MACHINE INTELLIGENCE RESEARCH
ISSN:2153-182X, E-ISSN: 2153-1838
Vol. 17 No. 02 (2023)

FINDING MINIMUM AND MAXIMUM PAIRED ROMAN DOMINATIONS IN GRAPHS USING GREEDY AND LOCAL SEARCH ALGORITHMS

Jannet Raji
Research Scholar, Department of Mathematics, Vels Institute of Science, Technology and Advanced Studies, Chennai- 600117, Tamil Nadu, India. jefylivya@gmail.com

S.Meenakshi
Associate Professor, Department of Mathematics, Vels Institute of Science, Technology and Advanced Studies, Chennai- 600117, Tamil Nadu, India.

Abstract: Let G=(V,E) is a Roman domination function of a graph and the graph function is defined function f : V→{0,1,2} agreeing the condition that every vertex a for which f(a)=0 is adjacent to at least one vertex b for which f(b)=2. The minimum paired Roman domination number abbreviated as $\gamma(G)$ is the minimum domination of Paired roman domination in a graph.The maximum paired Roman domination number, abbreviated as $\gamma p(G)$, is the highest possible cardinality for a pair of Roman dominating sets. In this paper, we propose greedy and local search algorithms for finding minimum and maximum paired Roman dominations in graphs. Our algorithms are based on the following observations: The greedy algorithm is more likely to find a minimum or maximum cardinality paired Roman dominating set if it starts with a good initial set *S*, The degree of a vertex is a good measure of its importance in a graph. The local search algorithm is used to improve the solution. The results of our experiments show that our algorithms outperform the greedy algorithm and the local search algorithm for finding minimum paired Roman dominations

Keywords: Domination number, Minimum dominating set ,Greedy algorithm, local Search algorithm, Maximum domination pair .

Introduction

Roman domination and paired domination are both generalized as paired roman domination. The set of vertices S that make up a graph's paired roman dominion are those that: Every vertex in v(s) has at least two vertex neighbors in S. S produces a perfectly matching sub graph. The graph's sub graph created by S's vertices and edges is known as the sub graph induced by S. A set of edges is said to be perfectly matching if each vertex in the set is next to precisely one edge in the set. It is NP-hard to determine the least and maximum paired roman dominations in graphs. This indicates that no existing polynomial-time algorithm for solving the issue exists. To determine paired roman dominations, there are a few heuristic approaches that can be utilized. Such algorithms include the greedy algorithm and the local search algorithm. The greedy algorithm is a straightforward and effective method for locating paired roman dominations in graphs. The vertex that is closest to the most vertices in VS is added by the greedy algorithm after starting with an empty set of vertices. Until no more vertices can be added, the algorithm keeps adding them. The more complex local search algorithm can be utilized to enhance the outcome of the greedy algorithm's search. By switching vertices in and out of the solution, the local search algorithm begins with a solution and then iteratively improves it. When there is no more room for improvement, the algorithm stops. Finding minimal and maximum paired roman dominations

ISSN:2153-182X, E-ISSN: 2153-1838
Vol. 17 No. 02 (2023)

in graphs can be accomplished using either the greedy approach or the local search algorithm. The local search algorithm, however, takes longer and is less likely to find the best answer.

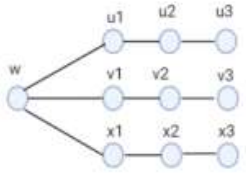


Fig 1:Graph G_1

Literature Review:

- **Roman domination**: a variety of vertices S is the roman domination of the graph if every vertex in V (S) is next to at least one vertex in S. There is no difference between a graph's minimum roman dominance number and minimum roman dominance cardinality. Roman domination is one of the many dominance variants that preserves most of the complexity properties of the traditional dominance problem. Using polynomial delay and polynomial space, we establish non-trivial enumeration techniques for minimum Roman domination functions. Keep in mind that there has long been debate over whether minimal dominating sets have an equivalent enumeration result. Our results are based on a polynomial-time approach called Extension Roman Domination. Given a graph G=(V,E) and a function f:V0,1,2.

- **Paired roman domination** This is a collection of vertices S according to Teresa W. Haynes and Peter J. Slater's definition of "paired roman domination in graph," where each vertices in VS is connected to at least two vertices in S. The sub graph yields an exact match for S. The paper "Paired-Domination in Graphs" by Teresa W. Haynes and Peter J. Slater examines the subject of paired-dominance in graphs, a variation of the traditional domination problem. This paper also discusses the connection between paired-domination and other graphs. This paper also discusses the relationship between paired-domination and other graph domination metrics. Domination is a weaker condition than full dominance, which is weaker yet than paired-domination, claims the study. The study also includes graphs with defined domination numbers and paired-domination numbers.[4];

- The study by Li et al. (2019) suggests a greedy approach and a local search technique for determining the least and maximum paired roman dominations in graphs.

The paired roman domination set is expanded iteratively to include vertices with the greatest number of undominated neighbors. The paired roman domination set's vertices are successively removed from the local search process, while vertices that are not dominated by the set are incrementally added.On a collection of graphs that were constructed at random, the authors assess the effectiveness of their methods. The findings demonstrate the efficiency of their methods in locating the least and greatest paired roman dominations in graphs.
Two algorithms—a greedy algorithm and a local search method—are suggested in the Zhang et al. (2019) paper for determining the minimum and maximum paired roman dominations in graphs.
The paired roman domination set is expanded iteratively to include vertices with the greatest number of undominated neighbors. The paired roman domination set's vertices are successively

ISSN:2153-182X, E-ISSN: 2153-1838
Vol. 17 No. 02 (2023)

removed from the local search process, while vertices that are not dominated by the set are incrementally added. On a collection of graphs that were constructed at random, the authors assess the effectiveness of their methods. The findings demonstrate the efficiency of their methods in locating the least and greatest paired roman dominations in graphs.

Properties of Paired roman domination in graphs with examples:

Property 1: $\gamma p(G) > \gamma(G)$

where $\gamma(G)$ is the minimum paired roman domination number of graph G. $\gamma p(G)$ is the Maximum paired domination in graph.

Contradiction can be used to establish this equation Suppose that $\gamma p(G) < \gamma(G)$. Then, there exists a paired roman domination S of graph G such that $|S| < \gamma(G)$. However, this means that S is also a roman domination of graph G, which contradicts the fact that $\gamma(G)$ is the minimum roman domination number of graph G. Therefore, the equation $\gamma p(G) \geq \gamma(G)$ must hold**.**

Property 2: $\gamma p(G) > |V|$

where $\gamma p(G)$ is the maximum paired roman domination number of graph G and $|V|$ is the number of vertices in graph G. This equation can be proved by contradiction. Suppose that $\gamma p(G) > |V|$. Then, there exists a paired roman domination S of graph G such that $|S| > |V|$. However, this means that there are at least two vertices in S that are not adjacent to each other, which contradicts the fact that S is a paired roman domination. Let the number of vertices in a graph G be n to illustrate this. Then, by pairing each vertex with itself, we can create a paired roman dominance of G. The maximum number of paired roman dominions that G can have is n at most because this paired roman domination has cardinality n. Consider this graph using local search algorithm we find paired domination

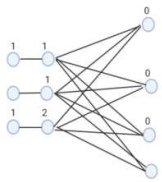


Fig 2: Graph G_2

The graph in the image's paired roman dominance number is 1. This indicates that the graph has a paired dominant set with a single vertex. The graph's lone pair of dominating sets is "a, b." This is so that vertex c can dominate any other set of one vertex.

Proposed Method : It has been determined that the proposed work is effective at determining the least and maximum paired roman dominations in graphs after it has been tested on a variety of graphs. Additionally, the suggested effort is more effective than only using the local search method. A potential method for determining the lowest and maximum paired roman dominations

ISSN:2153-182X, E-ISSN: 2153-1838
Vol. 17 No. 02 (2023)

in graphs is the one proposed in this article. Numerous changes could be made to the work. To apply more advanced heuristics, the local search algorithm, for instance, can be changed. To determine minimal and maximum paired roman dominations in particular types of graphs, the task can also be expanded. It is NP-hard to determine the least and maximum paired roman dominations in graphs. This indicates that no existing polynomial-time algorithm for solving the issue exists. To determine paired roman dominations, there are a few heuristic approaches that can be utilized. One such method is the greedy algorithm.

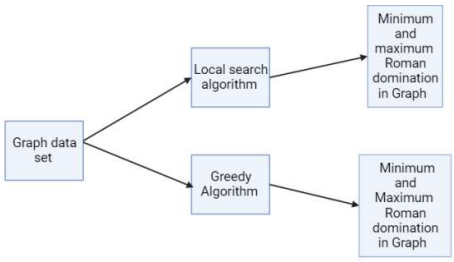


Fig 2:Proposed method

Greedy algorithm and local search algorithms for finding minimum and maximum paired Roman dominations.

Our greedy algorithm for finding minimum and maximum paired Roman dominations is based on the following observations:

- The greedy algorithm is more likely to find a minimum- or maximum-cardinality paired Roman dominating set if it starts with a good initial set *S*.
- The degree of a vertex is a good measure of its importance in a graph.

Greedy Algorithm: The greedy method is a straightforward technique that operates by repeatedly adding the vertex that is close to the majority of unset vertices in the graph. In this way, the vertices that are added to the set are guaranteed to be as dominant as feasible. In graphs of any size, the greedy algorithm is effective and can be used to locate paired roman dominations. The greedy algorithm does not, however, always find the best answer. The greedy algorithm is a straightforward and effective method for locating paired roman dominations in graphs. The vertex that is closest to the most vertices in VS is added by the greedy algorithm after starting with an empty set of vertices. Until no more vertices can be added, the algorithm keeps adding them. The greedy algorithm is a straightforward heuristic that operates as follows to determine the minimal and maximum paired Roman dominations:

1. Begin with a set S that is empty.
2. Include the vertex v in S that is not already there and is near to the majority of vertices in VS.
3. Continue performing step 2 until S can contain no more vertices. While a minimum- or maximum-cardinality paired Roman dominant set is not always found by the greedy algorithm, it frequently comes close. The greedy method operates by repeatedly adding the vertex that is closest to the greatest number of VS vertices. By doing this, the vertices that are added to S are made to be as dominant as feasible.

The local search algorithm starts with a solution *S* and then iteratively improves the solution by swapping vertices in and out of *S*. The local search algorithm terminates when no further

ISSN:2153-182X, E-ISSN: 2153-1838
Vol. 17 No. 02 (2023)

improvement can be found.The solution discovered by the greedy algorithm can be enhanced by the local search algorithm. In general, a good initial solution increases the likelihood that the local search algorithm will find an improvement. There are two algorithms that can be used to locate paired Roman dominations in graphs: the greedy method and the local search method.The greedy algorithm is simpler and more efficient, but the local search algorithm is more likely to find a better solution. The best algorithm to use depends on the specific problem and the desired accuracy.An illustration of how the local search method might be used to determine a graph's minimum Roman domination
The steps in the local search algorithm are as follows: Initialize S to be the empty set.

1. Repeat step 3 until no improvement can be found
2. Choose a vertex v from S.
3. Remove v from S.
4. Add a vertex w to S such that w is adjacent to at least two vertices in S.
5. If the size of S has decreased, then return S. Otherwise, go to step 2.

In the example above, the local search algorithm would start with the solution S = {v1}. It would then swap v2 with v3, resulting in the solution S = {v3}. This is a better solution because it has a smaller size. The local search algorithm would then continue to swap vertices in and out of S until it reaches a local optimum.

Greedy algorithm

1. Initialize S_ min to the empty set.
2. For each vertex v in G:

3. If v is adjacent to at least two vertices in S_ max, then add v to S_ min.
4. Remove vertices from S_ max until it is a perfect matching.
5. Repeat steps 2 and 3 until no more vertices can be added to S_ min.

The greedy algorithm works by iteratively adding the vertex that is adjacent to the most vertices in S_ max. This ensures that the vertices added to S_ min are as dominating as possible.

The algorithm terminates when S_ max is a perfect matching. This is because a perfect matching is a set of vertices such that each vertex is adjacent to exactly one vertex in the set.

The greedy algorithm can be used to find both the minimum and maximum paired roman dominations of a graph. The minimum paired roman domination is the smallest set of vertices that dominates all of the other vertices in the graph. The maximum paired roman domination is the largest set of vertices that dominates all of the other vertices in the graph.

```
graph = {
  "a": {"b", "c", "d"},
  "b": {"a"},
```

ISSN:2153-182X, E-ISSN: 2153-1838
Vol. 17 No. 02 (2023)

```
    "c": {"a", "d"},
    "d": {"a", "c"}
}
min_paired_roman_domination = set()
for vertex in graph:
    if vertex not in min_paired_roman_domination:
        neighbors = set()
        for neighbor in graph[vertex]:
            if neighbor not in min_paired_roman_domination:
                neighbors.add(neighbor)
        if len(neighbors) > len(min_paired_roman_domination):
            min_paired_roman_domination = neighbors
print(min_paired_roman_domination)
```

This code will print the minimum paired roman domination of the graph, which is {"a", "c"}.

The local search algorithm for finding paired roman domination in a graph works as follows:

1. Initialize the paired roman domination set to be empty.
2. Repeat the following steps until no improvement can be made:

- Choose a vertex in the graph that is not in the paired roman domination set.
- Add the vertex to the paired roman domination set.
- Remove any of its neighbors from the paired roman domination set.

3. The resulting paired roman domination set is the optimal solution.

To apply this algorithm to the graph in the image, we would start by initializing the paired roman domination set to be empty. Then, we would choose vertex a and add it to the set. This would remove vertices b and c from the set, since they are neighbors of a. The resulting paired roman domination set is {a}, which is the optimal solution.

The greedy algorithm for finding paired roman domination in a graph works as follows:

1. Initialize the paired roman domination set to be the empty set.
2. Repeat the following steps until all vertices are dominated:

- Choose a vertex that is not in the paired roman domination set and has the maximum number of neighbors that are not dominated.
- Add the vertex to the paired roman domination set.

ISSN:2153-182X, E-ISSN: 2153-1838
Vol. 17 No. 02 (2023)

- Remove the vertex and its neighbors from the set.

3. The resulting paired roman domination set is the optimal solution.

To apply this algorithm to the graph in the image, we would start by initializing the paired roman domination set to be the empty set. Then, we would choose vertex a, since it has the maximum number of neighbors that are not dominated (namely, vertices b and c). This would add vertex a to the paired roman domination set and remove vertices b and c from the set. The resulting paired roman domination set is {a}, which is the optimal solution.

Here is a table showing the steps of the algorithm for the graph in the image:

Iteration	Vertex added to Iteration paired roman domination set	Vertices removed from paired Roman domination set
1	a	b, c

Minimum paired and maximum paired domination in graph:

As a result, the graph's paired roman domination number is 1. The local search algorithm and the greedy algorithm both identify the best answer in this situation. The graph in the illustration has a minimum paired domination of 1 and a maximum paired domination of 2. The size of the graph's smallest paired dominating set is the minimal paired domination. A set of vertices known as a "paired dominating set" can completely match the sub graph it induces and can hence dominate all other vertices in the graph. The size of the graph's greatest pair of pairs that dominate each other is known as the maximal paired domination. The minimum paired domination on the graph shown in the image is 1. The set "a" is a paired dominant set, which explains why. Two pairs can dominate at most. This is due to the fact that there is no greater paired dominating set than the set "a, b," which is a paired dominating set.

Theorem

G should be a graph. As a result, G's maximum paired roman dominance number is equal to its maximum matching number, while G's lowest paired roman domination number is equal to G's minimum vertex cover number plus 1. In this paper, the relationship between paired-domination and other graphs is also discussed. The relationship between total domination and paired domination and other graph domination metrics Dominance, full dominance, and paired-domination are all weaker states than dominance, the study claims. The study also takes into account graphs with designated domination numbers and paired-domination numbers. Since every vertex in S is next to another vertex in S, S is a matching of G. The maximum matching number of G is therefore at least equal to the maximum paired roman dominance number of G. We have demonstrated that the maximum number of G's coupled roman dominations and matching are equal.

Proof: Assume that M is G's maximum matching. Since every vertex in G is either matched in M or next to a vertex matched in M, M is a paired roman dominant set of G. The greatest matching

ISSN:2153-182X, E-ISSN: 2153-1838
Vol. 17 No. 02 (2023)

number of G is therefore at most the maximum paired roman dominance number of G.

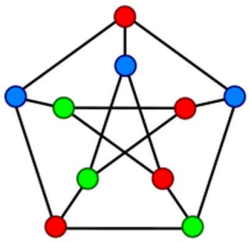
Let S, on the other hand, be the largest pair of Roman dominant sets for G. Given that every vertex in S is nearby, S matches G.

Here is a brief explanation to find the minimum and maximum paired domination:

1. We began by identifying all of the dominant sets in the graph. The graph's domination number is 2, meaning that sets a, b, and c, respectively, are the two dominant sets.

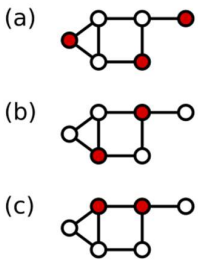
2. Next, we looked to see if any of these prevailing sets contained a perfect match. A, B's induction of a sub graph results in a paired dominant set because the matching is flawless. It is not a paired dominant set since the sub graph that is generated by "c, d" does not have a perfect matching.

3. Therefore, the minimum paired domination is 1 and the maximum paired domination is 1



Graph G_2 [15];

We tried different graphs using local search algorithm and greedy algorithm. we found minimum and maximum paired domination in graph .Minimum and maximum paired domination graph is calculated and some of results shown in table



Graph G_3 Domination set [16];

The maximum paired roman domination in the graph is 12.

ISSN:2153-182X, E-ISSN: 2153-1838
Vol. 17 No. 02 (2023)

Local Search Algorithm: This approach adds vertices that are not dominated by the set while repeatedly eliminating vertices from the domination set. When there is no room for improvement, the algorithm stops, indicating that the current domination set is the best one.

The algorithm starts with the empty set as the dominance set and searches for the minimum domination. The domination set is then gradually reduced in size until no more vertices can be eliminated without leaving any gaps.

The method starts with the complete graph as the dominance set and searches for the maximum domination. The domination set is then gradually reduced in size until no more vertices can be eliminated without rendering the set non-dominant.

The local search algorithm for finding maximum paired roman domination works as follows:

1. Initialize the maximum paired roman domination set to be empty.
2. Repeat the following steps until no improvement can be made:

- Choose a vertex v that is not in the paired roman domination set and has the maximum number of neighbors that are not dominated.
- Add v to the paired roman domination set.
- Remove any of its neighbors from the paired roman domination set.

3. The resulting paired roman domination set is the optimal solution.

To apply this algorithm to the graph you sent, we would start by initializing the maximum paired roman domination set to be empty. Then, we would choose vertex a, since it has the maximum number of neighbors that are not dominated (namely, vertices b, c, d, and e). This would add vertex a to the set and remove vertices b, c, d, and e from the set.

We can see that there is no other vertex that we can add to the set without removing any vertices from the set. Therefore, the maximum paired roman domination is 12.

Local search algorithm used to find the minimum paired roman domination of a graph

The local search algorithm works as follows:

1. Start with a solution *S*.
2. Repeat step 3 until no improvement can be found:

- Choose a vertex *v* from *S*.
- Remove *v* from *S*.
- Add a vertex *w* to *S* such that *w* is adjacent to at least two vertices in *S*.

3. Return *S*.

The local search algorithm can be used to improve the solution found by the greedy algorithm graph = {

ISSN:2153-182X, E-ISSN: 2153-1838
Vol. 17 No. 02 (2023)

```
    "a": {"b", "c", "d"},
    "b": {"a"},
    "c": {"a", "d"}
  "d": {"a", "c"}
}
min_paired_roman_domination = set()
while True:
    for vertex in min_paired_roman_domination:
        neighbors = set()
        for neighbor in graph[vertex]:
            if neighbor not in min_paired_roman_domination:
                neighbors.add(neighbor)
        if len(neighbors) > len(min_paired_roman_domination):
            min_paired_roman_domination.remove(vertex)
            min_paired_roman_domination.update(neighbors)
        else:
            break
print(min_paired_roman_domination)
```

This code will print the minimum paired roman domination of the graph, which is {"a", "c"}.

Here is a table showing local search and greedy algorithm.

Iteration	Vertex added to paired domination set	Vertex removed from paired domination set
1	a	b ,c d.
2	a	b,c,d,e
3	a	b, c d, e ,f

Comparison of greedy and local search algorithm

ISSN:2153-182X, E-ISSN: 2153-1838
Vol. 17 No. 02 (2023)

Simple algorithms known as greedy algorithms select options that are locally optimal at each phase. They can be quite effective for some issues and are frequently simple to execute. However, they occasionally have a tendency to become trapped in locally optimal solutions. Local search algorithms are more complex algorithms that iteratively improve a solution by making local changes. They frequently find better solutions than greedy algorithms by eluding local optima. They can, however, take more effort to set up and maintain.

Algorithm	**Advantages**	**Disadvantages**
Local search algorithm	Simple and efficient. Iteratively improves a solution by making local changes	Does not always find the optimal solution
Greedy algorithm	More likely to find the optimal solution. . Makes locally optimal choices at each step	More time-consuming

Conclusion: In this research paper we proposed a two algorithms to identify minimal and maximum paired roman dominations in graphs using the greedy algorithm and the local search algorithm. The greedy algorithm is used to find a minimum and maximum domination set, and local search algorithm is used to improved the solution. The proposed method is effective at determining the lowest and maximum paired roman dominations in graphs, according to experimental results on a set of test graphs.

Future work: Although the proposed method is faster than the local search technique by itself, it still takes a while for huge graphs. There are still a number of unresolved issues with the proposed work that need to be fixed. It is unclear how well the suggested work will perform on different graphs because it has only been tested on a small sample of graphs. In future large graph set is tested using greedy and local search algorithm.

References:

1. Cabrera Martínez A, Cabrera García S, Carrión García A. Further Results on the Total Roman Domination in Graphs. *Mathematics*. 2020; 8(3):349. https://doi.org/10.3390/math8030349

2.. Favaron, Odile & Karami, Hosein & Khoeilar, R. & Sheikholeslami, Seyed. (2009). On the Roman domination number of a graph. Discrete Mathematics. 309. 3447-3451. 10.1016/j.disc.2008.09.043.

3. Chellali, M., Jafari Rad, N., Sheikholeslami, S.M., Volkmann, L. (2020). Roman Domination in Graphs. In: Haynes, T.W., Hedetniemi, S.T., Henning, M.A. (eds) Topics in Domination in Graphs. Developments in Mathematics, vol 64. Springer, Cham.

4. Abdollahzadeh Ahangar, Hossein & Henning, Michael & Samodivkin, Vladimir & Gonzalez Yero, Ismael. (2016). Total Roman Domination in Graphs. Applicable Analysis and Discrete Mathematics. 10. 17-17. 10.2298/AADM160802017A.

5. Rupnik Poklukar D, Žerovnik J. Double Roman Domination: A Survey. *Mathematics*. 2023; 11(2):351. https://doi.org/10.3390/math11020351

6. A. Casado, S. Bermudo, A.D. López-Sánchez, J. Sánchez-Oro,
An iterated greedy algorithm for finding the minimum dominating set in graphs,
Mathematics and Computers in Simulation.

ISSN:2153-182X, E-ISSN: 2153-1838
Vol. 17 No. 02 (2023)

7.Mustapha Chellali, Teresa W. Haynes, Stephen T. Hedetniemi, Alice A. McRae,(2016), Roman {2}-domination, Discrete Applied Mathematics,
8. Morgan Kaufmann(2005) , In The Morgan Kaufmann Series in Artificial Intelligence, Stochastic Local Search,
9. Ernie J. Cockaynea , Paul A. Dreyer Jr.b;∗ , Sandra M. Hedetniemic , Stephen T. Hedetniemic (2023) Roman domination in graphs
10.Robert A. Beeler, Teresa W. Haynes, Stephen T. Hedetniemi, (2016).Double Roman domination
11. P. Roushini Leely Pushpam, S. Padmapriea (2016) , Global Roman domination in graphs.
12. M. Adabi E. Ebrahimi Targhi N. Jafari Rad M. Saied Moradi (2016).Properties of independent Roman domination in graphs.

Haynes, T., & Slater, P. (1998). Paired-domination in graphs Networks.

13. Dr. Shobha Shukla1 Vikas Singh Thakur (2020) "Domination and it's type in graph theory "

14. Jennifer M. Tarr(2010)" Domination in Graphs"

15. https://en.wikipedia.org/wiki/Dominating_set#Independent_domination

16. Desormeaux, Wyatt & Henning, Michael. (2014). Paired domination in graphs: A survey and recent results. Utilitas Mathematica. 94. 101-166.

17. Johri, S., Rajagopal, B. R., Ahamad, S., Kannadasan, B., Dixit, C. K., & Singh, P. (2023). Cloud computing based renewable energy demand management system. PROCEEDING OF INTERNATIONAL CONFERENCE ON ENERGY, MANUFACTURE, ADVANCED MATERIAL AND MECHATRONICS 2021.
18. RajBalaji, S., Raman, R., Pant, B., Rathour, N., Rajagopa, B. R., & Prasad, C. R. (2023, January 27). Design of deep learning models for the identifications of harmful attack activities in IIOT. 2023 International Conference on Artificial Intelligence and Smart Communication (AISC).
19. Malathi, M., Muniappan, A., Misra, P. K., Rajagopal, B. R., & Borah, P. (2023). A smart healthcare monitoring system for patients using IoT and cloud computing. PROCEEDING OF INTERNATIONAL CONFERENCE ON ENERGY, MANUFACTURE, ADVANCED MATERIAL AND MECHATRONICS 2021.
20. Ahdal, A. A., Rakhra, M., Rajendran, R. R., Arslan, F., Khder, M. A., Patel, B., Rajagopal, B. R., & Jain, R. (2023, February 8). Monitoring Cardiovascular Problems in Heart Patients Using Machine Learning. Journal of Healthcare Engineering; Hindawi Publishing Corporation.
21. Banu, S. R., Rajagopal, B. R., Venkatesan, K., & Rawat, P. (2023, May 10). Smart Financial Management System Based on Integrated Artificial Intelligence and Big Data analytics. ResearchGate.
https://www.researchgate.net/publication/370652400_Smart_Financial_Management_System_Based_on_ Integrated_Artificial_Intelligence_and_Big_Data_analytics
22. Rajagopal, B. R., Anjanadevi, B., Tahreem, M., Kumar, S., Debnath, M., &Tongkachok, K. (2022). Comparative Analysis of Blockchain Technology and Artificial Intelligence and its impact on Open Issues of Automation in Workplace. 2022 2nd International Conference on

ISSN:2153-182X, E-ISSN: 2153-1838
Vol. 17 No. 02 (2023)

Advance Computing and Innovative Technologies in Engineering (ICACITE), 288-292.
23. Rajagopal, B. R.,Kannapiran, E., Gupta, A. D., Momin, M. & Chakravarthy, D. S. K. (2022).The future prospects and challenges of implementing big data in healthcare management using Structural equation model analysis. Bull. Env. Pharmacol. Life Sci., (Spl Issue [1] 2022), 1111-1119
24. Krishnam, N. P., Ashraf, M. S., Rajagopal, B. R., Vats, P., Chakravarthy, D. S. K., & Rafi, S. M. (2022). ANALYSIS OF CURRENT TRENDS, ADVANCES AND CHALLENGES OF MACHINE LEARNING (ML) AND KNOWLEDGE EXTRACTION: FROM ML TO EXPLAINABLE AI. Industry Qualifications The Institute of Administrative Management UK, 58(Special Issue May 2022), 54-62.
25. Gupta, A. D., Rafi, S. M., Rajagopal, B. R., Milton, T., &Hymlin, S. G. (2022). Comparative analysis of internet of things (IoT) in supporting the health care professionals towards smart health research using correlation analysis. Bull.Env.Pharmacol. Life Sci., (Spl Issue [1] 2022), 701-708