A Project report submitted in
partial fulfillment of the requirement for the degree of

Bachelor of Technology
in
Computer Science Engineering

on

PSEUDO-COREs

Submitted By

Dheeraj Chaudhary (17BCS009)
Somyadeep Shrivastava (17BCS028)
Department of Computer Science Engineering

Under the guidance of

Dr. Yayati Gupta
Assistant Professor
Department of Computer Science and Engineering

Dharwad
ज्ञानेन विकासः

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY DHARWAD

Acknowledgement

We wish to express our gratitude towards Dr. Yayati Gupta, Assistant Professor, CSE Department, IIIT Dharwad for her valuable guidance and immense support throughout the working duration of this project. She helped us even in her busy schedule and made sure that all our queries are resolved.

We are thankful to her for showing faith in us and providing an opportunity to work on such a challenging project. Without her constant support this might not have been possible.

Dheeraj Chaudhary (17BCS009)

Somyadeep Shrivastava (17BCS028)

Table of Contents

Problem Statement 5

Background of the Problem 5

Our Contributions 6

Analysis 7

Analysis of Cascading Power : 7

Analysis of path algorithm to reach core : 8

Analysis of path algorithm to reach pseudo cores : 9

Analysis of Leakage Power : 11

Challenges Encounter 12

Future Work 13

Individual Contribution 14

References 15

Appendix 16

Appendix A: Dataset Overview 16

Appendix B: Plots generated from the Algorithms 17

Plot of Cascading Powers of Network that were used 17

Fig. (1.a) Cascading Power in Facebook Friendship Network 17

Fig. (1.b) Cascading Power in Weblog 17

Fig. (1.c) Cascading Power in Email-University 18

Comparison of Algorithms for Facebook friendship Network 18

Fig. (2.a) Comparison of Algorithms to reach Core 18

Fig. (2.b) Comparison of Algorithms to reach Pseudo-core node 19

Comparison of Algorithms for Minnesota's-road Network 19

Fig. (3.a) Comparison of Algorithms to reach Core node 19

Comparison of Algorithms for Email-University Network 20

Fig. (4.a) Comparison of Algorithms to reach Core node 20

Fig. (4.b) Comparison of Algorithms to reach Pseudo- core node 20

Comparison of Algorithms for Weblog Network 21

Fig. (5.a) Comparison of Algorithms to reach Core node 21

Fig. (5.b) Comparison of Algorithms to reach Pseudo-core node 22

Visualization of Leakage Powers in Facebook friendship Network 23

Fig. (6.a) Leakage Power vs Shell Number 23

Fig. (6.b) Leakage Power vs Node Count 23

Fig. (6.c) Leakage Power vs Density 24

Visualization of Leakage Powers in Email-University Network 24

Fig. (7.a) Leakage Power vs Shell Number 24

Fig. (7.b) Leakage Power vs Node Count 25

Fig. (7.c) Leakage Power vs Density 25

Visualization of Leakage Powers in Weblog Network 26

Fig. (8.a) Leakage Power vs Shell Number 26

Fig. (8.b) Leakage Power vs Node Count 26

Fig. (8.c) Leakage Power vs Density 27

Problem Statement

The main objective of the project is to study and examine thoroughly the virality of a meme with respect to network structure divided into shells by applying k-shell decomposition. Determining the spreading power of a shell by analyzing cascading power of a shell and finding pseudo cores which have comparable spreading power as the core. Considering various properties of shells and implementing various path algorithms to traverse from periphery to core nodes. Comparing these path algorithms on the basis of the number of steps taken by a path finding algorithm to reach the core node, by considering every periphery node as an instance. Analyze the performance of these algorithms in both cases when core nodes are target or when the pseudo cores are considered as a target. Further in the project we study why some memes didn't go viral by looking at Leakage power of the shells.

Background of the Problem

Most of the time we have seen that for various product based companies who is looking up for their sales growth, political parties who wants their work and ideology to spread to maximum number of people government agencies who wants their new rules to reach the maximum possible citizens, or the organisation who needs to garner new traffic, always wants that somehow their stuffs (Let's say memes) to get viral in the country/world.

So, if we look at everything in a network form, considering entities as nodes and relationships between them as edges, then we can say that to make these memes viral, stands out to be an integral part of research in complex networks. Leaving a few exceptions, we can say that, virality power of the nodes which have the maximum number of relationships, representing them as core nodes, is much higher. So if these memes somehow reach these core nodes then it's going to be viral in most of the cases. But, **[1]** says that, there's ample possibility for other nodes too which are not core nodes, to viral a meme, we call them as Pseudo-cores. So, our objective is to find the path to reach these Pseudo cores in an efficient way to make our memes viral.

Our Contributions

First of all our project finds the cascading power of each shell for various datasets listed in **Table 1** of **Appendix A** by averaging out the infecting power of all nodes for a shell. The extra part we implemented other than the original paper is to also keep a track of deviation of cascading power from its mean value for each shell , Thus plots in Appendix B **Fig (1.a), Fig (1.b), Fig (1.c)** show average cascading power of each shell and how much is the deviation for each shell. We have implemented the DFS and Teleportation variant of the original shell and shell degree based path algorithm in the paper. The DFS algorithm searches for its neighbors on conditions and maintains a stack while the teleportation randomly chooses a node in the graph after all its neighbors become already visited in search space. A slight variation also has been done in the intersection part of the original algorithm while implementing teleportation as in traversing it first find the neighbours with maximum core and then take intersection with total number of unvisited nodes , thus limiting the search space and arriving at the solution fast because we want it to reach out to random nodes as soon as it can (so that it pick up a node of core shell or near to it) rather than keep on looking at the nodes in its vicinity. Further in the project we have visualized leakage power in three different ways - leakage power with respect to shell number , leakage power with respect to number of nodes in a shell and leakage power with respect to density of shell. We have implemented our own algorithm for calculation of leakage power as there was no such algorithm defined in the paper originally . All the algorithms discussed above are applied to four datasets of different categories - social network , road network , webgraph and interaction network . While the social network dataset(facebook) is the same as used in paper for analysis , the other three are different from the original paper.

Analysis

Analysis of Cascading Power :

In **Appendix B Fig (1.a)** , **Fig (1.b)** , **Fig (1.c)** shows plots for cascading power for various networks. After observing **Fig (1.a)** carefully we can clearly state that from around shell number 25 we can see that the graph becomes somewhat parallel to x-axis for facebook friendship network . Similar is the case for Fig **(1.b)** from shell number 6 and for it **Fig (1.c)** it is shell no **.** Thus we can conclude that it is not only core nodes which enjoy maximum infecting power , there are other shells too in the network which have comparable cascading power termed as pseudo - cores in the original paper . We can also see that deviation of cascading power from its mean position for shell number keeps on decreasing as the graph approaches the core node. Thus we can say nodes in pseudo cores and core shells enjoy somewhat the same power as their fellow nodes in the same shell . Henceforth every node in a core shell or pseudo core is important enough for virality of a meme.

Note: : for further sections we have used short naming convention for shell degree based algorithm as degree based algorithm.

Analysis of path algorithm to reach core :

Facebook - Friendship Network- Fig (2.a) of **Appendix B** display plot for comparison of algorithms to reach facebook network core node. Here we can see that degree based teleport covers almost 100 percent of the instances in less than equal to 28 steps. The shell based dfs cover all of its instances in less than equal to 130 steps. While the shell based teleport completes its instances in around 281 steps and the degree based dfs completes only its 80 percent of instances in 281 steps .

Road Network- Fig (3.a) of **Appendix B** display plot for comparison of algorithms to reach road network core nodes. Here we can see that degree based teleport covers 100 percent of the instances in less than equal to 4 steps. The shell based teleport covers 100 percent of instances in less than equal to 5 steps . While the shell based dfs completes only its

97 percent of instances around 7 steps and the degree based dfs completes only its 95 percent of instances in 7 steps.

Email-Uni Network - Fig (4.a) of **Appendix B** display plot for comparison of algorithms to reach email network core nodes. Here we can see that both degree based teleport and degree based dfs covers 100 percent of the instances in less than equal to 5 steps. While the shell based dfs completes its 100 percent of instances in less than 50 steps and the degree based dfs completes only all of its instances in 60 steps.

Weblog Network - Fig (5.a) of **Appendix B** display plot for comparison of algorithms to reach weblog network core nodes. Here we can see that both degree based teleport and degree based dfs covers 100 percent of the instances in less than equal to 6 steps. While the shell based dfs completes its 100 percent of instances in less than 9 steps and the degree based dfs completes only all of its instances in 11 steps.

Hence we can conclude that the degree based teleport path algorithm finds the shortest path to reach the core node in a minimum number of steps. The shell based teleport always takes much longer time to cover all of the instances, while the performance of shell based dfs and degree based is varying.

Analysis of path algorithm to reach pseudo cores :

Facebook - Friendship Network - Fig (2.b) of **Appendix B** display plot for comparison of algorithms to reach facebook network pseudo core nodes (i.e all nodes from 25th shell are pseudocore according to cascading power). All algorithms have the same performance uptill 85 percent of the instances that's why initially all of them coincide. Here we can see that degree based teleport covers almost 100 percent of the instances in less than 10 steps while degree based dfs does it for only 90 percent of instances. The shell based dfs cover instances in less than equal to 100 steps . While the shell based teleport completes 100 percent of the instances in around 40 steps.

Email-Uni Network - Fig (4.b) of **Appendix B** display plot for comparison of algorithms to reach email network psuedo core nodes (i.e all nodes from 7th shell are pseudocore according to cascading power). Here we can see all the four algorithms have the same performance as all of their line plot coincides.

Weblog Network - Fig 5.b of Appendix B display plot for comparison of algorithms to reach weblog network pseudo core nodes(i.e all nodes from 6th shell are pseudocore according to cascading power). Here we can see that both degree based teleport and degree based shell covers almost 100 percent of the instances in less than equal to 4 steps. The shell based dfs also cover instances in less than equal to 4 steps . While the shell based teleport completes its instances in around 5 steps.

Hence we can conclude that the degree based teleport path algorithm finds the shortest path to reach the pseudo core node. In many of the cases shell based teleport takes much longer time to cover all of the instances than the rest , while the performance of shell based dfs and degree based is varying.

The algorithms converge to target nodes much faster in case of pseudo cores than in case of core nodes. Structure of the network also plays an important role in performance of algorithms . As we can see in some cases various algos might have the same performance since it required very less number of steps to reach the target node since the graph might be well interconnected . The degree based teleportation algorithm above all has the best performance both for core and pseudo core as the target.

Analysis of Leakage Power :

Facebook Friendship Network - Fig (6.a) of **Appendix B** shows the plot for leakage power with respect to shell number, it can be clearly seen that intermediary shells have the highest leakage power while it gets zero as it tends to core node. **Fig (6.b)** shows plot for leakage power with respect to nodecount. While there are more points at less number of nodes , there is sparsity at large numbers of nodes . The leakage power is higher between 1-100 number of nodes. **Fig (6.c)** shows plot for leakage power with respect to density of the shell. Here the leakage power is maximum for density of order of 10^{-2} which looks like zero in graph while the leakage power is zero for shell having density almost 1 which clearly states that these are not having much outer connections hence they have least leakage power.

Email Uni Network - Fig (7.a) of **Appendix B** shows the plot for leakage power with respect to shell number, it can be clearly seen that shells near to periphery and intermediary shells have higher leakage power while it gets zero as it tends to core node. **Fig (7.b)** shows a plot for leakage power with respect to nodecount. Shells having node count between 80-150 have higher leakage power . **Fig (7.c)** shows plot for leakage power with respect to density of the shell. Here the leakage power is maximum for density of order of 10^{-2} which looks like zero in graph while the leakage power is zero for shell having density almost 1 which clearly states that these are not having much outer connections hence they have least leakage power.

Weblog Network - Fig (8.a) of **Appendix B** shows the plot for leakage power with respect to shell number ,it can be clearly seen that shells near to periphery and intermediary shells have higher leakage power while it gets zero as it tends to core node. **Fig (8.b)** shows a plot for leakage power with respect to nodecount. Shells having node count between 50-150 have higher leakage power . **Fig (8.c)** shows plot for leakage power with respect to density of the shell. Here the leakage power is maximum for density of order of 10^{-2} which looks like zero in graph while the leakage power is zero for shell having density almost 1 which clearly states that these are not having much outer connections hence they have least leakage power.

Thus we can conclude that the plot between shell number and leakage power will tend to zero as it reaches core node while it is high for intermediary or periphery nodes. The most intuitive plot is the plot between leakage power and density as those who have high density can't have more connection to outer shell henceforth less leakage power and here the meme might get stuck. However the plot between leakage power vs number of nodes shows varying results.

Challenges Encounter

Our very first challenge revolves around the fact that running the cascading algorithms on a large dataset (having nodes >= 2k and edges >= 2k) takes a lot of time for facebook friendship itself,it took us one day to get each shell cascading power. Thus we were bound to look for a dataset which was not that big.

Implementing the path based algorithm was similar to solving a logical problem but it was tough to check whether what we have implemented is correct as to do dry run takes a lot of time checking path what the algo gives and what we expected .

As the work is first of its kind so it is tough to find references(other than source paper) or source codes , henceforth everything has been implemented from scratch .

Future Work

After analyzing various networks this could be seen that finding the shortest path is the ultimate key to virality of a meme. While running the path based algorithms we observed there are some shells in which our path traversal somewhat gets stuck and takes a little time to jump to the next higher core shell. This could be because of that shell having nodes mostly connected to nodes of the same shell or as we have seen in the leakage power plots that some shells have less leakage power . Henceforth it is necessary that the path algorithms should be implemented keeping leakage power in mind.

The project demands some more techniques other than visualizing cascading power to determine who are exactly pseudo cores in the network. Furthermore it is needed to apply these techniques and build real world applications derived from these ideas which could be used by various companies and research scientists .

Individual Contribution

We divided the work equally , henceforth contribution on our part is the same for both of us towards the project.

Here is the division listed :-

17BCS009 - Implementing DFS algorithm both shell and degree based. Analyzing Facebook and road networks by applying path algorithms. Implementing cascading algo and visualizing plot for each network listed in **Appendix A**.

17BCS028 - Implementing Teleportation algorithm both shell and degree based. Analyzing weblog and email-uni network by applying path algorithms. Implementing Leakage power algo and visualizing plot for each network listed in **Appendix A.**

References

1. Yayati Gupta , Debarati Das , S.Iyengar (2016). Pseudo-Cores: The Terminus of an Intelligent ViralMeme'sTrajectory.10.1007/978-3-319-30569-1_16 https://arxiv.org/abs/1507.07833

Appendix

Appendix A: Dataset Overview

Below **Table 1** gives a short description of all the Datasets used in this Report.

Category	**Name**	**Description**
Social Network	Facebook	Facebook is the most popular Social Networking Site today. This dataset consists of anonymized friendship relations from Facebook. The network contains 4,039 nodes and 88,234 edges.
Road Network	Minnesota's-road	This dataset is a link structure of Minnesota's roads which is a midwestern US State. The network contains 2,643 nodes and 3,303 edges.
Interaction Network	Email-University	Email communication network at the University Rovira i Virgili in Tarragona in the south of Catalonia in Spain. Nodes are users and edges indicate that at least one email was sent. The network contains 1,134 nodes and 5.451 edges.
Webgraph	Weblog	This dataset consists of webgraphs made up of poll-blogs. The network contains 643 nodes and 2,280 edges.

Appendix B: Plots generated from the Algorithms

1. Plot of Cascading Powers of Network that were used

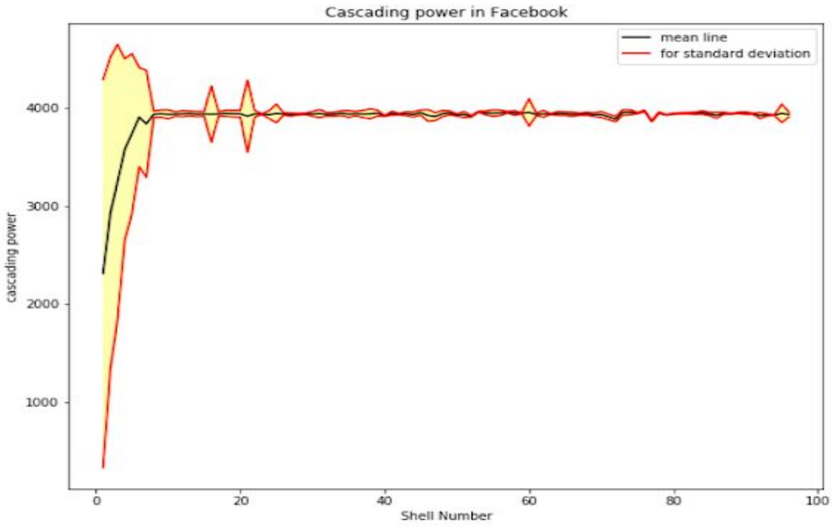

Fig. (1.a) Cascading Power in Facebook Friendship Network

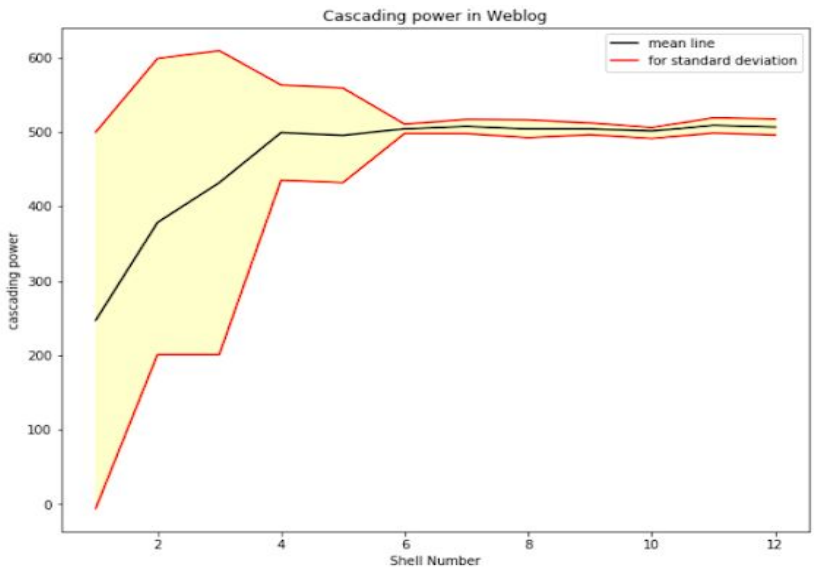

Fig. (1.b) Cascading Power in Weblog

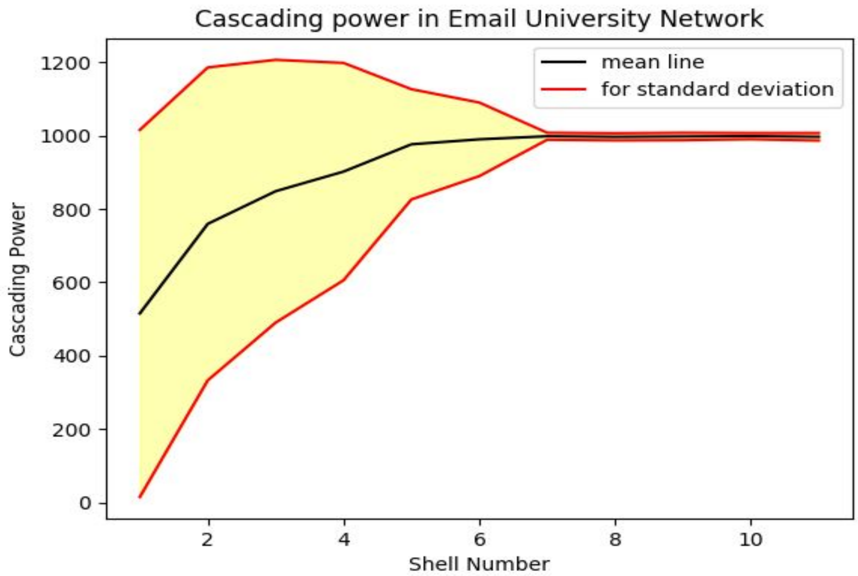

Fig. (1.c) Cascading Power in Email-University

2. Comparison of Algorithms for Facebook friendship Network

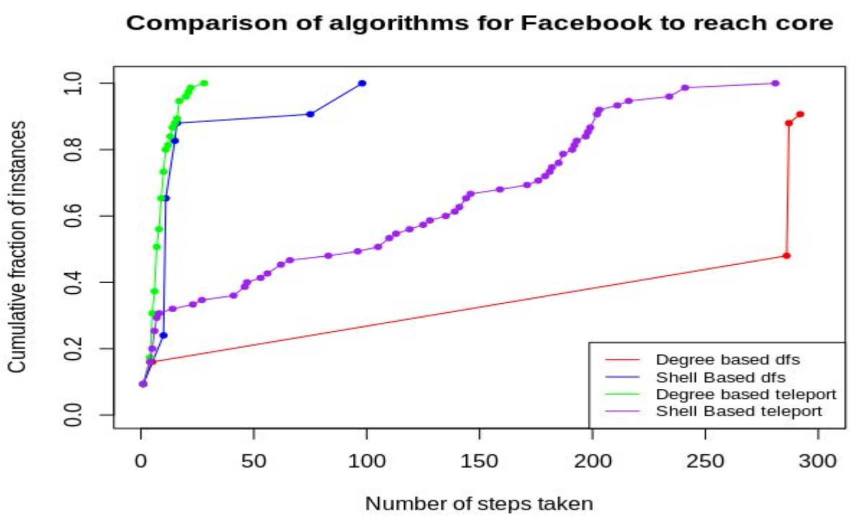

Fig. (2.a) Comparison of Algorithms to reach Core

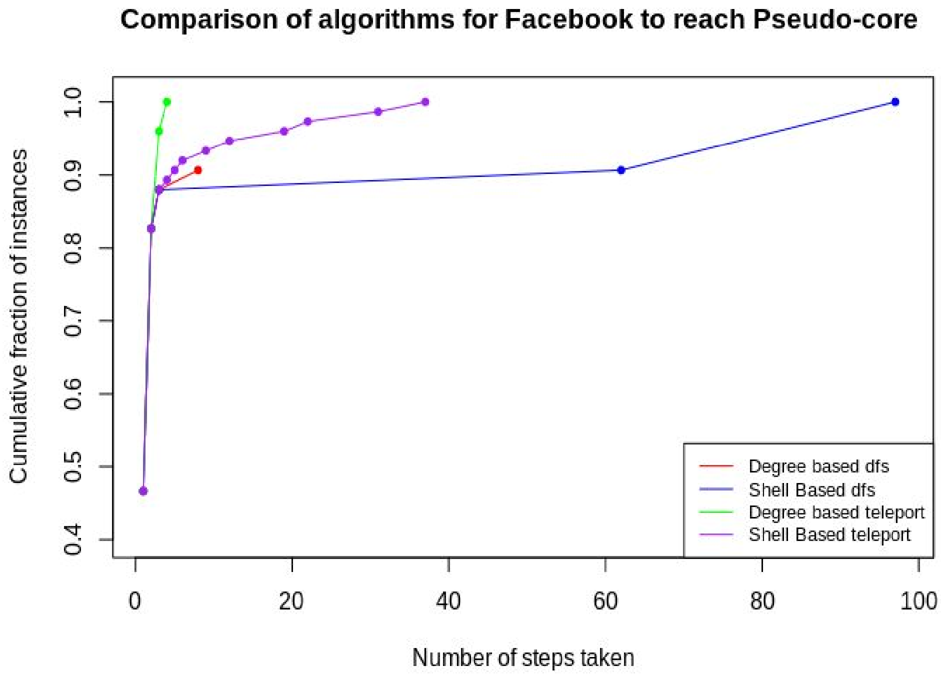

Fig. (2.b) Comparison of Algorithms to reach Pseudo-core node

3. Comparison of Algorithms for Minnesota's-road Network

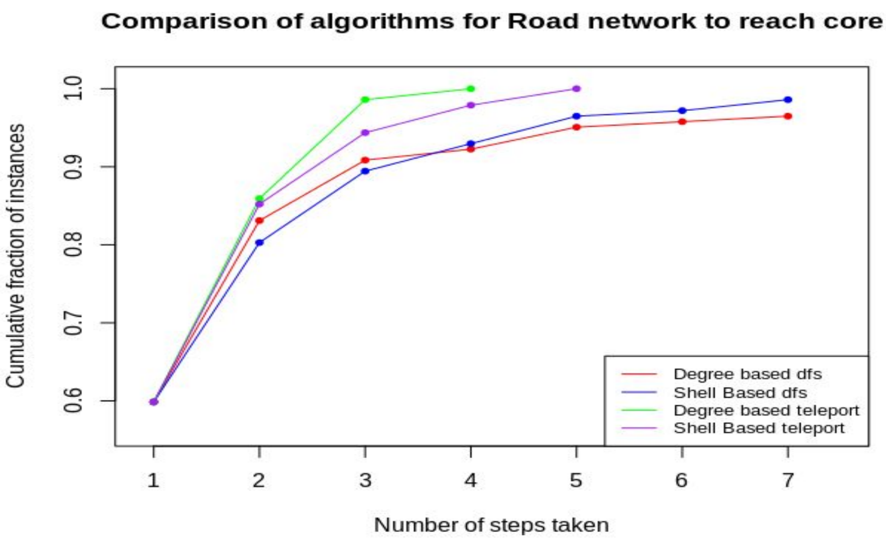

Fig. (3.a) Comparison of Algorithms to reach Core node

4. Comparison of Algorithms for Email-University Network

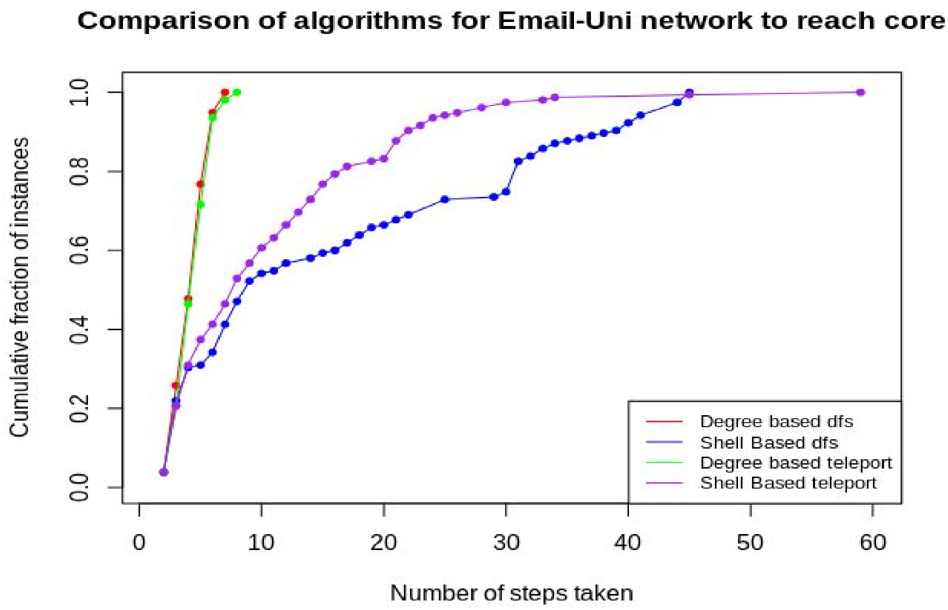

Fig. (4.a) Comparison of Algorithms to reach Core node

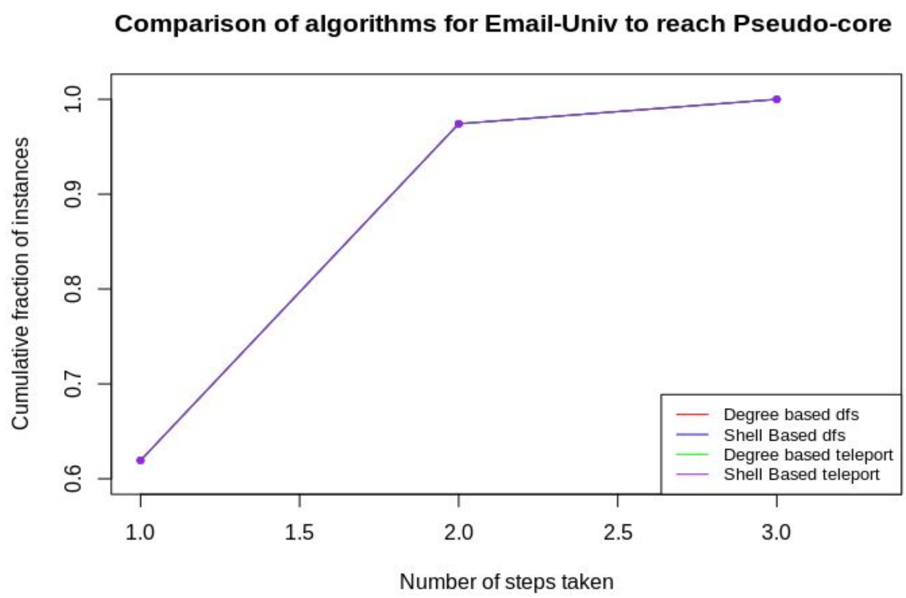

Fig. (4.b) Comparison of Algorithms to reach Pseudo- core node

5. Comparison of Algorithms for Weblog Network

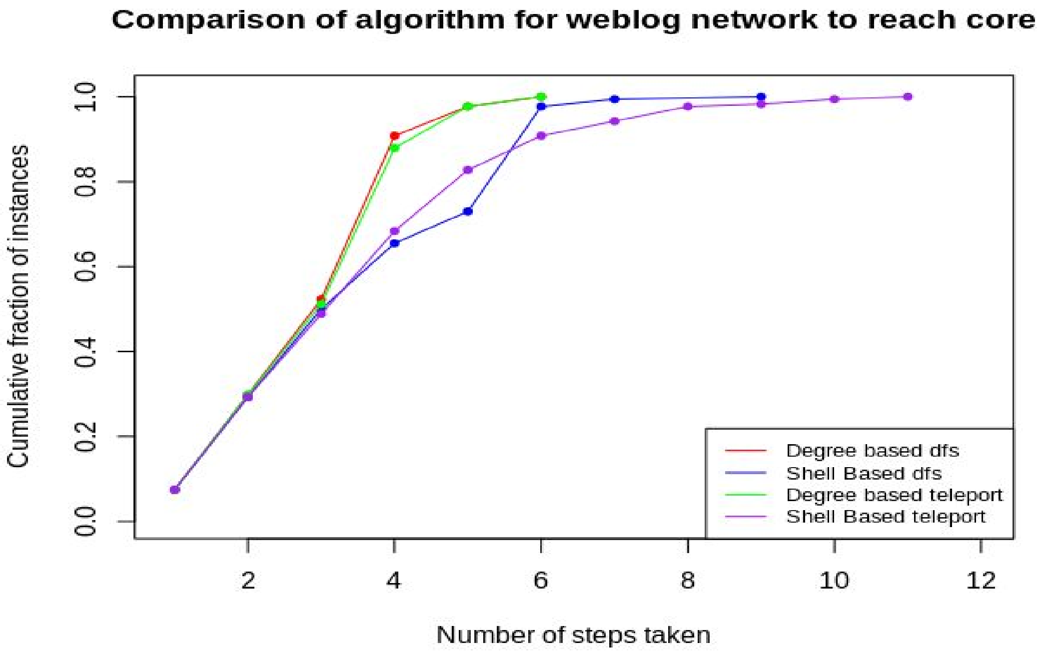

Fig. (5.a) Comparison of Algorithms to reach Core node

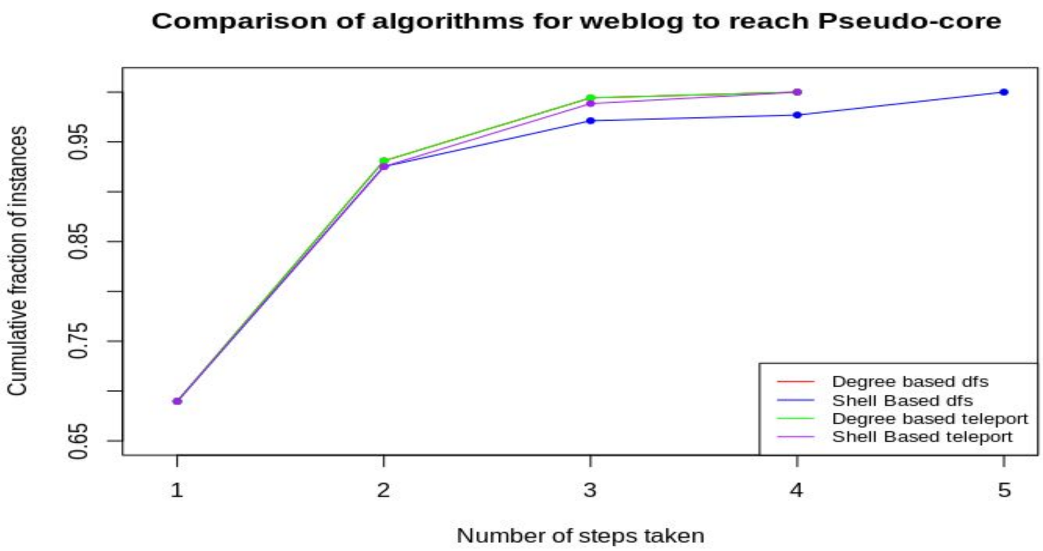

Fig. (5.b) Comparison of Algorithms to reach Pseudo-core node

6. Visualization of Leakage Powers in Facebook friendship Network

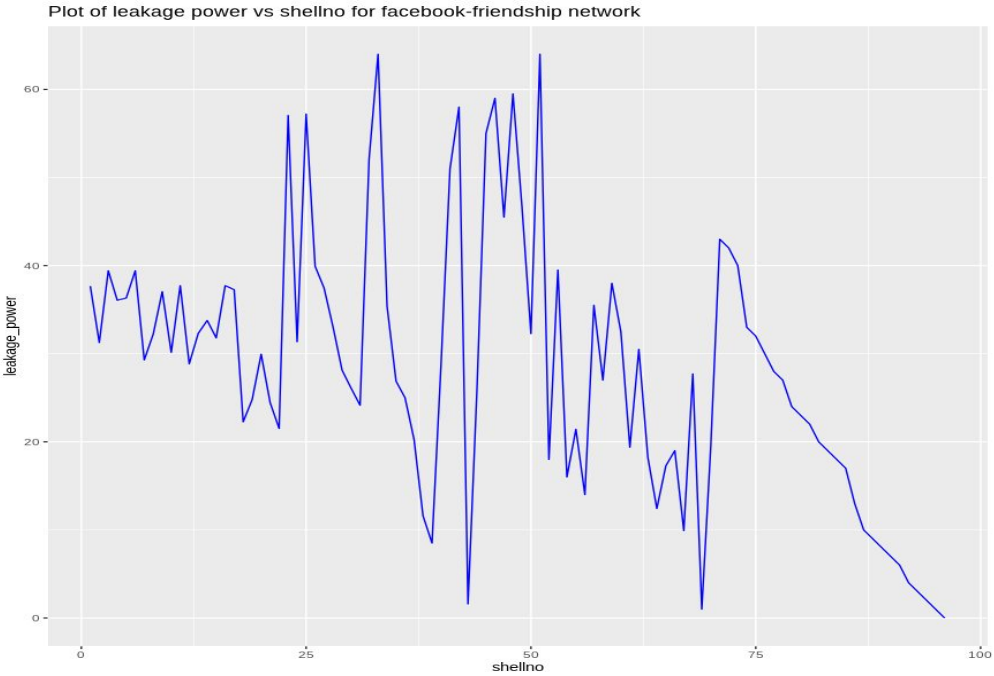

Fig. (6.a) Leakage Power vs Shell Number

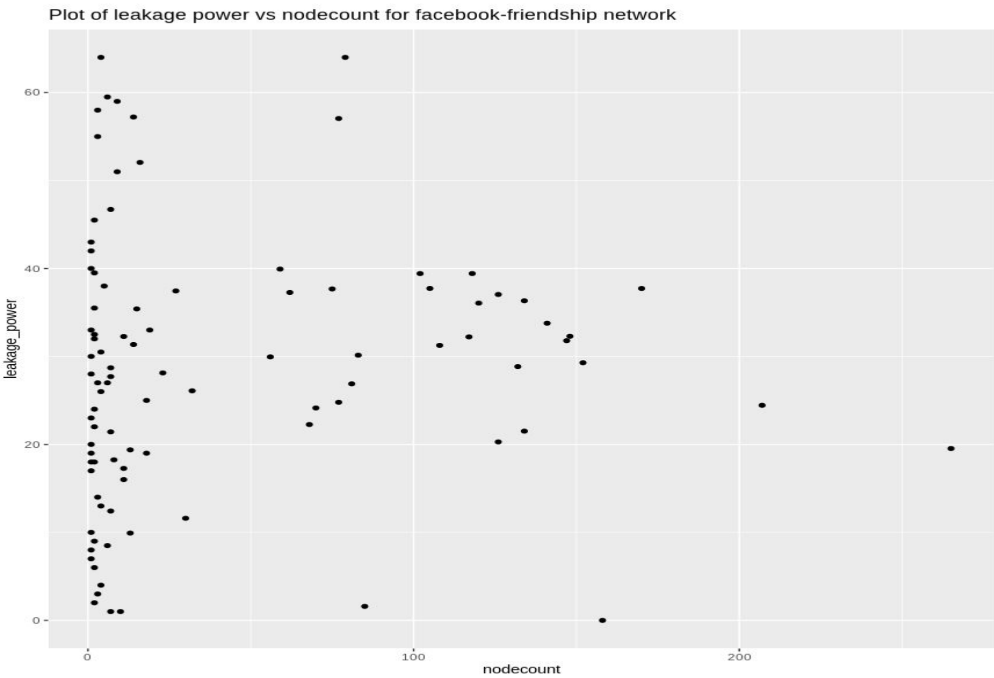

Fig. (6.b) Leakage Power vs Node Count

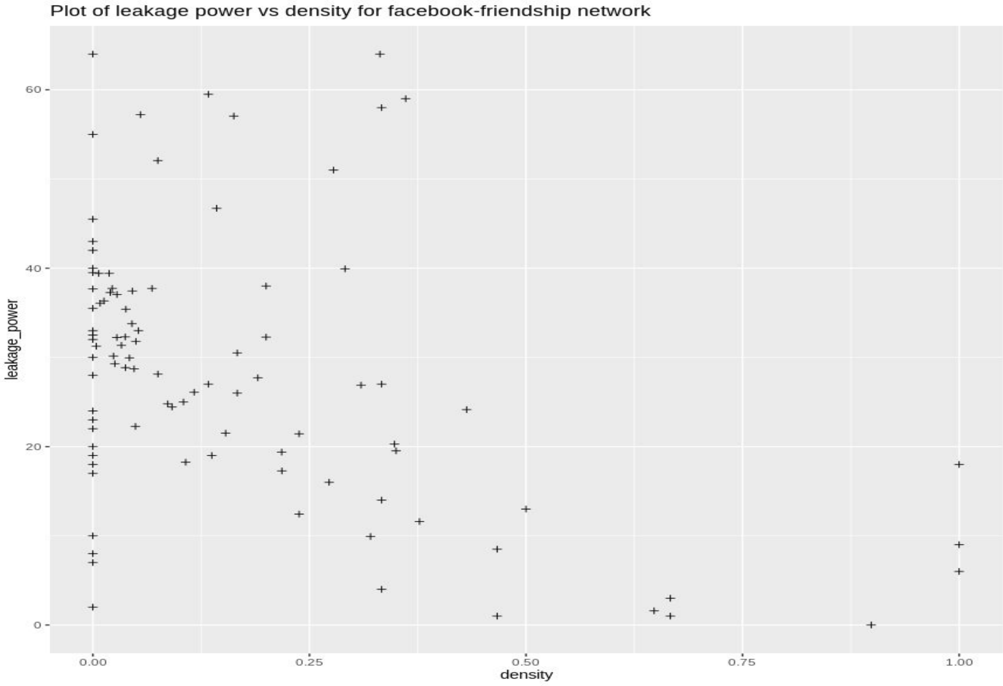

Fig. (6.c) Leakage Power vs Density

7. Visualization of Leakage Powers in Email-University Network

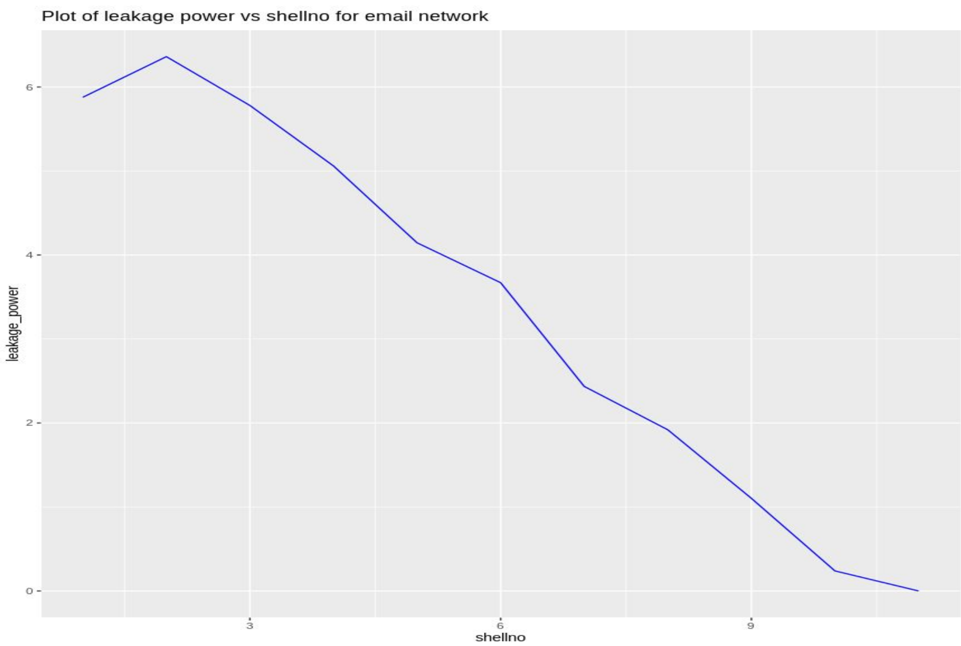

Fig. (7.a) Leakage Power vs Shell Number

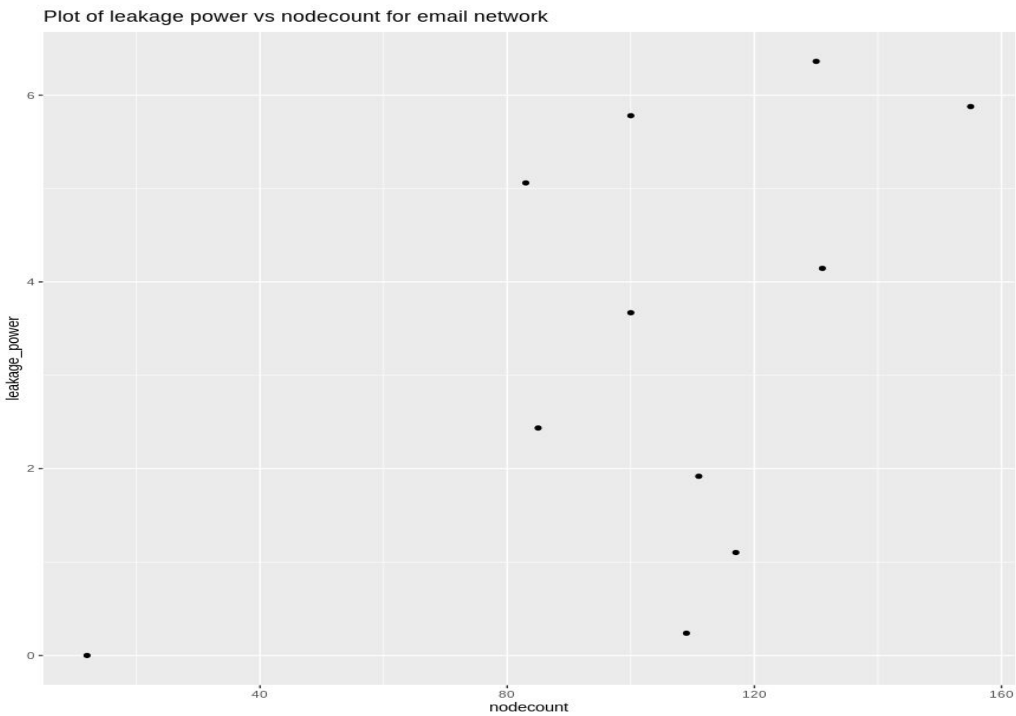

Fig. (7.b) Leakage Power vs Node Count

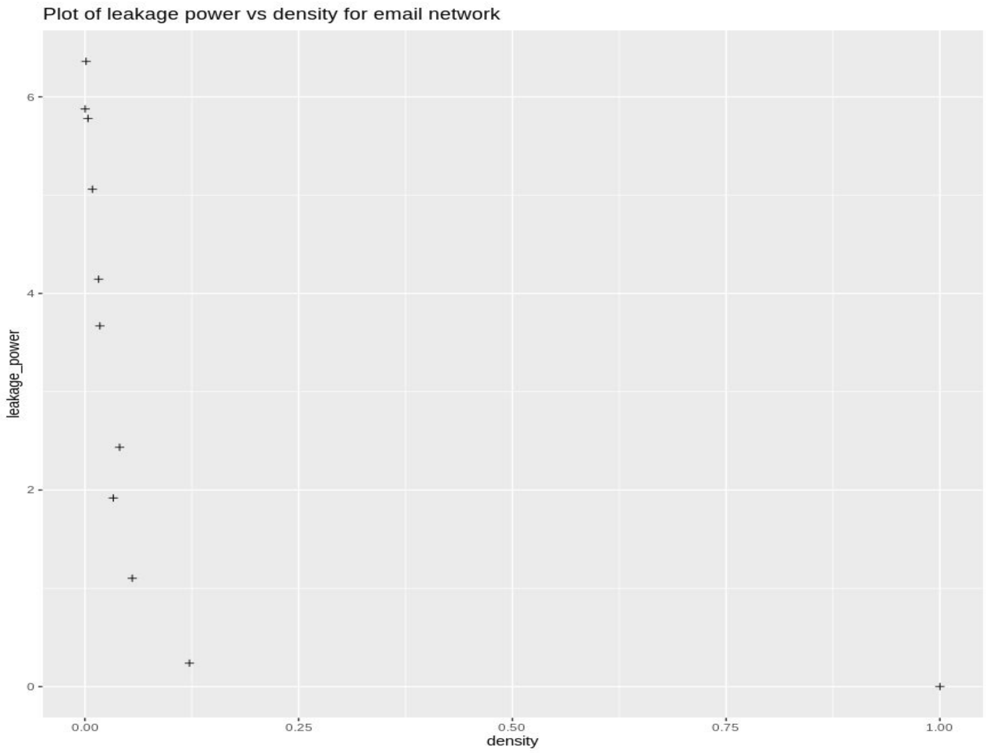

Fig. (7.c) Leakage Power vs Density

8. Visualization of Leakage Powers in Weblog Network

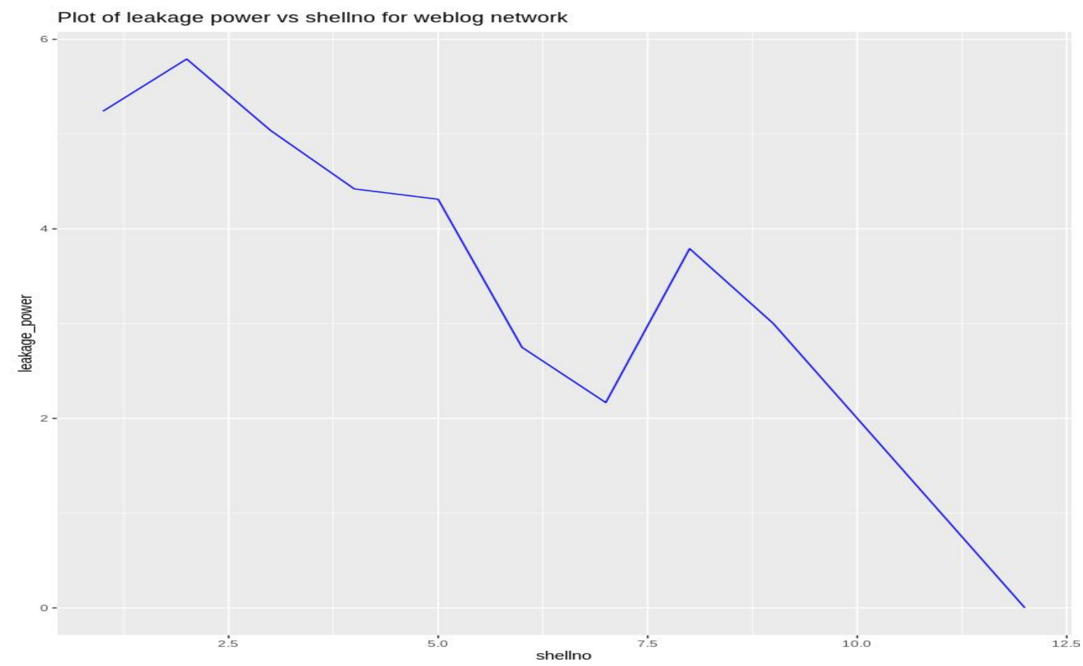

Fig. (8.a) Leakage Power vs Shell Number

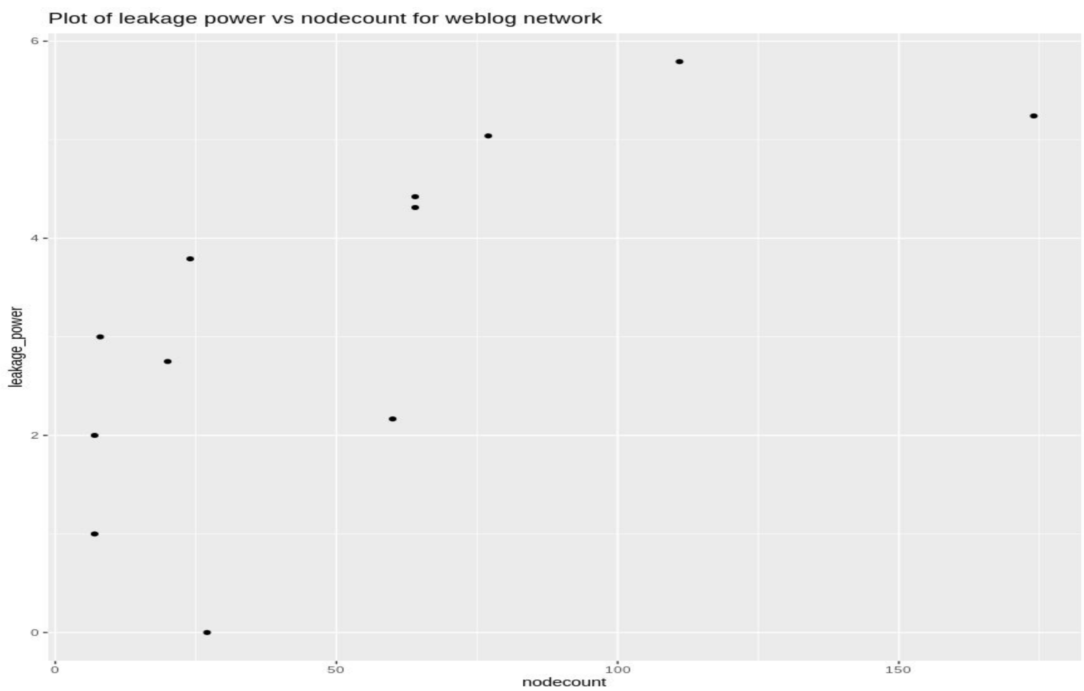

Fig. (8.b) Leakage Power vs Node Count

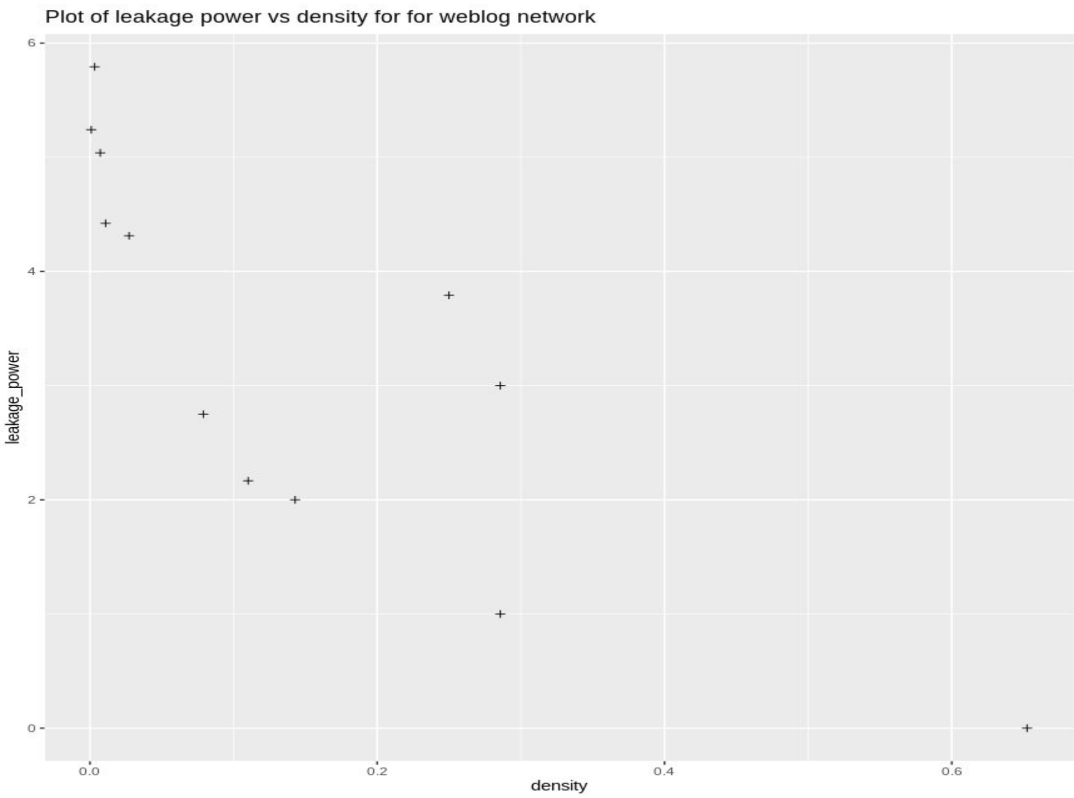

Fig. (8.c) Leakage Power vs Density