**INFORMATION RETRIEVAL**

**ASSIGNMENT**

**Community Detection as a Graph partitioning problem**

**for Friend’s community analysis**

**Submitted by:**

**Nistha Shukla - 2015H112175P**

**Farha Naaz - 2015H112177P**

**Rashmi Gulhane - 2015H112187P**

**Akansha Gupta - 2015H112181P**

1. **PROBLEM STATEMENT:**

The community detection problem involves grouping of similar users into clusters, where users in a group are strongly bonded with each other than the other members in the network.This is used in applications like detecting a spam community,finding communities in social network,etc.

1. **BACKGROUND OF PROBLEM:**
2. **Description of the selected application domain:**

A community is a subset of vertices which are *completely* connected to each other. In the technical parlance, a community is a subgraph which forms a *clique.*Our Community detection problem is concerned with finding an the friendship network among the bloggers. We have a repository that contains various facets of blog data including blog site metadata like, user defined tags, predefihow ned categories, blog site description; blog post level metadata like, user defined tags, date and time of posting; blog posts; blogger name; blog post comments; and blogger social network. From all the given data we try to find out community that have friendship and out of each community we can choose a person for publicity from each community.

1. **Motivation of the problem:**

Community detection is an important aspect in discovering the complex structure of social networks, and is of great importance in sociology, biology and computer science, disciplines where systems are often represented as graphs. A social network graph is a representation of the real world social network with the nodes representing the participating entities, or in this case, research papers or their authors, and the relations between these entities are represented by edges between them. By finding a sets of community we can find the common interest of the group of people, rising trends among the big communities can be found, spam community can be detected, ease of advertisement, etc. There are many applications of the community detection problem that motivated us to choose this problem.

1. **Technical issues**

The technical issues that we faced were running out of memory due to the fact that a large friendship network were formed on a large dataset. When these large networks are formed, the system crashed in the middle. But when the program is run on a smaller version of the dataset then it works completely fine and friendship network can be easily detected.

1. **RELATED WORK:**
2. **Literature survey**

Finding communities within an arbitrary network can be a [computationally](https://en.wikipedia.org/wiki/Computational_complexity_theory) difficult task. The number of communities, if any, within the network is typically unknown and the communities are often of unequal size and/or density. Despite these difficulties, however, several methods for community finding have been developed and employed with varying levels of success.

We analyzed many algorithms for community detection. Minimum-cut method is one of the oldest algorithms for dividing a network into parts. This method uses in-load balancing for parallel computing in order to minimize communication between processor nodes. So it is less than ideal for finding community structure in general networks. Now consider Hierarchical methods despite its simplicity, technique has several shortcomings. It tends to create clusters of single nodes in the outer regions of the graphs and there are no clear criteria regarding the most representative clusters that can be found in a dendrogram.

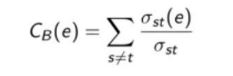
One of the most widely used methods for community detection is modularity maximization. [Modularity](https://en.wikipedia.org/wiki/Modularity_(networks)) is a benefit function that measures the quality of a particular division of a network into communities. The modularity maximization method detects communities by searching over possible divisions of a network for one or more that have particularly high modularity. Popular modularity maximization approach is the [Louvain method](https://en.wikipedia.org/wiki/Louvain_Modularity), which iteratively optimizes local communities until global modularity can no longer be improved given perturbations to the current community state.

Newman and Girvan’s proposed algorithm overcomes these issues. They brought a new concept, popularly known as ―**”edge betweenness”** to detect the community in large and complex networks. But the steps defined by Newman-Girvan will give us only dendrograms of the network. We stop at the level when we get maximum modularity. This basically is a mix of two approaches to achieve communities which overcomes the drawbacks of both of the approaches.

1. **SYSTEM DESCRIPTION AND MODULES:**
2. **Newman-Girvan algorithm (Edge-Betweenness)**

Newman ­Girvan algorithm focuses on the edges that connect communities. It is based on the simple idea that the shortest path between nodes from different communities will always include the edges that connect the different communities.

Betweenness value for any edge is calculated as:

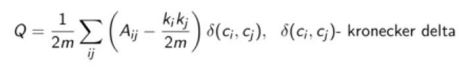


**Algorithm**

1. Find the edge of highest betweenness ­ or multiple edges of highest betweenness. Remove these edges from the graph. This may cause the graph to separate into multiple components. If so, this is the first level of regions in the partitioning of the graph.
2. Now recalculate all betweenness, and again remove the edge or the edges of highest betweenness. This may break some of the existing components into smaller components.
3. Repeat the above steps as long as edges remain in the graph.

As the algorithm proceeds, edges are removed from the graph, and so the graph splits into different connected components. These connected components can be thought of as different communities. In each iteration of the algorithm, we find the modularity value for the connected components (communities). The set of connected components with the highest modularity value is the final desired output.

The modularity score is used as a measure for community “quality”. Higher the modularity score, the better the communities. The modularity score ranges between [­0.5, 1), and is calculated as:



**Pseudo Code**

**Algorithm:** Edge Betweenness

**Input**: graph G(V,E)

**Output:** Dendrogram

**repeat**

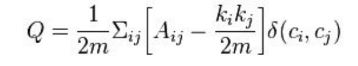
For all **e E** compute edge betweenness **CB(e) ;**

Remove edges ei with largest **CB(ei) ;**

**until** edges left;

**2) Louvain Method (Maximising modularity)**

The Louvain algorithm partitions the graph by optimising the ​graph modularity measure. Modularity is a scale value between ­-1 and 1 that measures the density of edges inside communities to edges outside communities. For weighted graphs, this measure can be calculated as:

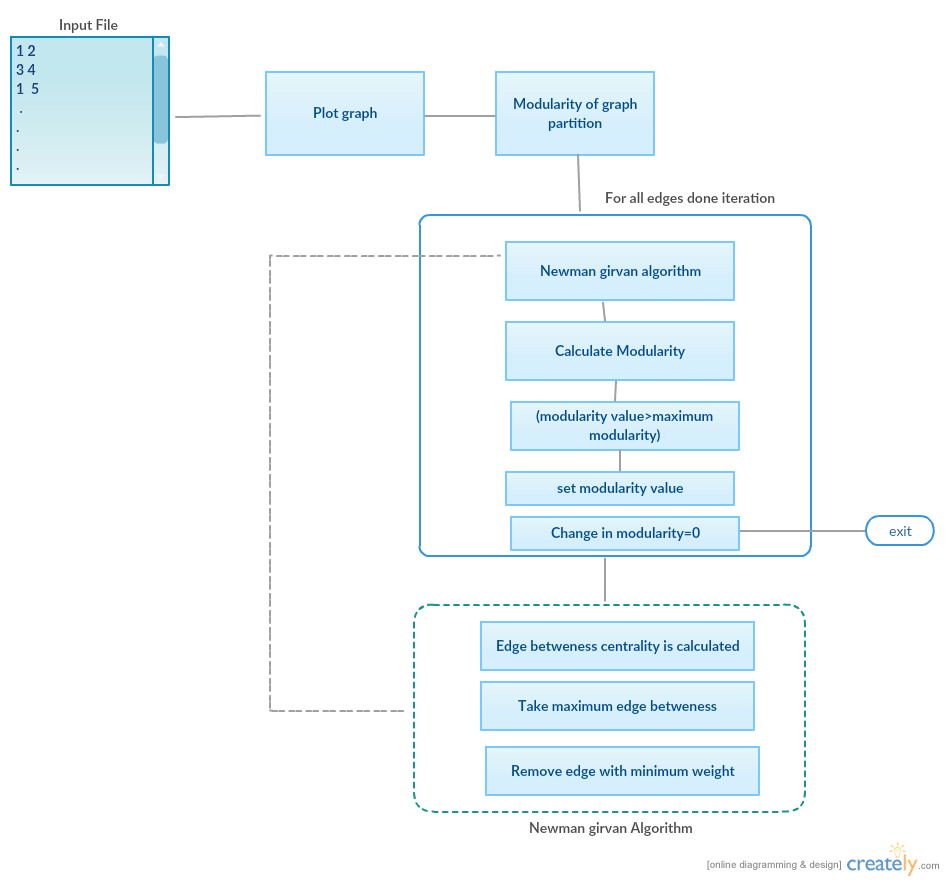


Where Aij represents the edge weight between nodes i and j. ki and kj are the sum of the weights of the edges attached to nodes and respectively. m is half the sum of all edge weights in the graph. ci and cj are the communities of the nodes.

The algorithm iterates, repeatedly through the following two steps, until the change in modularity is close to nill.

**3) GUI Designing:**

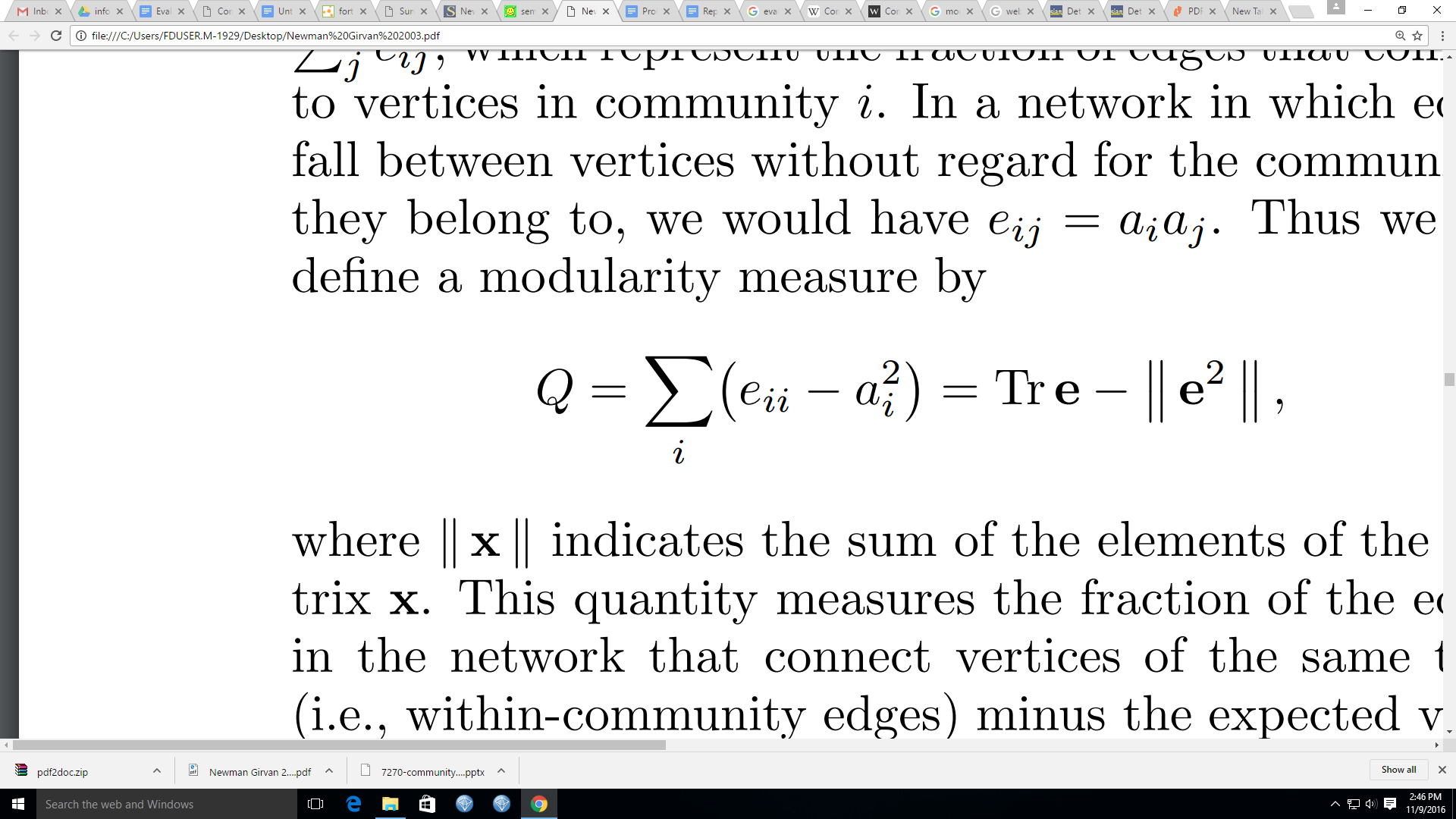
The Graphical user interface was implemented in python to provide the users a easy and interactive environment. The user browses for the input file and then gets the communities in the graph as results. Also for the ease of user,the results are plotted onto a graph.



1. **EVALUATION STRATEGY:**

The evaluation strategy used are:

1. **Modularity:** This measure define the quality of a particular division of the network.



Here tr e is the trace of the matrix i.e k ×k symmetric matrix e whose element eij is the fraction of all edges in the network that link vertices in community i to vertices in community j. K is defined as the total number of communities that are being formed. ||e||indicates the sum of the elements of the matrix . The row (or column) sums ai = j eij , which represent the fraction of edges that connect to vertices in community i. If the number of within-community edges is no better than random, we will get Q = 0. Values approaching Q = 1, which is the maximum, indicate strong community structure

|  |  |
| --- | --- |
| **Nodes** | **Modularity Value** |
| 50 | 0.68112 |
| 100 | 0.49653 |
| 300 | 0.72659 |

1. **Internal Density:** Density is defined by the number of edges (ms) in subset S divided by the total number of possible edges between all nodes (ns(ns-1)/2). The "2" is there to cancel out duplicated edges.

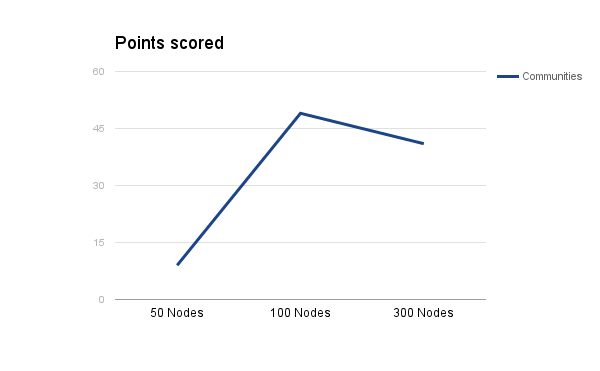
(We have found the internal density of the most dense community from the entire set) .Internal density for 50,100,300 nodes is 0.3,0.13 and 0.011

**Internal Density = \_\_\_\_ms\_\_\_\_\_**

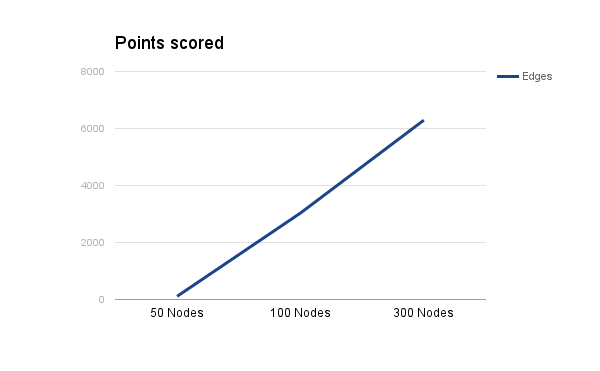
**(ns(ns-10)/2)**

1. **RESULTS:**

**Number of Nodes Vs Communities**



**Number of Nodes Vs Edges**

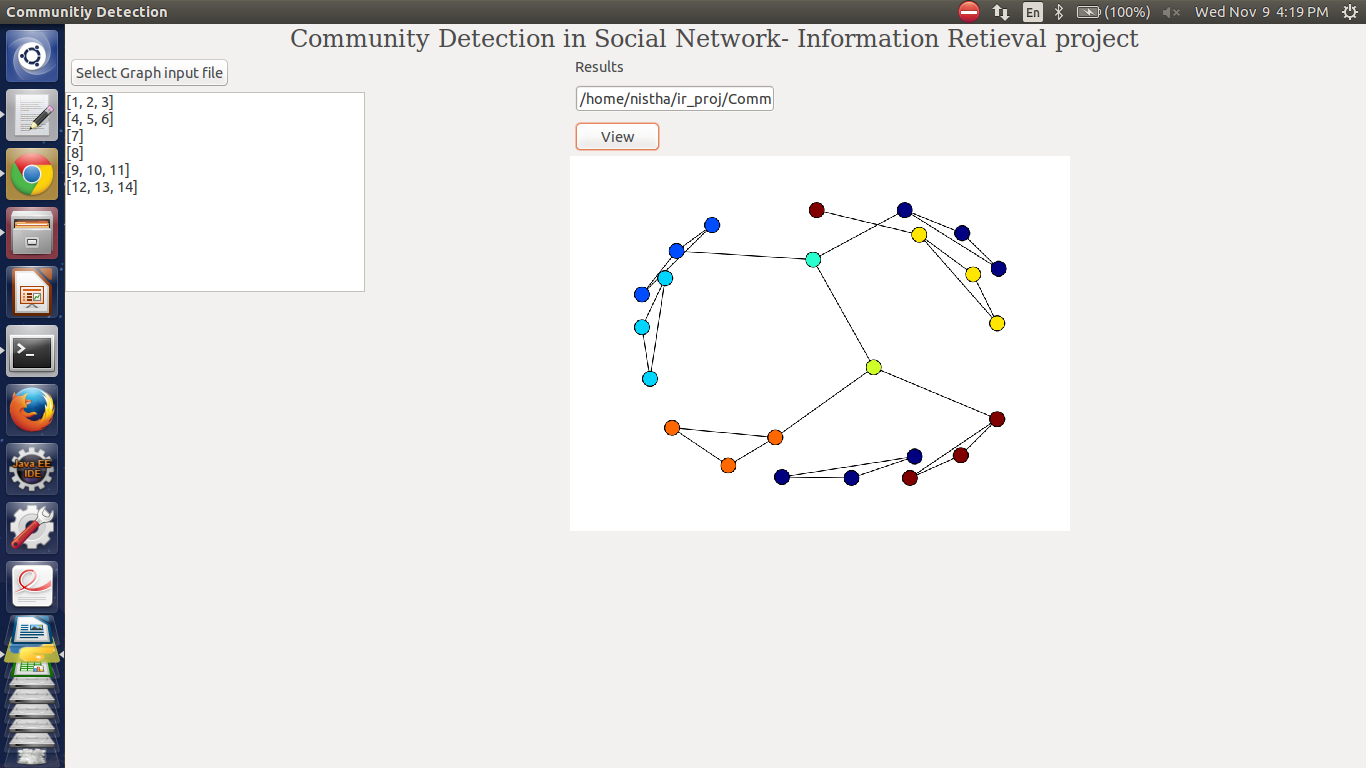


**Screen Shots:**

**Initial Selection of input graph and intermediate results**

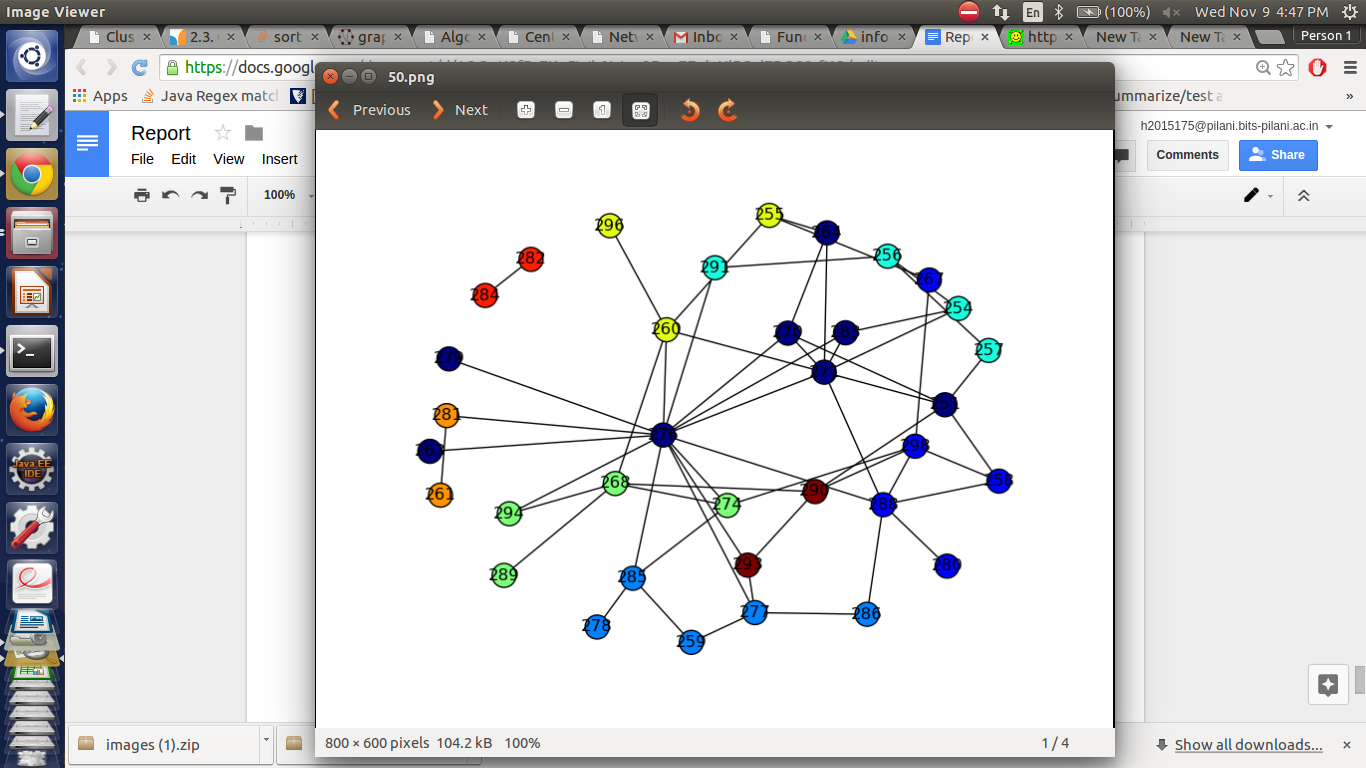
|  |  |
| --- | --- |
|  |  |

**Graphical User Interface for the Application**

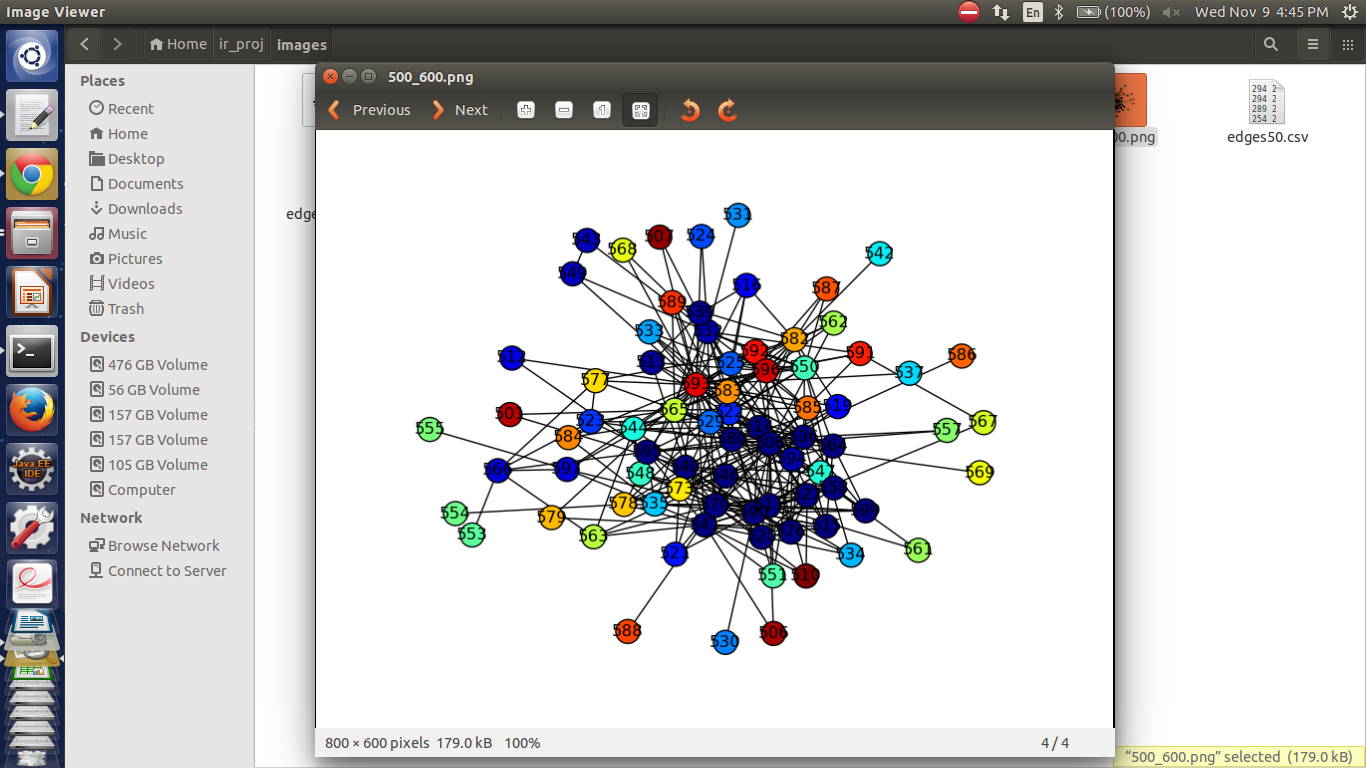


**Communities**

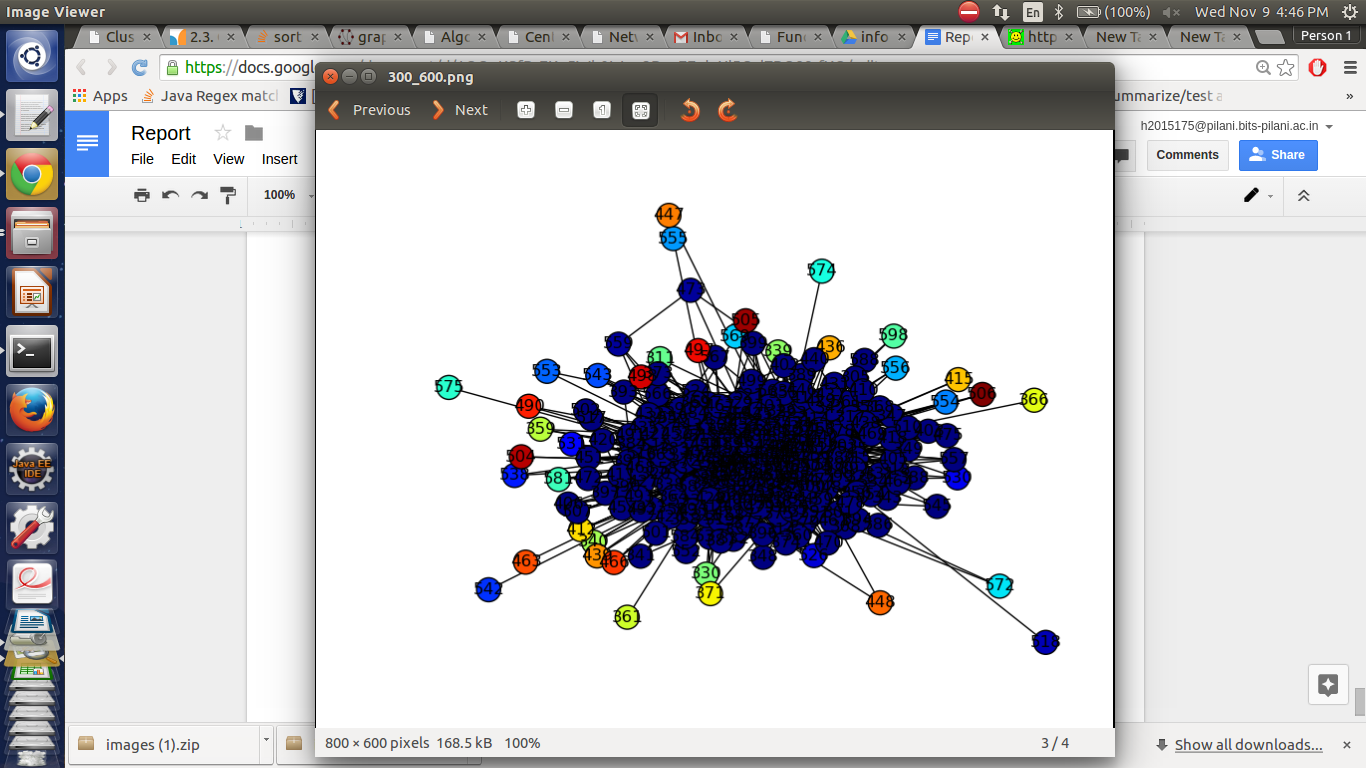
**Detected Communities for 50 Nodes**



**Detected Communities for 100 Nodes**



**Detected Communities for 300 Nodes**



1. **IMPROVEMENT SUGGESTED:**

Identifying influential spreaders in complex networks has a significant impact on understanding and control of spreading process in networks. It is essential to the acceleration of information diffusion, inhibition of gossip and spread of a virus. The measurement of node importance has been the key concern of researchers, and many indicators that are used to describe node importance have been successively proposed. After determining the communities and analyzing them, we can find out the influential nodes in the communities.

For the identification of the influential nodes, after calculating the different communities we determine the node which can cover the maximum depth in the community. For this we, for each community we select a node which has a max depth count. By this way, we can select the most important node in a cluster, which can be target for several publicity applications.

1. **CONCLUSION:**

The report describes the designing and implementation of the detection of communities in a social network. The communities are detected in the **blogcatalog dataset .** The communities detected show how the densely connected users fall under one community, which is determined by the application of newman girvan algorithm which removes edges by edge betweenness score along with modularity calculations for convergence. Also the report suggests evaluation of the communities determined and depicts improvements to the form of identification of influential node in the community.

1. **REFERENCES:**
2. M. E. J. Newman; Detecting community structure in networks.
3. **Louvain\_Modularity** https://en.wikipedia.org/wiki/Louvain\_Modularity
4. **A general method of community detection by identifying community centers with affinity propagation** Wei-Feng Guo, Shao-Wu Zhang Key Laboratory of Information Fusion Technology of Ministry of Education School of Automation, Northwestern Polytechnical University, Xi’an, 710072, China
5. **Community detection by consensus genetic-based algorithm for directed networks Stefano** B. B. R. P. Mathias a , Val ́erio Rosset a , Mari ́a C. V. Nascimento
6. **Discussion of the community detection algorithm based on statistical inference** Liangxun Shuo , Bianfang Chai Shijiazhuang University of Economics, Shijiazhuang 050031, China
7. **Dataset BlogCatalog**- http://socialcomputing.asu.edu/datasets/BlogCatalog

.