# Dynamic Force-directed Graph with Weighted Nodes for Scholar Network Visualization

Khalid Al-Walid Mohd. Aris[1], Chitra Ramasamy[2], Teh Noranis Mohd. Aris[3], Maslina Zolkepli[4]
Faculty of Computer Science and Information Technology, Universiti Putra Malaysia,
Serdang, Malaysia

***Abstract*—Through the growth of portals and venues to publish academic publications, the number of academic publications is growing exponentially in recent years. An effective exploration and fast navigation in the collection of academic publications become an urgent need to help academic researchers find publications related to their research and the surrounding community. A scholar network visualization approach is proposed to help users to explore a large number of academic publications concerning the strength of the relationship between each publication. The approach is realized by creating a web-based interface using D3 JavaScript algorithm that allows the visualization to focus on how data are connected to each other more accurately than the conventional lines of data seen in traditional data representation. The proposed approach visualizes data by incorporating a force-directed graph with weighted nodes and vertices to give more descriptive information of millions of raw data such as author names, publication title, publication year, publication venue and number of citations from the scholar network dataset. By introducing a weighted relationship in the network visualization, the proposed approach can give a more insightful detail of each publication such as a highly cited publication by looking at and exploring the generated interactive graph. The proposal is targeted to be incorporated into a larger-scale scholar network analytical dashboard that can offer various visualization approaches under one flagship application.**

***Keywords*—Force-directed graph; weighted network; citation network; D3 algorithm**

## I. Introduction

The existence of numerous academic social networking websites such as Google Scholar and ScienceDirect has accommodated scholars to publish their scientific publications to the public effortlessly. The purpose of this platform is to acknowledge people about a specific topic in certain disciplines.

A concern regarding academic social networking websites is how to handle the flood of information offered by the websites. People can no longer rely on traditional ways to deal with the outgrowth of scientific publications. Traditional searching and browsing functions at academic social networking websites have become outdated as more time is needed to browse through each publication to see their relevancy.

The overload of information could lead to a lack of efficiency and a lengthy period of time spent searching for valuable information. It will lead to failure in receiving a full, in-depth overview of the desired topics and domains. Therefore, a mechanism to efficiently handle the flood of information needs to be introduced, as it would speed up the process of searching and understanding the scientific network and communities in one specific discipline. An efficient search and analysis of academic networks can also help non-specialists from other disciplines quickly find existing networks that they are interested in.

In general, on academic social networking websites, the browsing function has a fixed classification algorithm that cannot provide a user with desired topics and domains. Some non-specialist users are not able to understand the jargon used in domains unfamiliar to them. Fortunately, humans are intensely visual creatures. Normally, people can read a pattern of the growth of diseases by looking at a chart, and even children can describe a bar chart and extract information from it. For that reason, an efficient scholar network visualisation approach can be an alternative way for users to replace traditional searching and browsing functionalities on academic social networking websites.

The scholar network visualization approach is considered part of data analytics and visualization, which has become a highly active field of research in recent times due to the information overload all around us. Data visualization, which deals with brain psycho-visual vision and cognitive capacities, is a privileged tool to analyze one’s environment. Network visualization research can be defined by the techniques that allow humans to visualize data through a network graph that presents a network of connected entities and nodes visually.

Many visualization tools have been introduced in recent years [1,2]. They offer many useful functions, such as data processing and visual analytics. Therefore, it has simplified the process of data visualization for users, whether they have any programming knowledge or not. The tools give users the capability to transform the data into interactive charts that are more understandable and readable by everyone. Data visualization is commonly utilized in business intelligence, scientific visualization, and analytical analysis. There are two types of visualization tools: visualization tools with programming languages and visualization tools without programming languages.

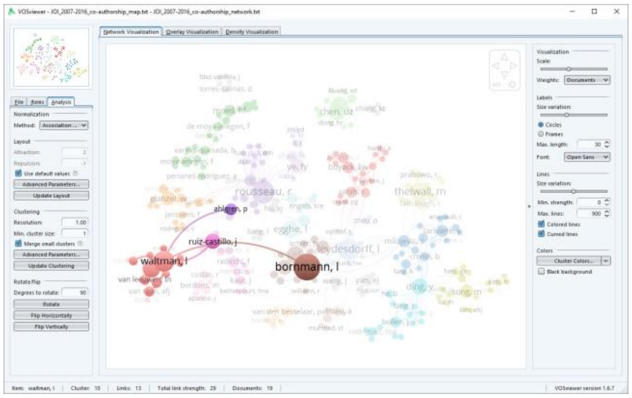

Fig. 1. Network visualization generated in VOSviewer

Tableau [3] is a software that is used mainly in business intelligence and analytics. It supports various file formats, such as txt, xlsx, csv and json. Data can also be imported from online servers such as MySQL and Oracle. Tableau can generate a suitable graph automatically by extracting the header of each variable in our dataset. Users can also use the drag-and-drop feature to add rows and columns and select a chart type. A web-based application called Infogram [4] can complete data visualization quickly; first-time users just need to register, and they can upload their own data files in various formats, such as xlsx or csv to the website. Users can also import data from Google Drive, Dropbox, OneDrive, or a JSON feed. One of the disadvantages of Infogram is data privacy. An open-source JavaScript library, D3.js [5] combines HTML and CSS methods. On D3.js official website, it provides plenty of examples with the source code to inspire users to create their own data visualization. All the graphs generated will be in svg format.

R programming also provides a package called Ggplot2 [6], which is an open-source package to visualize by generating charts. Compared to basic R graphs, the Ggplot2 package allows the user to edit the plotting component of the graph. Ggplot2 also has its own repository on Github that provides the user with an annual case study competition to show their skills. Users have a chance to use the package, and in return, they can contribute codes back to ggplot2.

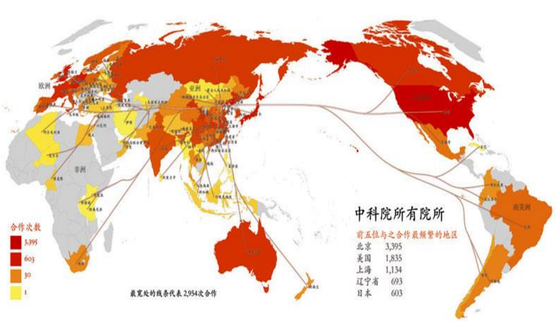

Fig. 2. Chinese Academy of Science Co-author network generated in Sci2

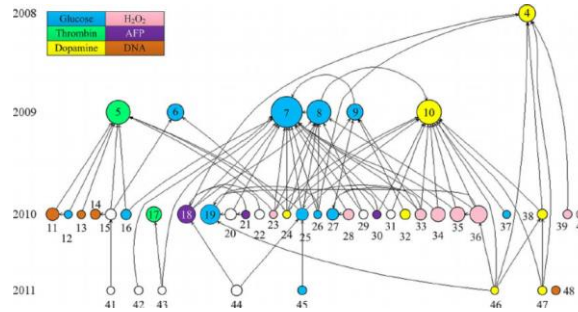


Fig. 3. Mapping papers with highest local citation score generated in HistCite

Another free visualization and analysis tool called VOSviewer [7] is able to construct and visualize bibliometric networks. It visualizes scholarly data into bibliometric networks by clustering solutions. Users can visualize their data by importing the data files from Web of Science (WOS), Pajek, and Graph Modelling Language (GML). The networks can be saved as a bitmap file or in vector format. Fig. 1 shows network visualization generated in VOSviewer. Sci2 [8] that represents The Science of Science is an open-source tool that supports temporal, geospatial, topical, and network studies. It also generates different kinds of networks. The network that generates from small datasets can be explored interactively and the network from large datasets can be rendered in Postscript files that users can convert. Fig. 2 shows a co-authorship network from the Chinese Academy of Science generated in Sci2.

HistCite [9] is used to visualize scholarly data and bibliometric analysis, including the productive authors, the scale of journals, the frequency of words, the types of documents, and the ranking of institutions. A bibliography's dataset will be converted into time-based networks called historiographs by HistCite. The historiograph assists the user in understanding the subject's main publishing events as well as the impact of the chronology on networks. Fig. 3 shows the mapping of 45 papers with the highest local citation score generated in HistCite.

BibExcel [10] is used to do multiple types of bibliometric analysis, such as citation analysis, cluster analysis, and co-citation analysis. The system allows users to select a catalogue from their data and add it as a variable in the data matrix of output files. Users can also export the files that include the data matrix and import them into other visualization tools such as Gephi, Pajek, and VOSviewer to continue their analysis. Fig. 4 shows the mapping science using BibExcel and Pajek.

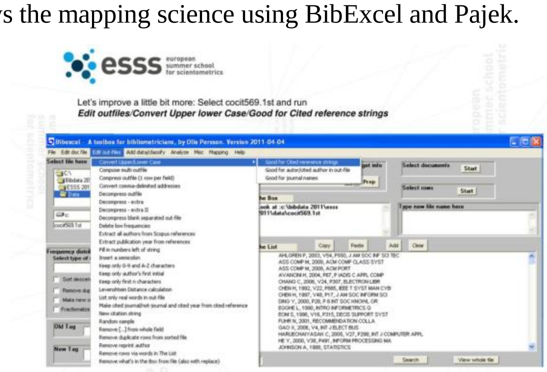


Fig. 4. Mapping science using BibExcel and Pajek

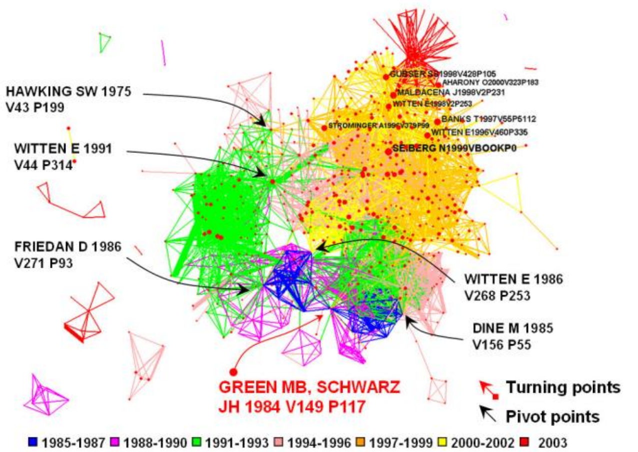


Fig. 5. Visualizing patterns and trends in scientific literature using CiteSpace

A Java application called CiteSpace [11] facilitates the user by detecting, visualizing, and analyzing increasing trends and critical changes in scientific literature. It combines information visualization methods and bibliometrics with the algorithm of data mining to read the pattern in citation data. Fig. 5 shows the visualizing patterns and trends visualised in the scientific literature using CiteSpace. A fuzzy-based clustering visualization approach, Bibliographic Big Data Visualization [12] offers a hybrid fuzzy clustering-based visualization by applying the Fruchterman-Reingold algorithm. The visualization can divide the nodes into soft clusters, but they lack the strength of the connection between the nodes. Fig. 6 shows the fuzzy clustering in Bibliographic Big Data Visualization.

By implementing query optimization and the spectral centrality measure [13], an improved scholar data visualisation was proposed, in which the scholar data is visualised in a network diagram using the centrality measure for better and faster decision making. By using the concept of a word cloud, the visualization offers a weighted network visualization. Fig. 7 shows the enhanced bibliographic data retrieval using query optimization and the spectral centrality measure. Table I compares existing scholar network visualisation approaches, including their functions and limitations.

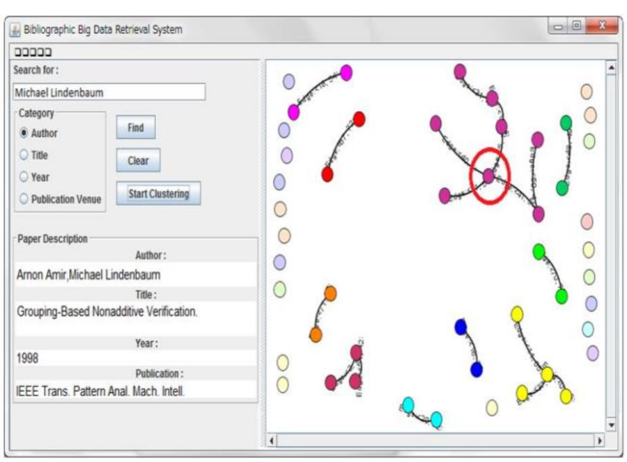


Fig. 6. Fuzzy Clustering in Bibliographic Big Data Visualization System

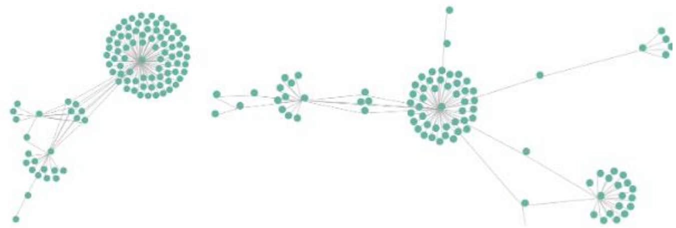

Fig. 7. Enhanced Bibliographic Data Retrieval Using Query Optimization and Spectral Centrality Measure

TABLE I. COMPARISON OF EXISTING SCHOLAR NETWORK VISUALIZATION APPROACHES

| Name | Main function | Data format | Platform | Limitation |
|---|---|---|---|---|
| Tableau[3] | General data analysis | .txt .csv .xlsx | Windows Mac OS | Not programmable to improve to algorithm |
| Infogram[4] | General data analysis | JSON | Windows Mac OS | Not programmable to improve to algorithm |
| D3[5] | General data analysis | JSON | Windows Mac OS | Limited data size |
| Ggplot2[6] | Chart visualization | .csv | Windows Mac OS | Slow to create graphics |
| VOS Viewer[7] | Citation analysis | WOS Pajek,GML | Windows Mac OS | Only support node network diagram and heat map |
| Sci2[8] | Network analysis | .txt .csv | Windows Mac OS X Linux | High memory footprint when process large datasets. |
| HistCite[9] | Static analysis | WOS | Windows but only on IE | Only support data from WOS. |
| Bib Excel[10] | Process scholar database | WOS Med-line | Wind ows | Not easy to use without its help document |
| Cite Space[11] | Co-citation analysis | WOS Pub-Med & arXiv | Windows | Cannot delete irrelevant node |

| | | | | |
|---|---|---|---|---|
| Bibliographic Big Data Visualization [12] | Fuzzy Citation Network analysis | AMiner | Java | Takes more than 2 minutes to produce visualization result due to clustering process. |
| Bibliographic Data Retrieval Using Spectral Centrality Measure[13] | Hybrid Clustering Citation Network analysis | AMiner | Python MongoDB | Only supports JSON/XML format dataset |

Most recently, NetV.js [14] high-efficiency visualization approach was introduced for large-scale graphs. It is an open-source JavaScript library that supports the fast visualization of large-scale graph data at an interactive frame rate with a commodity computer. It consists of the Graph Model Manager, the Rendering Engine, and the Interaction Manager. While D3.js library can support up to 20,000 nodes and 400,000 edges, NetV.js can support up to 50 thousand nodes and 1 million edges. For the scholar network dataset used in this study, D3.js is sufficient to produce the visualization as the dataset only has 800 nodes, but to produce large scale graphs, NetV.js is more suitable to be used as the visualization approach.

Another recent approach for visualizing large real-world (social) network data on a high-resolution tiled display system was introduced on a tiled display system consisting of multiple screens [15]. The high resolution tiled display approach used GPUs to ensure an interactive setting with real-time visualization. GPUs are gaining popularity for large-scale datasets because they can process visualization much faster.

Section II describes the scholar network dataset from AMiner and the Fruchterman-Reingold force-directed graph applied in this study. The application of Fruchterman-Reingold to the scholar network dataset and color scheme for graph nodes and vertices visualization is presented in Section III. Section IV discusses the scholar network visualization produced from the analytics, and the research conducted in this study is summarized in Section V.

## II. Materials and Method

### A. *Scholar Network Dataset from AMiner*

This section describes the dataset used in the proposed approach. The dataset is acquired from the AMiner website [16-21]. AMiner is a free online web service used to index, search, and mine big scientific data. Data acquired from AMiner is suitable for data analytics operations on academic publication information to identify connections between researchers, conferences, and publications. Some of the insights that can be produced are expert findings, geographic search, trend analysis, reviewer and examiner recommendation, association search, course search, academic performance evaluation, and research domain modeling.

The Scholar network dataset from AMiner consists of eight attributes. Table II shows the data schema of the Scholar network dataset. The citation data is extracted from DBLP, ACM, MAG, and other sources. The dataset attributes include publication id, publication title, publication authors, publication venue, published year, citation number, citing publications' id, and abstract.

### B. *Fruchterman-Reingold Algorithm for Vertices and Edges Visualization*

Force-directed graph [22-25] is used to visualize the scholar network as it provides the ability to convey the relationship between data, the weightage of the relationship, and the flow often brings out the untold insights into the limelight.

The advantages of a force-directed graph include its flexibility to adapt to increasing criteria, its intuitiveness to make a graph easy to be predicted and understood, and its simplicity in terms of fast implementation using minimal lines of code. The interactivity a force-directed graph can offer is also a big advantage as users prefer to interact with the interface for a deeper understanding of the visualization. Lastly, the force-directed graph has a strong theoretical foundation due to its usefulness in multiple fields such as physics and statistics.

In the proposed study, the Fruchterman-Reingold [26] algorithm is selected to become the visualization approach for the scholar network. The Fruchterman-Reingold algorithm offers a dynamic force-directed graph suitable for edge crossing reduction and planar graph drawing. The algorithm introduces two principles, which are the vertices connected by an edge should be drawn near each other and the vertices should not be drawn too close to each other.

TABLE II. Data Schema of the Scholar Network Dataset

| Field Name | Description | Example |
|---|---|---|
| Id | Publication ID | 013ea675-bb58-42f8-a423-f5534546b2b1 |
| Title | Publication title | Prediction of consensus binding mode geometries for related chemical series of positive allosteric modulators of adenosine and muscarinic acetylcholine receptors |
| Authors | Publication authors | ["Leon A. Sakkal", "Kyle Z. Rajkowski", "Roger S. Armen"] |
| Venue | Publication venue | Journal of Computational Chemistry |
| Year | Published year | 2017 |
| Citation | Citation number | 0 |
| Reference | Citing publications' id | ["4f4f200c-0764-4fef-9718-b8bccf303dba", "aa699fbffabe-40e4-bd68-46eaf333f7b1"] |

Suppose $f_a$ and $f_r$ are the attractive and repulsive forces respectively, with $d$ as the distance between the two vertices and $k$ as the radius of the empty area around a vertex, then

$$f_a(d) = d^2 k \quad (1)$$

$$f_r(d) = -k^2 \quad (2)$$

Given a graph $G = (V, E)$, the combined force applied on vertex $v$ is:

$$(v) = \sum_{(u,v)\in E} f_{a,uv} + \sum_{(u,v)\in V*V} f_{r,\ uv} \quad (3)$$

Fig. 8 shows the general flow of the Fruchterman-Reingold algorithm. In Fruchterman-Reingold, each node applies a repellant force on other nodes that are inversely proportional to the distance between those nodes, and each arc applies an attractive force on its endpoints proportional to the square of the distance between those nodes. Therefore, as linked nodes grow more distant from one another, the attractive force activates quickly and the repellant force drops off, so linked nodes will have the tendency to get back closer to one another. Similarly, as the nodes get increasingly close, the repellant force activates rapidly while the attractive force ceases, and the nodes will be pushed away from each other. Only when the nodes are at a well-adjusted distance from one another, the forces begin to balance; therefore the nodes will slowly stop moving. To keep track of the forces on each node, a $\Delta x$ and $\Delta y$ value for each node is maintained, where they store the gain forces on that node along the $x$ and $y$ axes. The algorithm is constantly tracking the location of each node since it's possible that a node might be repelled entirely vertically, in which case it will have a strong force in the $y$ direction but no force in the $x$ direction, or horizontally, where strong force in the $x$ direction, no force in the $y$ direction. The gain forces in each direction beginning at zero but will be adjusted by the interactions of each node with each other node. Fig. 9 depicts the pseudocode of the Fruchterman-Reingold algorithm.

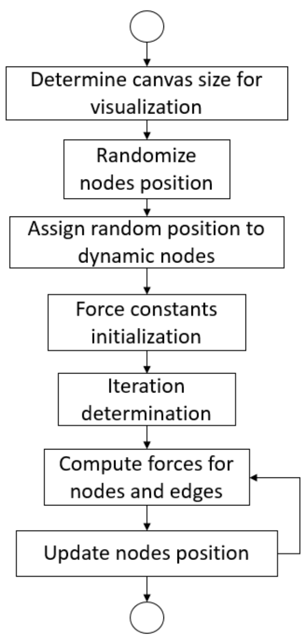


Fig. 8. Flow diagram of the Fruchterman-Reingold algorithm

```
area:= W * L; {W and L are the width and length of the frame}
G := (V, E); {the vertices are assigned random initial positions}
k := √(area/|V|);
function f_a(x) := begin return x²/k end;
function f_r(x) := begin return k²/x end;
for i := 1 to iterations do begin
    {calculate repulsive forces}
    for v in V do begin
        {each vertex has two vectors: .pos and .disp
        v.disp := 0;
        for u in V do
            if (u ≠ v) then begin
            {δ is the difference vector between the positions of the two vertices}
                δ := v.pos − u.pos;
                v.disp := v.disp + (δ/|δ|) * f_r(|δ|)
            end
    end
    {calculate attractive forces}
    for e in E do begin
        {each edges is an ordered pair of vertices .vand.u}
        δ := e.v.pos − e.u.pos;
        e.v.disp := e.v.disp − (δ/|δ|) * f_a(|δ|);
        e.u.disp := e.u.disp + (δ/|δ|) * f_a(|δ|)
    end
    {limit max displacement to temperature t and prevent from displacement
outside frame}
    for v in V do begin
        v.pos := v.pos + (v.disp/|v.disp|) * min(v.disp, t);
        v.pos.x := min(W/2, max(−W/2, v.pos.x));
        v.pos.y := min(L/2, max(−L/2, v.pos.y))
    end
    {reduce the temperature as the layout approaches a better configuration}
    t := cool(t)
end
```

Fig. 9. Pseudocode of the Fruchterman-Reingold algorithm

The Fruchterman-Reingold algorithm is applied in the experiment through a plugin in D3.js [5].

## III. Results

### A. Data Preprocessing and Exploration on the Scholar Network Dataset

The initial data consists of academic publications from 1936 to 2018. To ensure that the visualization process is fast and the graph produced is manageable, data earlier than 2010 is excluded from the experiment. Only data from 2010 and above will be visualized in the final visualization. The initial attributes of the dataset include publication id, publication title, authors' name, publication venue, published year, citation number, citing publication ID, and abstract. In the experiment, only 5 attributes are included: the authors' name, publication ID, title, number of citations, and year of publication. From the scholar dataset exploration, there is an increasing number of academic publications from year to year. Fig. 10 shows the histogram of publications from 2010 to early 2018.

A network graph has two key data elements, nodes/vertices, and links/edges. All nodes must have unique identifiers. In each node, it is possible to add as many custom variables as necessary. Links must have a valid node id as a source and a target, and they can be text or numbers. Fig. 11 shows a snapshot of the cleaned dataset ready to be visualized in the D3 algorithm. The intention is to develop a node-to-node relationship to emphasize the relationship between authors and their publications. Every node is connected to the target node with the same relationship. Another feature of the proposed approach is that every node will have a different color based on the year it was published to the public, and the radius of the node will correspond with the number of citations in every academic publication. If the user enters the author's name in the space provided, it will highlight the other nodes that are related to it. If users hover the mouse over one of the nodes, they can see the details for every academic publication they want. The information will be displayed on the left side of the graph.

### B. Color Scheme for Graph Nodes and Vertices Visualization

The color palette for the nodes and vertices was chosen according to the Web Content Accessibility Guidelines (WCAG) [27] which suggests the minimum contrast ratio between text or image and background is 4.5:1. Table III describes the color ratio for every color used in the force-directed graph visualization. Ten colors are chosen to represent 10 different clusters of the scholar network to be visualized in the graph. If more than 10 network clusters exist, the same color will be repeated in other clusters.

TABLE III. COLOR RATIO FOR EVERY COLOR IN THE FORCE-DIRECTED GRAPH

| Color Code | Contrast Ratio | Color |
|---|---|---|
| #ffffff | 18.37 | |
| #ffb646 | 10.52 | |
| #ff863d | 7.63 | |
| #ff8882 | 7.57 | |
| #00aa9f | 6.34 | |
| #1d9c3d | 5.14 | |
| #ff352e | 5.07 | |
| #c06c30 | 4.74 | |
| #9262f8 | 4.65 | |
| #0781df | 4.56 | |

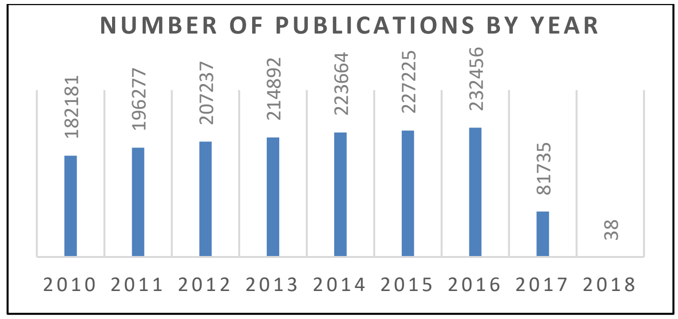


Fig. 10. Number of academic publications from 2010 to 2018 from AMiner

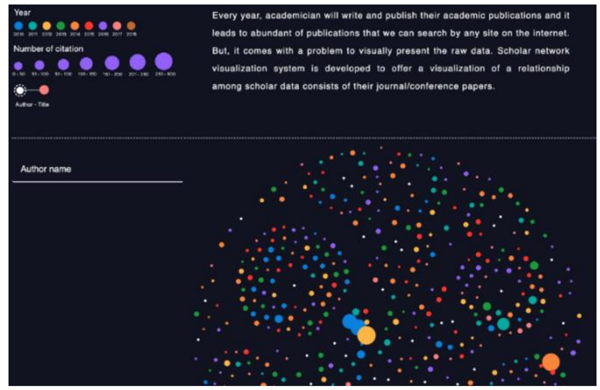


Fig. 11. The visualization of the Scholar Network Visualization approach (Initial Graph)

## IV. Discussions

This section discusses the scholar network visualization produced using the D3 library based on the experiment done on the scholar network dataset.

The Fruchterman-Reingold algorithm analyses the scholar network dataset to produce a dynamic force-directed graph visualization, and the visualization is created using the D3 algorithm [28]. D3 is a JavaScript library for manipulating documents based on data. It can bring data to life using HTML, SVG, and CSS. D3's emphasis on web standards offers the full capabilities of modern browsers without tying the data to a proprietary framework, combining powerful visualization components with a data-driven approach to DOM manipulation. It is an increasingly popular approach to data analytics visualization as it can produce sophisticated data visualization that is fast, interactive, and shareable across many platforms.

The graph produced contains nodes linked by lines that represent the relationship between the nodes. D3 implements the Fruchterman-Reingold algorithm to give the user more control over the layout. It implements three primary forces upon the nodes at each tick:

- The sum of the forces acting on each node by all other nodes
- The force pushing and pulling between two linked nodes
- The force pulling each node to a focal point, usually the center of the user-defined space.

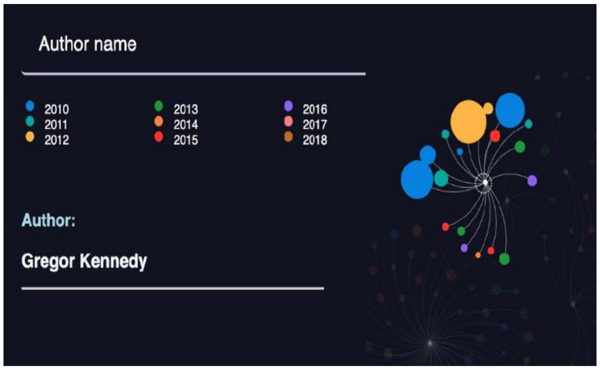


Fig. 12. The highlighted nodes after the user search for “Gregor Kennedy”

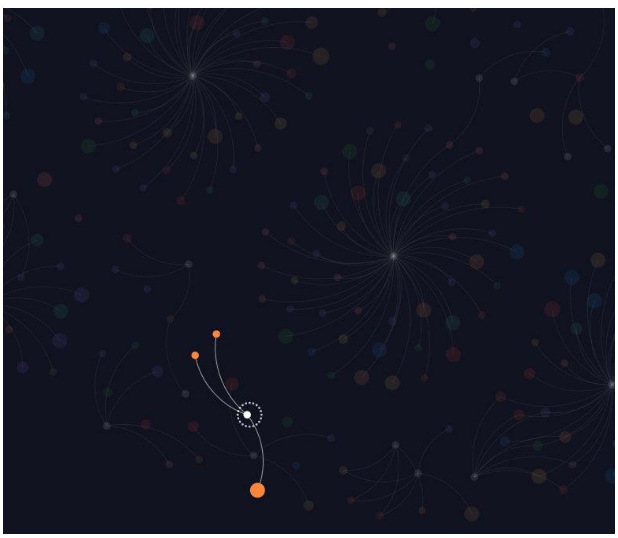

Fig. 13. Visualization of academic publications from author name “Maslina Zolkepli”


Fig. 14. Visualization of academic publications from author name “Mahdi Abavisani”

Fig. 12 shows the live interface of the D3 algorithm. Initially, when no keyword is supplied by the user, the generated graph consists of all the data that exists in the scholar network dataset. This is feasible when the dataset has between 50 and 500 nodes. If the number of nodes gets larger than 500, the graph becomes too crowded and difficult to comprehend and understand. To ensure that the graph is manageable, data cleaning and pre-processing are required to ensure only recent data is included in the dataset.

Fig. 13 shows the live interface when the keyword "Gregor Kennedy" is supplied by the user. Based on the keyword, a new graph is generated with only data that is related to "Gregor Kennedy". The nodes are weighted according to the number of citations. The larger the node, the higher the number of citations each publication has. Each node is also linked to other nodes that show the link between a publication and its citations. Fig. 14 shows the scholar network visualization result for the keyword "Maslina Zolkepli". Only vertices that contain the information related to the supplied keyword will be highlighted, and users can focus on the highlighted note. Unrelated vertices will be disabled, thus saving resources and time to produce the visualization and speeding up the process for further actions. Fig. 15 shows the scholar network visualization result for the keyword "Mahdi Abavisani". When the mouse cursor is placed on the node, the description of the publication is displayed in the left-hand corner of the page.

## V. Conclusion

The abundance of scholarly data that is available right now brings a variety of opportunities and challenges for scholarly data analysis. Users are more aware of the significance of applying visualization technologies to different datasets to comprehend the science itself. Thus, the scholar network visualization approach plays a role in addressing the problems that arise from large volumes of diverse and important data.

In this paper, a scholarly network visualization approach is proposed by incorporating weighted nodes and vertices in a dynamic force-directed graph generation. By offering weighted nodes and vertices, users can get an informative view of the visualization. The network graph is dynamic and responds to the user’s action in order to focus on several important nodes as requested by the user. The nodes can be further explored by clicking on them, and the related information will be displayed. The proposed approach is expected to increase the significance of data visualization and highlight some insights for people.

Some of the suggestions for the improvement of the scholar network visualization approach in the future are that it should be able to categorize visualization into specific fields and domains to decrease the visualization complexity. It also should be able to use various visualization techniques that can handle large-scale graphs, such as NetV.js and network visualization using a tiled display system, to deal with the ever-increasing complexity of the data. By exploring more ways to visualize data, the scope of the data can also be increased to show more relationships between the data in the best way possible.

## REFERENCES

[1] J.M. Brunetti, S. Auer, R. Garcia, J. Klimek, M. Necasky, "Formal Linked Data Visualization Model," in Proc. Intl. Conf. on Inf. Integration and Web-based Appl. & Svcs, ACM, New York, NY, USA, 2013, pp. 309-318.

[2] F. Desimoni, L. Po, "Empirical evaluation of Linked Data visualization tools," Future Generation Computer Systems, vol. 112, pp. 258-282, 2020.

[3] J. Hoelscher, A. Mortimer, "Using Tableau to visualize data and drive decision-making," Journal of Accounting EducationI, vol. 44, pp. 49-59, 2018.

[4] F. Khouzam, N. Sharaf, M. Saad, C. Sabty, S. Abdennadher, "Automatic Infogram Generation for Online Journalism," in 23rd International Conference Information Visualisation (IV) 2019, IEEE,Paris, France, 2019, pp. 56-6.

[5] A.A Khade, "Performing Customer Behavior Analysis using Big Data Analytics," Procedia Computer Science, vol. 79, pp. 986-992, 2016.

[6] H. Wickham, "ggplot2: Elegant Graphics for Data Analysis (Use R!)," 2nd ed., New York, USA: Springer-Verlag, 2016.

[7] L. Xie, Z. Chen, H. Wang, C. Zheng, J. Jiang, "Bibliometric and Visualized Analysis of Scientific Publications on Atlantoaxial Spine Surgery Based on Web of Science and VOSviewer," World Neurosurgery, vol. 137, pp. 435-442, 2020.

[8] K. Börner, "Plug-and-Play Macroscopes: Network Workbench (NWB), Science of Science Tool (Sci2), and Epidemiology Tool (Epic)," in Encyclopedia of Social Network Analysis and Mining, pp. 1280-1290, 2014.

[9] E. Garfield, "From the science of science to Scientometrics visualizing the history of science with HistCite software," Journal of Informetrics, vol. 3, no.3, pp. 173-179, 2009.

[10] O. Persson, R. Danell, J. Wiborg Schneider, "How to use Bibexcel for various types of bibliometric analysis," in Celebrating scholarly communication studies: A Festschrift for Olle Persson at his 60th Birthday, ed. F. Åström, R. Danell, B. Larsen, J. Schneider, 2009, pp. 9–24.

[11] C. Chen, F. I. Sanjuan, J. L. Hou, "The structure and dynamics of co-citation clusters: A multiple-perspective co-citation analysis," J. Assoc. Inf. Sci. Technol., vol. 61, pp. 1386-1409, 2010.

[12] M. Zolkepli, F. Dong, K. Hirota, "Visualizing Fuzzy Relationship in Bibliographic Big Da-ta using hybrid approach combining fuzzy c-means and Newman-Girvan algorithm," Journal of Advanced Computational Intelligence and Intelligent Informatics(JACIII), vol. 18, no.6, pp. 896-907, 2014.

[13] C. Ramasamy, M. Zolkepli, "Enhanced Bibliographic Data Retrieval and Visualization Using Query Optimization and Spectral Centrality Measure," Journal of Advanced Research in Dynamical and Control Systems(JARDCS) vol.11, no.3, pp. 1734-1742, 2019.

[14] D. Han, J. Pan, X. Zhao, W. Chen, "NetV.js: A web-based library for high-efficiency visualization of large-scale graphs and networks," Visual Informatics, vol. 5, no. 1, pp. 61-66, 2021.

[15] G.G. Brinkmann, K.F.D. Rietveld, F.J. Verbeek, F.W. Takes, "Real-time interactive visualization of large networks on a tiled display system," Displays, vol. 73,pp. 102164, 2022.

[16] J. Tang, J., A. C. M. Fong, B. Wang, J. Zhang, "A Unified Probabilistic Framework for Name Disambiguation in Digital Library," IEEE Transaction on Knowledge and Data Engineering (TKDE), vol. 24, no. 66, pp. 975-987, 2012.

[17] J. Tang, D. Zhang, L. Yao, "Social Network Extraction of Academic Researchers," in Proceedings of 2007 IEEE International Conference on Data Mining(ICDM'2007), pp. 292-301, 2007.

[18] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, Z. Su, "ArnetMiner: Extraction and Mining of Academic Social Networks," in Proceedings of the 14th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining (SIGKDD'2008), pp. 990-998, 2008.

[19] J. Tang, J. Zhang, R. Jin, Z. Yang, K. Cai, L. Zhang, Z. Su, "Topic Level Expertise Search over Heterogeneous Networks," Machine Learning Journal, vol. 82, no. 2, pp. 211-237, 2011.

[20] J. Tang, L. Yao, D. Zhang, J. Zhang, "A Combination Approach to Web User Profiling," ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 5, no. 1, 2010.

[21] H. Wan, Y. Zhang, J. Zhang, J. Tang, "AMiner: Search and Mining of Academic Social Networks," Data Intelligence, vol.1, no.1, pp. 58–76, 2019.

[22] J. Lu, Y.W Si, "Clustering-based force-directed algorithms for 3D graph," The Journal of Supercomputing, vol. 76, no. 6, pp. 9654–9715, 2020.

[23] S.H. Cheong, Y.W. Si, R.K. Wong, "Online force-directed algorithms for visualization of dynamic graphs," Information Sciences, vol. 556, pp. 223-255, 2021.

[24] R. Tamassia, "Handbook of Graph Drawing and Visualization," 1st ed., London, England: Chapman & Hall/CRC, 2016.

[25] D. L. Reingold "Chapter 4 - Installation, orientation, and layout," in Analyzing Social Media Networks with NodeXL, 2nd ed., Cambridge, MA, USA: Morgan Kaufmann, 2020, pp. 55-66.

[26] T.M. Fruchterman, E.M. Reingold, "Graph drawing by force-directed placement," Software: Practice and Experience, vol. 21, no. 11, pp. 1129–1164, 1991.

[27] S.H. Li, D.C. Yen, W.H. Lu, T.L. Lin, "Migrating from WCAG 1.0 to WCAG 2.0 – A comparative study based on Web Content Accessibility Guidelines in Taiwan," Computers in Human Behavior, vol. 28, no. 1, pp. 87-96, 2012.

[28] R. W. Milton, "Geospatial Computing: Architectures and Algorithms for Mapping Applications," Ph. D. dissertation, The Bartlett Centre for Advanced Spatial Analysis, University College London, London, England 2019. [Online]. Available: https://discovery.ucl.ac.uk/id/eprint/10072340/.