\underline{10}$ m_9.

The support factor of a subgraph in a directed graph can be computed by counting its isomorphism. The produced subgraph is not anti-monotone because extension graph appears longer time than its subgraph. For example, Figure 1(a) shows that subgraph having single node DB appears thrice while its extension DB 4 AI appears quad times.Moreover, there isa wide importance of anti-monotone support which allows the development of approachesin which search space is pruned, in which exhaustive search cannot be avoided [9]. There are a variety of anti-monotone support metrics described in literature such as harmful overlap (HO), minimum image based (MIB), and maximum independent sets (MIS). These support metrics are different from each other in terms of complexity and degree of overlapping allowed between isomorphism in a subgraph [10].The other reasons are:-

1. The NP-complete comprises of computation of HO and MIS.
2. The outcomes of alternative metrics with superset are provided. The computational cost of MIS and HO is very high in which unqualified graphs are discarded.The MIB metrics is explained as follows:-

Definition 2 LetN_1…….N_m is a set of isomorphism of a subgraph Z (T_Z, D_Z, K_Z) in a graph C_g. Assume N(y) = {N_1(y),………..,…..N_m(y)} is a set which consists of distinct nodes in C_g whose functions N_1…….N_m map a node y ∈T_Z. The MIS support of Z in C_g, denoted as z_{Cg}(Z), is stated as z_{Cg}(Z) = min{t | t= | N(y)| for all y ∈T_Z}.

For example, the subgraph in Z_1 in Fig. 1(b) and the graph C_g of Fig. 1(a), we get N(y_1) = {m_1, m_5, m_6}, N(y_2) = {m_3,m_4,m_8} and N(y_3) ={ m_3,m_4,m_9}, thus z_{Cg}(Z_1) =3. In comparison, the respective MIS metric value approaches to 2. As its isomorphism is $m_1\underline{4}$ $m_3\underline{10}m_4$ and m_5 $\underline{4}m_4$ $\underline{10}m_3$ overlapping takes place and MIS metric is regarded. The problem of frequent mining in subgraph is stated as below:-

Problem 1:- A graph C_g is given in figure 1 with a support threshold ∂, the mining problem in the frequent subgraph of isomorphism is defined as determining all subgraphs Z in C_g such as zC_g(Z)≥ ∂. In this problem, actual appearances of the subgraphs are not considered such that these appearances are greater than ∂. This is very important in various applications [11]. There are some applications, which demands exact frequencies as in graph indexing [12]. It is difficult to implement the problem 1 as a computational cost is very high. Moreover, its implementation depends upon the NP-hard isomorphism problem. The labels of both edges and node are enforced to match with definition 1.

For example in Figure 1, there is only one isomorphism in subgraph Z_2 which is formed by nodes y_1, y2, and y_3. Researchers argue that such matching is restrictive, which is relaxed by indirect relationship and difference in edges of the subgraph and graph. This aspect also considers subgraphs of $m_6\underline{4}m_8\underline{20}m_7$is a match of Z_2 because DM and DB are connected indirectly. This match is considered a pattern. The definition of mining frequent patterns is taken from [13]. Moreover, a distance metric is employed to compute the difference between nodes. Specifically, a metric function is also used which satisfies the inequality property. The weights associated with the edge labels are used to determine the distance function. Figure 1 shows the collaborative graph C_g, in which distance function is used $\Delta_h(m, y)$ defined as a number of nodes, which connects the m and y in the shortest path. For example, the value of $\Delta_h(m_0, m_3)$ is 2. In addition, we may also use $\Delta_p(m, y)$ stated as a minimum sum which connects the path between m and y. Consider an example $\Delta_p(m_{6,}m_7)$ = ¼ +1/20 = 0.3. Specifically, stronger collaboration between nodes determined by shorter distance. Figure 2 describes the values ofΔ_p for the graph C_g of Figure 1. The original edges are represented by the solid lines while transitions are represented by the dotted lines.

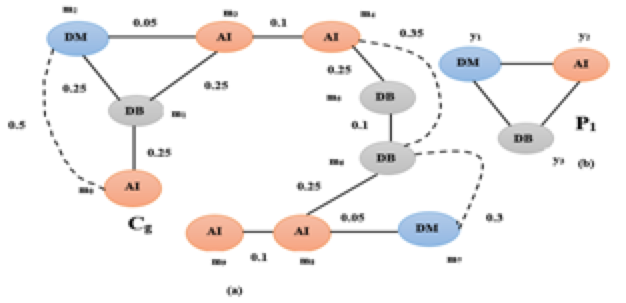

Fig. 2 (a) represents the distance Δ_p for the C_g of Fig. 1(b) represents the pattern P_1.

Definition 3 A graph J= (T_J, D_J, K_J) is a pattern of a graph C_g (T, D, K) if $T_J \subseteq$ T, K_J (y) = K(y) for all y ∈ T_J and K_J(e) =φ for all e ∈ D_J.

In other words, a pattern corresponds to subgraph in which edge labels are not considered. For example, a pattern J_1 of the graph C_g is represented in Figure 2b.

Definition 4 Let J= (T_J, D_J, K_J) is a pattern of a graph C_g =(T, D, K) in which Δ is a distance function and τ is a user-defined threshold. An injection of pattern J to C_G is an embedded function ∅ : T_J→ T which satisfies

$$K_J(y) = K(\emptyset(y)) \text{ for all nodes}$$

III. RELATED WORK

The subgraph detection in a large graph has various applications in different domains like computer vision, electronics, designing, intelligence, and biology [13]. Pattern detection using graph mining also has various possibilities for research. Since it is known that the graph requirements varied according to the applications. An attempt has been in research to detect the subgraphs using pattern mining. The weighted edge detection method has been used in past studies to detect the patterns [14]. A Spark model has been used for the frequent subgraph mining in a graph. The research has been focussed on the extension of subgraphs. An effort has been made on solving complex problems using optimizations methods [15]. Song et. al. proposes an approach to search a knowledgeable graph using the mining algorithm. The paper basically deals with the problem of bicriteria pattern mining. The knowledgeable graphs bounded by k-diversified summaries has been considered to trade off accuracy and speed [16]. More specifically, more impact is given on online mining rather than constrained for a single large graph. But, mining algorithms do not handle problems in a single large graph due to centralization. The efficiency and high cost of communication make it less popular in mining graphs. To overcome this problem, the structure of the distributed graph was considered in past studies using Pregel. It is similar to mining algorithm in which different phases of the design algorithm has been considered. The selection, extension, and pruning have been done using a large graph. In addition, the real-life data set has been used to approximate the results [17].A more robust and distributed method to mine a massive graph is crucial. So, DistGraph algorithms solve the false negative problem in mining graphs to minimize communication in computational nodes [18]. Specifically, frequent subgraph mining considers small graphs in large graphs. The repetitive graphs create an issue which cumbersome by identifying isomorphisms of the graph. But, detection of isomorphism has been difficult. Thus reduction of repetitive graphs has been possible through the filtration technique. This simply reduces the overall time and complexity of the graph to a great extent [19]. The data eruption technique is also feasible for multigraphs. A MuGram technique applied to discover the frequent multigraph patterns. The swift discovery enables the detection of subgraphs from multigraphs to handle the complex data. The rich representation of data through graphs make the extraction of data simple and generic [20]. Although, literature studies for mining graphs based on frequency threshold whichassume the data of many graphs in a single large graph. The threshold approach is cost-oriented. GRAMI used for mining frequent patterns assumes the threshold value for a reduction in computation cost. Graphs often capture powerful information ensemble through graph node. But instances having minimal set often satisfies the requirement by avoiding the computational cost. A more robust approach known as CGRAMI solves the problem of complex mining [21].The low accuracy problem has been mitigated using the null model. The model basically uses the edges and vertex to extract information from the graph. In addition, the indexing has been attained using the build location approach. The model improves the accuracy of the classification approach using the designed algorithm [22]. But, still,the problem of computation makes the model inefficient. In this paper, a robust approach for frequent pattern mining has been proposed which reduces the computation time. This paper follows the GRAMI approach for efficient results and edge set value has been determined from the graph. The proposed approachhas been compared to GRAMI model for effective results.

IV. PROPOSED WORK MODEL

The proposed work model is inspired by the GRAMI approach. The preliminaries of the proposed approach are as follows.

It is an architecture which focuses on the isomorphic issue of the sub-graph mining architecture. The constraint satisfaction problem (CSP) is deeply discussed and analyzed using the Grami approach. The isomorphism of the subgraph is mapped by CSP where each domain has set of variables x with vertex set V. The mining set contains lamda as the frequency threshold and checks which edge set matches the frequency threshold. It is also marked as a sub-graph extension which takes frequent edge sets.

Algorithm: SUBGRAPH EXTENSION

Input: The sub-graph G with edge set E

```
$For_{each}\ edge\ in\ Edge_{set}\ E$
$If\ E.edgevalue\ is\ equal\ to\ the\ edge$
$Counter++$
$End$
$If\ Counter > E.threshold$
$Sub_{graph}\ \ Encountered$
$Find\ Sub_{graph}\ in\ all\ Graphs, If\ found$
$Segregate$
$End$
```

The algorithm searches the edge set value into the entire graph. If the match manages to get a threshold more than its matching threshold, it is termed as frequent and in the similar fashion, if the sub-graph equalizes to a subsequent matching threshold then it is termed as isomorphic as per Grami.

If the edge set associated threshold is equal to demanding weight, the Grami does not allow it to pass through and it becomes the issue of proposed work model.

The proposed algorithm sets a threshold of 20% which also manages the 20% of edge weight and if any sustainable edge value is set to be matching, it is counted as accountable.

Proposed Algorithm Architecture

Required Input: Pattern of Graph, E_{VALUE} & Pattern Edge set

Obtained Output: Edge setRange

```
Begin
  For each pattern in Graphs// Patterns in the graphs has been initialized
    For each $e_{value}$ in a pattern// Edge set value has been initialized
      For each edge in Pattern.Edgeset// Initiated Loop according to pattern edge set
        $Edge_{value} = edge.value$// Edge Value from the structure of edge
        $Range_{Threshold} = Edge_{value} * 20/100$ // Limit of Range
        $Range = [\,Edge_{value} - Range_{Threshold}\ \ Edge_{value}\ \ Edge_{value} + Range_{Threshold}]$ // Compute range
        If $e.value.edgevalue \geq Range[0]$ or $e.value.edge_{value} \leq Range[1]$ or $Range[2]$
          $Subsequent\ Isomorphic\ {++}$
        $End_{If}$
      $End_{For}$
    $End_{For}$
  $End_{For}$
  Return:Range as an output
End
```

The proposed algorithm architecture sets a threshold of 20 % and creates a slab which starts from the edge value which is 20% less than that of the original edge value and goes to value which is 20% greater than the edge value.

The proposed work model aims to remove the isomorphism as well and hence the same algorithmic architecture is applied to the sub-graphs also. The following dataset has been applied to both Grami and proposed algorithm architecture.

A. Dataset:

The Twitter DB: The dataset is available on socialcompting.asu.edu/datasets which contains the tweets of Twitter and other social network data. The dataset is huge and contains 11 M node structure.

The dataset contains 85M edges structure. The nodes structure contains the user's file. This file acts as a dictionary for all users in the prescribed data set. The users specified in the node ids.

1. The representation of nodes is in the form of nodes.csv in the dataset.
2. The representation of edges is in the form of edges.csv in the dataset.

The edges are represented by the friendship and follower network between users. Directed relation represents edges.

It is a social news website which is a combination of SMS, e-mail, and instant messaging.

It is a new way to update about the latest news related to different subjects such as science, business, media, education, etc. But, the major issue with this dataset is that it is not labeled.

Even previous authors have put the label on their own. The Gaussian distribution has been followed for randomization.

V. RESULTS AND DISCUSSION

This section deals with the results and discussion of the proposed approach. The following parameters have been evaluated.

a) Time in Seconds: It is the total time unit which is consumed in order to execute the supplied pattern. Around 11M nodes were transversed, processed and the execution time is evaluated in seconds.

b) Total Memory Consumed: It is the consumed memory for the processing. Now as the sub-graphs are arranged in a subsequent order, the proposed work model will consume less amount of memory.

Table-I: Execution time

Support threshold	Time in seconds for GRAMI	Time in seconds for the proposed approach
2000	1150	1100
2100	1050	1000
2500	990	900
3000	980	890
3500	950	890

Table 1 shows the execution time for the proposed approach which is finally compared with the GRAMI approach. The execution time has been shown for both techniques under different support threshold. The average execution time for the GRAMI approach is 1024 seconds while that of the proposed approach is 956 seconds. Thus obtained accuracy is approximately $\frac{1024-956}{1024} \times 100 = 6.6\%$ using the proposed approach.

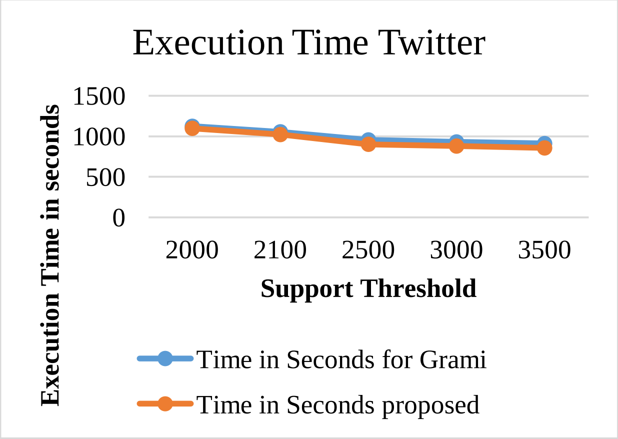

Fig.3.Execution Time

Figure 3 stated that the execution time for 2000 supported threshold is 1100 seconds. As the supported threshold increases, the execution time reduces. It is seen that after 2500, execution time remains constant for GRAMI. The proposed approach having support threshold of 2500, execution time becomes 900 seconds while that of GRAMI approach, it approaches to 1000 seconds. The execution time falls down for the subsequent thresholds. As per the implementation of the 20 % selection rule, the proposed work algorithm enhances the quality of execution and hence the execution time decreases by approximately 6.6%.The memory consumption is calculated in MB. As the sub-pattern increases, the space to store the data decreases.

Table-II: Memory Consumption

Support Threshold	Memory consumption using GRAMI approach	Memory consumption using the proposed technique
2000	380	300
2100	400	280
2500	400	300
3000	420	320
3500	430	380

Table 2 shows the consumption of memory for the proposed approach which is finally compared with the GRAMI approach. The memory consumption has been shown for both techniques under different support threshold. The average memory consumption for the GRAMI approach is 406MB while that of the proposed approach is 316MB. Thus obtained accuracy is approximately $\frac{406-316}{406} \times 100 =$ 22% using the proposed approach.

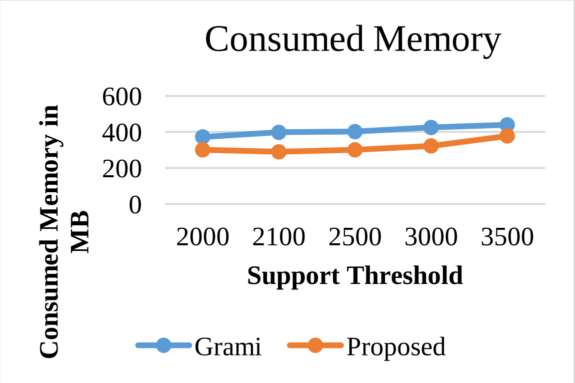

Fig. 4. Consumed Memory

Figure 4 represents the consumption of memory for the proposed approach and GRAMI approach. A comparison has been made on the basis of support threshold and consumed memory in MB. It is seen that memory consumption under 2000 support threshold is 300MB for the proposed approach and 380MB for the GRAMI approach. The consumption of memory for the GRAMI technique increases for the subsequent threshold. The proposed technique shows a slight increase in memory consumption. Hence, it is proved that memory consumption using GRAMI keeps on increasing and that of proposed technique follows a zigzag path like increase, then decrease.

VI. CONCLUSION

In a nutshell, it is known that many applications rely on graph mining for its reliability. This paper employs a novel framework for pattern mining in a directed graph, in which frequent patterns having isomorphism are removed. The proposed approach is based on the GRAMI concept in which versatile algorithms has beenimplemented for mining subgraphs. The weighted graph setprovides constrained results, in which weights are attached to vertexes for clarity. The modeling of the algorithm determines the edge set value in a graph, in which a crux idea of the GRAMI has been used. The threshold value is set through the algorithms and problem of constrained structured is sorted. In addition, the issue of isomorphism in a directed graph is removed. The parameters such as time and memory consumption are computed using the designed algorithm. The experimental results are demonstrated using the data set which depicts the effectiveness of the developed approach. The results concluded that execution time reduces and improved by 6.6% and less memory is consumed which is revamped by 22% in comparison to GRAMI approach.

REFERENCES

1. Zhang, S., Du, Z., Wang, J.T. and Jiang, H., 2018. Discovering frequent induced subgraphs from directed networks. *Intelligent Data Analysis*, *22*(6), pp.1279-1296.
2. H. Cheng, X. Yan, and J. Han, "Mining graph patterns," in Frequent Pattern Mining, pp. 307–338, 2014.
3. Y. Chen, X. Zhao, X. Lin, Y. Wang, and D. Guo," Efficient Mining of Frequent Patterns on Uncertain Graphs", *IEEE Transactions on Knowledge and Data Engineering*, *31*(2), pp.287-300, 2019.
4. Y. Cho and A. Zhang, "Predicting protein function by frequent functional association pattern mining in protein interaction networks," IEEE Transactions of Information Technology in Biomedicine, Vol. 14, No. 1, pp. 30–36, 2010.
5. Saha, T.K., Katebi, A., Dhifli, W.,and Al Hasan, M., 2017. Discovery of functional motifs from the interface region of oligomeric proteins using frequent subgraph mining. *IEEE/ACM transactions on computational biology and bioinformatics*.
6. D. Rhodes, S. Tomlins, S. Varambally, V. Mahavisno, T. Barrette, S. Kalyana-Sundaram, D. Ghosh, A. Pandey, and A. C. AM., "Probabilistic model of the human protein-protein interaction network," Nat Biotechnol., vol. 23, no. 8, pp. 1–9, 2005.
7. M. Deshpande, M. Kuramochi, and G. Karypis. Frequent sub-structure-based approaches for classifying chemical compounds. In *Proc. of ICDM*, pages 35–42, 2003.
8. Y. Gu, C. Gao, L. Wang, and G. Yu, "Subgraph similarity maximal all-matching over a large uncertain graph," World Wide Web, vol. 19, no. 5, pp. 755–782, 2016.

9. W. Zhang, X. Lin, Y. Zhang, K. Zhu, and G. Zhu, "Efficient probabilistic supergraph search," IEEE TKDE, vol. 28, no. 4, pp. 965–978, April 2016.B. Cabrera, 2019, "Using Subgraph Distributions for Characterizing Networks and Fitting Random Graph Models", In Emerging Research Challenges and Opportunities in Computational Social Network Analysis and Mining, Springer, Cham, pp. 107-129.
10. Wang, Y., Ramon, J., & Fannes, T., "An efficiently computable subgraph pattern support measure: counting independent observations", Data mining and knowledge discovery, Vol. 27, No. 3, pp 444-477, 2013.
11. X. Yan, P. S. Yu, and J. Han. Graph indexing: a frequent structure-based approach. In Proc. of SIGMOD, pp. 335–346, 2004.
12. Rao, B. and Mishra, S., An Approach to Detect Patterns (Sub-graphs) with Edge Weight in Graph Using Graph Mining Techniques, In Computational Intelligence in Data Mining, Springer, Singapore, pp. 807-817, 2019.
13. G. Preti, M. Lissandrini, D. Mottin, and Y. Velegrakis, "Mining patterns in graphs with multiple weights," Distributed and Parallel Databases, pp.1-39, 2019.
14. F. Qiao, X. Zhang, , P. Li, Z. Ding, S. Jia, and H. Wang, "A parallel approach for frequent subgraph mining in a single large graph using spark", Applied Sciences, Vol. 8, No. 2, pp.-230-239, 2018.
15. Song, Q., Wu, Y., Lin, P., Dong, L.X. and Sun, H., 2018. Mining summaries for knowledge graph search. IEEE Transactions on Knowledge and Data Engineering, 30(10), pp.1887-1900.
16. Bhatia, V. and Rani, R., 2018. Ap-FSM: A parallel algorithm for approximate frequent subgraph mining using Pregel. *Expert Systems with Applications*, *106*, pp.217-232.
17. Talukder, N. and Zaki, M.J., 2016. A distributed approach for graph mining in massive networks. *Data Mining and Knowledge Discovery*, *30*(5), pp.1024-1052.
18. Dhiman, A. and Jain, S.K., 2016. Optimizing Frequent Subgraph Mining for Single Large Graph. *Procedia Computer Science*, *89*, pp.378-385.
19. Ingalalli, V., Ienco, D. and Poncelet, P., 2018. Mining frequent subgraphs in multigraphs. *Information Sciences*, *451*, pp.50-66.
20. Elseidy, M., Abdelhamid, E., Skiadopoulos, S. and Kalnis, P., 2014. Grami: Frequent subgraph and pattern mining in a single large graph. *Proceedings of the VLDB Endowment*, *7*(7), pp.517-528.
21. Wang, Z., 2017. Research on subgraph distribution algorithm based on label null model. *Cluster Computing*, pp.1-13.

AUTHORS PROFILE

Shriya Sahu received M.Tech. degree in Computer Science and Engineering from MANIT, Bhopal, India, in 2012. She is currently working towards the PhD degree in the Department of Computer Science and Engineering at the MANIT, Bhopal. Her research interests include big data and graph mining.

Dr. Meenu Chawla is a professor and head in the Department of Computer Science and Engineering at the Maulana Azad National Institute of Technology, Bhopal, India. She has coordinated several workshops and also taken a number of expert lectures. She is a member of CSI and ISTE. She is having more than 25 years of teaching experience and published more than 50 papers in reputed international and national journals. Her research interests include wireless networking, big data, and sensor networks.

Dr. Nilay Khare is a professor in the Department of Computer Science and Engineering at the Maulana Azad National Institute of Technology, Bhopal, India. He is a life member of CSI and ISTE. He is having more than 30 years of teaching experience and published more than 80 papers in reputed international and national journals. His research interest includes big data, machine learning, theoretical computer science, and graph mining.