Analysis and searching of a heterogeneous publication network data.

CSC 698: Project in Computer Science

By:

Akhilesh Pathi

Advisors:

Dr. Lixin Fu, Dr.Chunjiang Zhu

Fall 2023

A logo of a university of north carolina

Description automatically generated

**Table of Contents**

1. Abstract……………………………………………………………………………………………………………………………….3
2. Introduction………………………………………………………………………………………………………………………………..3
3. Heterogeneous Network of Publication Data……………………………………………………………………………..4
4. Technologies and Architecture……………………………………………………………………………..………………..5
5. Dataset Loading and Preprocessing……………………………………………….……………………………………..5
6. Functionalities…………………………………………………………………………………………………………………………..6
7. Experimental Setup…………………………………………………………………………………………………………………..9
8. Node Pair Analysis…………………………………………………………………………………………………………………..10
9. Database………………………………………………………………………………………………………………………………..12
10. User Interface…………………………………………………………………………………………………………………………14
11. Conclusion……………………………………………………………………………………………………………………………….16
12. Future Work………………………………………………………………………………………………………………………….18
13. References……………………………………………………………………………………………………………………………….18

## 1.Abstract

This project, " Analysis and searching of a heterogeneous publication network data” aims to explore and visualize the complex networks of academic collaborations within the Association for Computing Machinery (ACM). Leveraging a blend of technologies including Python for data processing, Flask for web application development, MongoDB for database management, MATLAB for initial data handling, and HTML/CSS for frontend design, the project presents a comprehensive approach to understanding and analyzing academic networks.

Central to the project is the development of a web-based platform that allows users to interactively explore the ACM academic network. Using advanced data analysis techniques, such as community detection algorithms implemented in a Jupiter notebook, the project identifies and visualizes interconnected groups or communities within the ACM dataset. These visualizations are designed to reveal patterns of collaboration, highlight influential authors and publications, and provide insights into the structure and dynamics of academic research networks.

## 2.Introduction

In the realm of academic research, the intricate web of collaborations, citations, and publications forms a complex and dynamic network. This thesis delves into the analysis and searching of such a heterogeneous publication network, with a specific focus on the Association for Computing Machinery (ACM) dataset. Heterogeneous networks in academic settings are not just a mere collection of nodes and links; they are a rich tapestry of intellectual collaborations, knowledge dissemination, and scholarly influence. By exploring this network, we aim to unearth patterns and structures that are not immediately apparent in traditional academic analyses.

The impetus for this study stems from the increasing importance of understanding and navigating the complex landscape of academic research. In a world where interdisciplinary collaborations are becoming the norm, and the volume of published research is growing exponentially, there is a pressing need to develop tools and methodologies to analyze and search these vast networks effectively. This thesis responds to this need by applying advanced community detection algorithms to the ACM publication network and developing a web-based platform for its exploration and visualization.

**2.1 Project Background**

Academic networks are complex and multifaceted, often encompassing intricate patterns of collaboration that are not immediately apparent. These networks can be modeled as graphs, where nodes represent entities such as authors or publications, and edges represent relationships like co-authorship or citation. Analyzing these networks can provide valuable insights into how communities form around specific research areas, how information flows within the academic landscape, and how influential researchers contribute to their fields.

**2.2 Objectives**

The primary objective of this project is to apply community detection algorithms to the ACM dataset to identify and visualize these hidden structures. Community detection, a fundamental task in network

analysis, involves partitioning a network into communities or groups where nodes are more densely connected to each other than to nodes in other groups. This project aims to:

* Process and analyze the ACM graph data to detect underlying community structures.
* Develop a web-based platform that provides an interactive and user-friendly interface for exploring these communities.
* Offer insights into the patterns of collaboration within the ACM network, potentially aiding in the identification of key research areas and influential scholars.

**2.3 Significance**

By accomplishing these objectives, this project not only contributes to the academic community by providing a tool for exploring and understanding the dynamics of academic collaborations but also serves as a case study demonstrating the application of network analysis and web technology in the field of data science. The methodologies and findings of this project have the potential to be applied to other academic databases, thereby broadening our understanding of collaboration patterns across various fields of research.

## 3.Heterogeneous Network of Publication Data

The focus of this research lies in the exploration of a heterogeneous network, specifically within the context of academic publications. A heterogeneous network is characterized by its diverse types of nodes and edges, representing various elements and relationships within a system. In the realm of academic publications, these networks encapsulate a multitude of entities such as authors, papers, conferences, and citations, each linked through various relational dynamics.

Our study zeroes in on the Association for Computing Machinery (ACM) dataset, a comprehensive collection encapsulating a wide array of academic interactions. This dataset is a rich repository of information, encompassing numerous publications, authors, and their interrelations. It provides a unique opportunity to examine the multifaceted nature of academic collaborations and intellectual influences across different domains. By analyzing this dataset, we can uncover patterns of collaboration, trace the flow of ideas, and understand the evolving trends in computer science research.

The heterogeneous nature of this network poses both challenges and opportunities. The complexity of relationships and the diversity of entities demand sophisticated analytical approaches. This study addresses these complexities by employing advanced data processing techniques to unravel the underlying structures and dynamics of the ACM network.

## 4.Technologies and Architecture

The technological framework and architectural design of this project are pivotal in addressing the complexities inherent in analyzing a heterogeneous publication network. The project leverages a suite of technologies and tools to manage, process, and visualize the data effectively.

1. **Data Processing and Analysis**: Python, renowned for its powerful libraries and tools for data science, serves as the backbone for data processing and analysis. Libraries such as Pandas for data manipulation, NetworkX for graph analysis, and Matplotlib for data visualization, are extensively used to handle the complexities of the ACM dataset.
2. **Web-Based Platform Development**: The project features a custom-built web application, developed using Flask, a lightweight and flexible web framework in Python. Flask provides the necessary infrastructure to create an interactive and user-friendly interface for the analysis and visualization of the network data.
3. **Database Management**: To efficiently handle the extensive ACM dataset, a robust database management system is employed. Technologies like MongoDB, known for its scalability and flexibility in handling large datasets, are utilized to store and retrieve data seamlessly.
4. **Visualization Tools**: For effective representation of the complex relationships within the network, specialized visualization tools are incorporated. This includes integrating interactive graph visualization libraries that allow users to explore the network intuitively.
5. **Computational Environment**: The entire project is developed and tested in a high-performance computational environment, ensuring efficient handling of large-scale data and complex network computations.

## 5.Dataset Loading and Preprocessing

The dataset utilized in the project was sourced from the ACM (Association for Computing Machinery) and is contained within a MATLAB file named 'ACM.mat'. This file includes various matrices and arrays representing the complex relationships between papers, authors, and publication venues within the ACM repository.

Loading the Dataset

The dataset is loaded into the analysis environment using the **loadmat** function, which is designed to read MATLAB files. The structure of the loaded data is a dictionary with multiple keys corresponding to different types of data:

* **PvsA**: Represents a matrix of papers versus authors, indicating which authors have contributed to which papers.
* **PvsC**: A matrix of papers versus conferences, showing where each paper was presented or published.
* **PvsV**: Papers versus venues, which could include journals or conferences.
* **AvsF**: Authors versus fields, possibly indicating the fields of study or research areas of each author.

Additional keys such as 'RnormPvsA' and 'CnormPvsA' suggest that the dataset includes normalized representations of these relationships, which could be crucial for certain types of network analysis where the influence of the number of publications or co-authors is adjusted.

## 6. Functionalities

* **6.1 The Publication Dataset**

The foundation of this project is the publication dataset, specifically the ACM dataset. This dataset is a comprehensive collection of academic publications, authors, and their interconnections within the field of computing. It includes details such as paper titles, author names, publication venues, and citation information. The dataset's richness and complexity make it an ideal candidate for analyzing academic networks. Our system is designed to handle this dataset, enabling users to explore and analyze various aspects of the academic collaboration network.

* **6.2 Analysis and Pre-processing of Input Data**

Before any meaningful analysis, the input data undergoes a thorough pre-processing phase. This involves cleaning the data, handling missing or inconsistent entries, and transforming the data into a suitable format for analysis. Python scripts are used for this purpose, leveraging libraries such as Pandas for data manipulation. This step ensures the integrity and quality of the data, which is crucial for accurate and reliable analysis.

* **6.3 Search Application in Flask**

A significant functionality of the project is the search application developed using Flask, a Python web framework. This application allows users to query the publication dataset. Users can search for specific authors, papers, or keywords, and the system provides relevant results from the ACM dataset. This functionality enhances the accessibility and usability of the dataset, making it a valuable tool for researchers and academics.

* **6.4 Interface to Visualization Tool**

To aid in the interpretation and understanding of the complex relationships within the dataset, an interface to a visualization tool is developed. This tool uses technologies like pyvis to provide interactive and dynamic visualizations of the network. Users can visually explore connections between authors, trace citation networks, and discover community structures within the academic network.

* **6.5 Demo and Screenshots**

Input:

A screenshot of a computer

Description automatically generated

**Output:**

**A diagram of a paper

Description automatically generated**

**Input:**

A close-up of a computer screen

Description automatically generated

**Output :**

A screenshot of a book

Description automatically generated

**Input:**

**A screenshot of a computer

Description automatically generated**

**Output:**

**A screenshot of a search

Description automatically generated**

## 7.Experimental Setup

**7.1 Environment Configuration**

Our analysis was conducted in a controlled computing environment to ensure the reproducibility and reliability of the results. The primary tools and software used in the setup included:

* **Python Environment**: Python 3.x was selected for its extensive support for data analysis libraries.
* **MATLAB**: MATLAB R2021a was used for initial data extraction from **.mat** files.
* **Jupyter Notebook**: For interactive data manipulation and visualization.
* **Flask Web Framework**: Deployed for developing the interactive web application.
* **MongoDB**: Employed as the database to store and manage the graph data efficiently.

The computing environment was standardized across all experiments, with sufficient computational resources allocated to process the large ACM dataset.

**7.2 Data Preparation**

The ACM dataset was loaded from a **.mat** file, which contains the network graph represented as matrices and arrays encoding relationships between papers, authors, and venues. The data was subjected to the following preprocessing steps:

* **Normalization**: Matrices such as **PvsA** (papers vs. authors) were normalized to account for varying numbers of publications per author.
  + **Conversion**: Sparse matrices were converted to dense format when necessary to facilitate certain computations.
  + **Cleaning**: Data cleaning procedures were implemented to remove any anomalies or inconsistencies in the dataset.

**7.3 Experiment Design**

The experiments were designed to explore several facets of the ACM network:

* **Community Detection**: We employed algorithms such as modularity optimization and hierarchical clustering to identify communities within the network.
* **Collaborative Patterns**: Network metrics such as degree centrality and betweenness centrality were calculated to identify key nodes and collaboration patterns.
* **Temporal Analysis**: Where available, time-stamped data was used to understand the evolution of the network over time.

Each experiment was run multiple times to ensure consistency in the results. The outcomes were documented with precision, capturing details such as the size of detected communities, the number of nodes and edges analyzed, and the computational time taken for each process.

## 8.Node Pair Analysis:

Identifying node pairs with top and bottom similarity scores, and possibly visualizing or further analyzing these pairs.

**8.1 Finding edge relationships**

**A screen shot of a computer code

Description automatically generated**

**Output : A screen shot of a computer screen

Description automatically generated**

**Getting nodes for which, the triangle count is at least 1**

['Author\_27', 'Author\_130', 'Author\_195', 'Author\_208', 'Author\_239', 'Author\_243', 'Author\_257', 'Author\_272', 'Author\_366', 'Author\_499', 'Author\_517', 'Author\_569', 'Author\_584', 'Author\_607', 'Author\_617', 'Author\_644', 'Author\_688', 'Author\_711', 'Author\_732', 'Author\_741', 'Author\_749', 'Author\_786', 'Author\_803', 'Author\_809', 'Author\_853', 'Author\_887', 'Author\_976', 'Author\_982', 'Author\_1005', 'Author\_1021', 'Author\_1112', 'Author\_1295', 'Author\_1310', 'Author\_1339',so on.

**Counting the number of neighbors of each node**

Total number of neighbors of Venue\_104 : 46

Total number of neighbors of Venue\_107 : 249

**Getting node pairs with top12 and bottom 12 aggregated similarity scores.**

**Top 12**

[((18040, 8394), 0.6667), ((62334, 56628), 0.6269), ((22455, 2498), 0.625), ((49214, 56628), 0.5906), ((826, 55484), 0.589), ((8008, 5357), 0.5757), ((27757, 5357), 0.5731), ((55365, 22455), 0.5714), ((50093, 58380), 0.5654), ((14439, 58380), 0.558), ((31714, 50093), 0.5545), ((19446, 826), 0.5528)]

**Bottom 12**

[((55365, 30122), 0.1407), ((34455, 58380), 0.1399), ((37323, 30122), 0.1313), ((5879, 30122), 0.1306), ((3562, 53458), 0.1291), ((47671, 41252), 0.128), ((28816, 58380), 0.119), ((55365, 3562), 0.1154), ((3562, 983), 0.101), ((2498, 3562), 0.0842), ((37323, 3562), 0.0842), ((32267, 7360), 0.0589)]

## 9.Database:

**Usage of MongoDB in the Thesis Project**

In this project, MongoDB, a NoSQL database, plays a pivotal role in managing the complex and voluminous data associated with the heterogeneous publication network. The choice of MongoDB is driven by several of its inherent characteristics that are particularly beneficial for handling the ACM publication dataset:

1. **Handling Unstructured and Semi-structured Data:** The ACM dataset, like many publication datasets, is inherently unstructured or semi-structured with varied data formats (e.g., author names, publication titles, citation lists). MongoDB excels in storing this type of data due to its flexible document-oriented model.
2. **Scalability and Performance:** As the dataset encompasses a vast number of records with intricate relationships, MongoDB's ability to scale horizontally and handle large volumes of data efficiently is crucial. Its performance in terms of data retrieval and querying is particularly advantageous for the project's needs.
3. **Flexible Data Schema**: MongoDB does not require a predefined schema, allowing the data structure to be modified on the fly. This flexibility is essential in academic datasets where new fields or types of data may need to be incorporated as the research evolves**.**
4. **Efficient Data Retrieval:** MongoDB provides powerful query capabilities that enable efficient retrieval of complex data. This is particularly useful for the search functionalities of the web application, where users can query the database for specific authors, publications, or ids.
5. **Data Aggregation and Analysis:** MongoDB's aggregation framework allows for complex data processing and analysis directly within the database. This capability is used to perform preliminary analyses and prepare the data for further processing and visualization in the project.
6. **Integration with Python and Flask:** MongoDB's compatibility and ease of integration with Python and Flask – the core technologies used in this project – is a significant advantage. The PyMongo library provides a straightforward and effective way to connect MongoDB with the project's Python-based backend.
7. **Robustness and Reliability:** MongoDB offers robustness in data storage with features like automatic failover and data replication. This ensures the integrity and availability of the data, which is crucial for the reliability of the project.
8. **Support for Geospatial Queries:** Although not a primary requirement for this project, MongoDB's support for geospatial queries can be an added advantage for future expansions, especially if geographical analysis of publication data becomes relevant.

In this project, MongoDB is utilized as the backbone for data storage and management, facilitating efficient data handling and retrieval, which are essential for the analysis and visualization of the heterogeneous publication network.

**Demo:**

A screenshot of a computer

Description automatically generated

1. **\_id**: This is a unique identifier for the document, which is automatically generated by MongoDB when a new document is created. The **ObjectId** is a 12-byte BSON type, guaranteeing uniqueness within the collection. The format looks like **ObjectId('654c6fb4fc73e380dbaccd4a')**, where the value inside **ObjectId** is a hexadecimal representation of the identifier.
2. **type**: This field specifies the type of the document or the relationship it represents. In this case, the type is **"edge"**, which likely means this document is used to represent an edge in a graph database context.
3. **relation**: This field describes the kind of relationship the edge represents. The value **"AV"** could be an abbreviation or a code that stands for a specific type of relationship between two nodes in the graph. It needs to be interpreted in the context of the database schema or the application's domain logic.
4. **source\_node**: This field identifies the starting point or the source of the edge. In graph terminology, this would be the node from which the edge is emanating. Here, the source node is identified as **"Author\_70"**, which indicates that this edge is associated with the author node with an ID or name **Author\_70**.
5. **target\_node**: This field identifies the endpoint or the target of the edge. This is the node to which the edge is pointing. In this document, the target node is **"Venue\_0"**, suggesting that the edge points to a venue node with an ID or name **Venue\_0**.

Graphs play a crucial role in obtaining useful insights from intricate networks. They enable us to quantify the characteristics of nodes and edges, identify links, and comprehend the graph's general structure. Queries can be used to find influential authors, collaborative groups, and the significance of certain publications or venues within the framework of the ACM heterogeneous graph.

Using NetworkX for graph

A wide range of functions are available in NetworkX for graph . These features span from straightforward lookups, such obtaining a node's neighbours, to more intricate searches, including determining the shortest path between nodes or extracting subgraphs that satisfy requirements.

Insights from Queries The ability to query the graph effectively opens up a range of analytical possibilities. For instance, by querying the graph, we can identify the most prolific authors, the most cited papers, or emerging trends in research topics. These insights are not only valuable for academic researchers but also for institutions, publishers, and funding bodies looking to understand the dynamics of academic collaboration and knowledge creation.

## 10.User Interface

1. Importing Flask Blueprint: It imports Blueprint from Flask, a micro web framework written in Python.
2. Blueprint Creation: A Flask Blueprint named main\_bp is created. Blueprints are used in Flask to organize a group of related views and other code. The name 'main' implies that this Blueprint might be the main or central part of the web application.
3. Importing Routes: It imports routes from the current package. This indicates that the routing logic for the web application (which defines how different URLs are handled) is organized in a separate file or module, likely routes.py.

A screenshot of a search engine

Description automatically generated

**Author and Paper Search Functionality**

One of the key components of the web application developed in this project is the "Author and Paper Search" feature. This interface allows users to conduct targeted searches within the ACM dataset to find relevant information about authors, papers, and publication venues. The functionality is realized through a simple and intuitive user interface, providing the following search options:

* **Selected Authors**: This input field allows users to search for specific authors within the database. The functionality is likely designed to filter search results based on author names or identifiers inputted by the user.
* **Paper Search**: Users can utilize this search box to query the database for papers. This could be based on paper titles, keywords, or topics, enabling researchers to find publications related to their area of interest.
* **Venue Search**: This feature facilitates the search for specific venues. It can help users explore the repository of publications within particular conferences or journals, thus providing insights into the venues' research themes and significance.
* **Author Search**: Distinct from the "Selected Authors" field, this likely serves a broader search, possibly offering autocomplete suggestions as the user types, and could include additional metadata filters such as the author's affiliations or research areas.

**Co-authorsip relation**

A screenshot of a computer

Description automatically generated

1. **Title**: "Search for Authors and Their Papers" - This is a heading indicating that the search utility is intended to help users find authors and their associated academic papers.
2. **Search Fields**:
   * "Enter Author ID": This input field is labeled for the user to enter an author's identification number. It is common in academic databases for authors to have unique IDs that can be used to fetch their publications.
   * "OR Enter Author Name": This input field provides an alternative to searching by ID, allowing users to enter the name of an author instead. The use of "OR" suggests that the user should use either one of the fields but not simultaneously to perform a search.
3. **Search Button**: Labeled "Search", this button is presumably used to submit the search query to the system, which will then return results based on the author ID or author name provided.

**Search with paper ID**

**A screenshot of a computer

Description automatically generated**

1. **Title**: "Search for Authors and Their Papers" - This indicates that the primary function of this interface is to search for academic papers and their corresponding authors.
2. **Search Field**: There is an input field labeled "Enter Paper ID:" where users can input the identification number of a paper to search for.
3. **Search Button**: Next to the search field, there's a button labeled "Search", which users can click on after entering a paper ID to perform the search.
4. **Search Results**: Below the search field, the results of a search query are displayed:
   * "Paper: 12345" - This indicates the paper ID that was searched for, which in this case is "12345".
   * "Related Authors:" - Following this heading is a list of author IDs that appear to be linked to the paper. Each author is listed with an "Author\_" prefix followed by a unique number, such as:
     + Author\_1305
     + Author\_8076
     + Author\_8227
     + Author\_11104
   * These are likely identifiers within the database that uniquely identify authors who have contributed to or are associated with the paper.
5. **Venue Information**: The results also include a section labeled "Venue:" followed by a venue identifier "Venue\_191". This suggests that the paper was published or presented at this particular venue, and the identifier is unique to that venue within the database.
6. **Visualization Button**: At the bottom, there is a button labeled "Visualize", which implies that the user has the option to visualize the relationships or data associated with the search results. This could potentially display a graph or other visual representation of the paper, the authors, and the venue

## 11. Conclusion

This study embarked on the ambitious task of analyzing and searching a heterogeneous publication network, with a focus on the ACM dataset. Through the development and implementation of various functionalities, including data preprocessing, search applications in Flask, and advanced visualization interfaces, this project has demonstrated the potential of leveraging network analysis in the realm of academic research. The insights gleaned from this work contribute to a deeper understanding of collaboration patterns and knowledge dissemination within the computing community.

The journey of this project has been an exploration into the intricate world of academic networks, specifically through the lens of the ACM publication dataset. This study has underscored the complexity and richness of academic collaborations and how they can be effectively analyzed and visualized using advanced computational techniques and tools.

Throughout this research, we successfully developed and implemented a series of functionalities that serve as a testament to the power of combining network analysis with web technologies. Key achievements include:

1. **Efficient Data Preprocessing and Management**: The ability to process and manage large datasets like the ACM publication data efficiently has been a cornerstone of this project. Through meticulous data cleaning and organization, we ensured the integrity and reliability of the data used in our analysis.
2. **Advanced Network Analysis**: The application of NetworkX for graph analysis allowed us to delve deep into the network's structure, revealing patterns and connections that are not immediately visible. This analysis provided valuable insights into the dynamics of academic collaborations and knowledge flow.
3. **Interactive Web Application**: The development of a Flask-based web application marked a significant stride in making our analysis accessible. This platform not only democratizes data access but also provides a user-friendly interface for exploring complex network data.
4. **Dynamic Data Visualization**: Perhaps one of the most visually impactful achievements has been the implementation of interactive visualizations using pyvis. These visualizations bring the data to life, allowing users to intuitively understand and engage with the network's complexities.
5. **Practical Application and User Engagement**: The combination of a robust backend with an engaging frontend has created a tool that not only serves academic purposes but also offers practical applications for researchers, educators, and students alike.

The implications of this study extend beyond the immediate project. It highlights the potential of interdisciplinary approaches in tackling complex data analysis problems. By integrating concepts from network theory, data science, and web development, this thesis demonstrates the power of hybrid methodologies in extracting meaningful insights from complex datasets.

**Reflecting on Challenges and Learnings**

This journey was not without its challenges. Data management of large-scale networks, ensuring the scalability of the web application, and developing intuitive yet informative visualizations were among the hurdles faced and overcome. These challenges, however, paved the way for significant learnings, particularly in the realms of data processing, web application optimization, and user interface design.

**Looking Forward**

The completion of this project is not the end but a beginning. The methodologies and tools developed here open multiple avenues for future exploration and innovation. The versatility of the project's architecture allows for easy adaptation and expansion to incorporate more datasets, apply more advanced analytical techniques, and enhance user experience.

In conclusion, this project stands as a testament to the power of computational tools and techniques in understanding and visualizing complex networks. It also underscores the ever-increasing importance of interdisciplinary approaches in research, particularly in the field of data and network analysis.

## 12. Future Work

Looking ahead, there are several avenues for expanding and enhancing this research:

1. **Incorporating Additional Datasets**: Expanding the analysis to include datasets from other academic fields or integrating more extensive bibliographic data could provide a more holistic view of academic collaborations.
2. **Enhancing Algorithmic Approaches**: Implementing more sophisticated community detection algorithms or machine learning techniques could yield more nuanced insights into the network structure.
3. **Improving User Interface**: Further development of the user interface to include more intuitive navigation and interactive elements would enhance the user experience.
4. **Real-Time Data Analysis**: Integrating the capability to analyze data in real-time could provide more current insights into the evolving nature of academic networks.
5. **Cross-Disciplinary Applications**: Exploring applications of this research in other domains, such as industry collaborations or social networks, could broaden its impact.

## 13. References

* Shi, Chuan, et al. "A survey of heterogeneous information network analysis." IEEE Transactions on Knowledge and Data Engineering 29.1 (2016): 17-37.
* Sh, C., Li, Y., Zhang, J., Sun, Y., & Philip, S. Y. (2016). A survey of heterogeneous information network analysis. IEEE Transactions on Knowledge and Data Engineering, 29(1), 17-37.
* Shi, Chuan, Yitong Li, Jiawei Zhang, Yizhou Sun, and S. Yu Philip. "A survey of heterogeneous information network analysis." IEEE Transactions on Knowledge and Data Engineering 29, no. 1 (2016): 17-37.
* Shi, C., Li, Y., Zhang, J., Sun, Y. and Philip, S.Y., 2016. A survey of heterogeneous information network analysis. IEEE Transactions on Knowledge and Data Engineering, 29(1), pp.17-37.
* Shi C, Li Y, Zhang J, Sun Y, Philip SY. A survey of heterogeneous information network analysis. IEEE Transactions on Knowledge and Data Engineering. 2016 Aug 8;29(1):17-37.
* Master’s Project by Apoorva Tangirala under the super vision of Dr.Chunjiang Zhu