International Journal of Business, Human and Social Sciences
ISSN: 2517-9411
Vol:10, No:3, 2016

Applying Hybrid Graph Drawing and Clustering Methods on Stock Investment Analysis

Mouataz Zreika, Maria Estela Varua

***Abstract*—**Stock investment decisions are often made based on current events of the global economy and the analysis of historical data. Conversely, visual representation could assist investors' gain deeper understanding and better insight on stock market trends more efficiently. The trend analysis is based on long-term data collection. The study adopts a hybrid method that combines the Clustering algorithm and Force-directed algorithm to overcome the scalability problem when visualizing large data. This method exemplifies the potential relationships between each stock, as well as determining the degree of strength and connectivity, which will provide investors another understanding of the stock relationship for reference. Information derived from visualization will also help them make an informed decision. The results of the experiments show that the proposed method is able to produced visualized data aesthetically by providing clearer views for connectivity and edge weights.

***Keywords*—**Clustering, force-directed, graph drawing, stock investment analysis.

I. Introduction

STOCK investment decisions require time, knowledge, and awareness of the market, and are often made based on current events of the global economy and the analysis of historical data. The stock market contains a huge amount of data that vary over time. Moreover, the stock price of a company is influenced by various factors ranging from the performance of the company itself to the condition of the economy in general [1]. Thus, for investors to manage their portfolio well, they must analyze stock market data regularly in order to identify the potential connection between various companies, in addition to predicting the future movement of each stock based on available historical data. However, finding and analyzing useful information in such a complex data oriented market usually requires high level of analytical skills and effort from non-expert investors. This is where the current research will be of help. The proposed hybrid algorithm is designed to assist investors make better decisions based on stock market trend analyses.

To reduce the complexity of the analysis of stock market raw data, the hybrid visualization method was developed to examine the historical price movements of publicly traded stocks. The method is aimed to cluster similar stock together and provide investors the information that will enable them to predict future trends. The components of the hybrid visualization proposed are *Graph Drawing* and *Clustering*.

Mouataz Zreika is with the School of Business, Western Sydney University, Australia (corresponding author, e-mail: M.Zreika@westernsydney.edu.au).
Maria Estela Varua, is with the School of Business, Western Sydney University, Australia (e-mail: M.Varua@westernsydney.edu.au).

Visual representation is one of the most efficient ways to assist investors have a clearer overview of the movement of the stock market, as well as providing a deeper understanding of individual stocks. The application of graph drawing methods could provide visualized data with specific attributes such as weight, information which comes with graphically connections between each data element.

With the rapidly increasing size in networks, drawing a large graph with clear representations of data and its network structures is becoming a big challenge to the graph drawing community. The key issue here is not only to provide users with a comprehensive display of large graphs on the screen, but also a user-friendly navigable visual structure for users browsing through the structure to find a particular detail of the data [19]. In the past, some attempts to overcome this problem have proceeded in two main directions, namely:

- *Clustering*: Groups of related nodes are clustered into super-nodes. The user sees a summary of the graph with the super-nodes (clusters) and super-edges between the super-nodes (clusters). [2]-[4]. E.g. *K-mean* clustering method, *Markov Clustering* method.
- Navigation: The user sees only a small subset of the nodes and edges at any one time, and facilities are provided to navigate through the graph [5]-[11].

Clustering is the task of grouping a set of objects in such a way that objects in the same group (called cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters). Clustered graphs have been widely incorporated in *Graph Drawing* to overcome the problem of drawing large (or huge) graphs with thousands, or perhaps millions of nodes [18]. Cluster analysis itself is not one specific algorithm, but the general task to be solved. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them. Popular notions of clusters include groups with small distances among the cluster members, dense areas of the data space, intervals or particular statistical distributions. *Clustering* can therefore be formulated as a multi-objective optimization problem [12].

In practice, applying different *clustering* algorithms to the same clustered graphs might create very different final layouts. *Force-directed* layout algorithms use a physical analogy to draw graphs. A graph is viewed as a system of bodies with forces acting between the bodies. The algorithm seeks a configuration of the bodies with locally minimal energy, that is, a position for each body, such that the sum of forces on each body is zero. And the method is easy to understand, the results is normally good [2], [13]-[17]. However, *force-directed*

methods can deal with only a limited number of nodes due to its slow convergence time. In this study, an approach is proposed which combines *clustering* method and the traditional *force-directed* algorithm, to represent a clear overview of the whole structure on relevant stocks in reasonable convergence time by dividing a long convergence. The proposed method is applied to drawing weighted graphs. The early outcome of our approach indicates improvement in computation time and better graph aesthetics that provides a clearer view of the properties associated with the weighted graph in terms of its connectivity and edge weights. The preliminarily experimental results also show that the combination of the *clustered graph drawing* method and the *force-directed* layout algorithm could be applied in large graph drawing.

II. Methods

The proposed hybrid method that combines c*lustering* and *force-directed* algorithms was tested in experiments, in which we adopted the *clustering* method based on edge weight to group vertices for pre-handling and then applied forces within each cluster. Details are described in the following subsections.

A. Decrease Progressively Clustering on Weighted Graph (DPCW)

The *DPCW clustering* is based on the connectivity of vertices and weight on each edge in the graph. The basic idea is that if a vertex v_i is assigned in a cluster c_j, then we intend to include all its connected vertices with the most weights in the graph in this cluster. [19]

Suppose that $W = (w_0, w_2, \ldots, w_k)$ is the set of weights on every edge, w_k is the maximum weight, and w_0 is the minimum weight in W. Assume further that $G = (V, E)$ is a connected undirected weighted graph, where V is the set of vertices and E is the set of edges among V. A cluster graph $C = (G', T)$ consists of graph $G' = (V', E')$ and a rooted tree T, where G' is a sub-graph of G. The DPCW algorithm can be described as follows:

a) If $(v_m, v_n) \in V$, where $e_i = (v_m, v_n)$ and its weight $w_i = w_k$, then we add two vertices v_m and v_n into the same cluster c_k^1;
b) If $(v_{m1}, v_{n1}) \in V$, where $e_{i1} = (v_{m1}, v_{n1})$ and its weight $w_{i1} = w_k$.
1) If ($m = m_1$and $n \neq n_1$), then we add vertex v_{n1} into the cluster c_k^1;
2) If ($m \neq m_1$and $n = n_1$), then we add vertex v_{m1} into the cluster c_k^1;
3) If ($m = n_1$and $n \neq m_1$), then we add vertex v_{m1} into the cluster c_k^1;
4) If ($m \neq n_1$and $n = m_1$), then we add vertex v_{m1} into the cluster c_k^1;
5) If ($m \neq m_1$and $n \neq n_1$and$m \neq n_1$and $n \neq m_1$), then we add two vertices v_{m1} and v_{n1} into the same cluster c_k^2;
c) Repeat step (b) until every vertex satisfies the conditions described in (b) are included in clusters, and the cluster $c_k = \{c_k^1, c_k^2, \ldots, c_k^{xk}\}$;
d) Find the smaller weight $w_{k-1} \in W$, where $w_{k-1} < w_k$ and $w_{k-1} > \{w_0, w_2, \ldots, w_{k-2}\}$, set $w_k = w_{k-1}$, Repeat step (b) and (c) until every vertex satisfies the conditions described in (b) are included in clusters, and the cluster $c_{k-1} = \{c_{k-1}^1, \ldots, c_{k-1}^{x(k-1)}\}$;
e) Repeat step (d) until $w_i = w_0$ and every weight in W has been handled;
f) The final clusters $C = \{c_k^1, \ldots c_k^x, \ldots, c_{k-1}^1, \ldots c_{k-1}^{x(k-1)}, \ldots, c_0^1, \ldots c_0^{x0}\}$.

B. A Classical Force-Directed Algorithm

The *force-directed* algorithm aims to position nodes with as few crossing edges as possible by assigning forces among the set of nodes and edges for drawing graphs in an aesthetically pleasing way. The spring forces are used to keep all elements in reasonable distances in such a way that it not too close and not too far.

The *force-directed* algorithms achieve this by assigning forces amongst the set of edges and the set of nodes. The entire graph is then simulated as if it were a physical system. In the *force-directed* algorithm, we need to calculate all forces working on every element, and then place them to a suitable position to avoid edge crossings. The three steps for each iterative calculation are:

a) Calculate the effect of attractive forces $f_a(d) = d^2/k$ between adjacent vertices;
b) Calculate the effect of repulsive forces $f_r(d) = -k^2/d$ between all pairs of vertices;
c) Finally stop the iteration if f_a and f_r tend to not be changed.

where d represents the distance between two vertices while k is the optimal distance between vertices.

C. Relevant Rate/Weight Computing Algorithm

Based on the finalized layout using the combined methods of *DPCW* and *force-directed* algorithm, our solution provides investors an overview of all the stocks, as well as the graphical representation of the relationships. The steps involved are summarized as follows:

1. Trend Computing

(a) Individual Rate

Suppose the costs of an independent stock in two continuous business days are c_1 and c_2, the rate $r_1 = (c_2 - c_1) / c_2$, then the rate array of stock k is $r_k = \{r_1, r_2, \ldots, r_k\}$;

(b) Relevant Rate

Suppose the rates of two independent stocks in two same continuous business days are r_{mi} and r_{ni}, the rate $r_{mni} = (r_{ni} - c_{mi}) / c_{ni}$, then the rate comparison array of stock m and n is $r_{mn} = \{r_{mn}^1, r_{mn}^2, \ldots, r_{mn}^k\}$, all the different time periods are dropped, only those rates changes happened within the same time period on both stocks are taken into account.

2. Weight Computing

Edge thickness is applied for displaying how close the connection is between stocks based on rates computing from above, the transmission from Rate to Weight is shown as:

1) If the absolute value of the original relevant rate r_{mn} is bigger than 1, then the weight $w_{mn} = -1$, which means no connection;

2) If the absolute value of the original relevant rate r_{mn} is smaller than 1, then the weight w_{mn} = Int((1 - Abs(r_{mn})) * 10);

Based on the weight calculation results above, edge thickness is computed as 0.5 * w_{mn}.

III. Our Approach

This section describes the various steps of our approach. Graphically, the approach proposed in this study is depicted in Fig. 1.

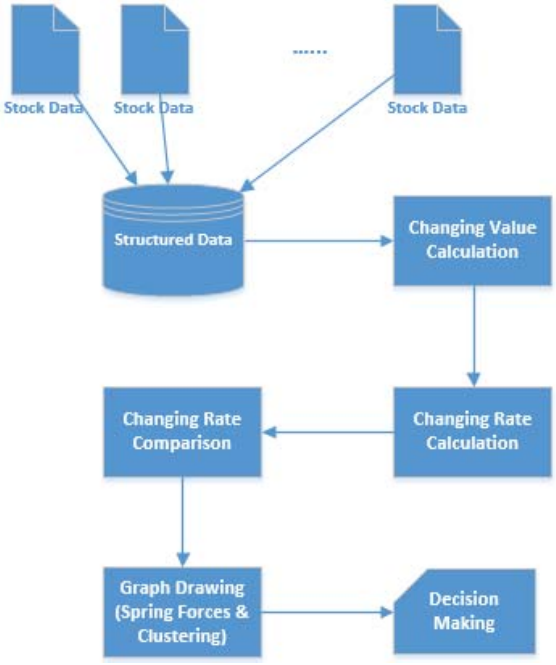


Fig. 1 Workflow of approach

Following Fig. 1, the proposed approach is summarized in the following steps:

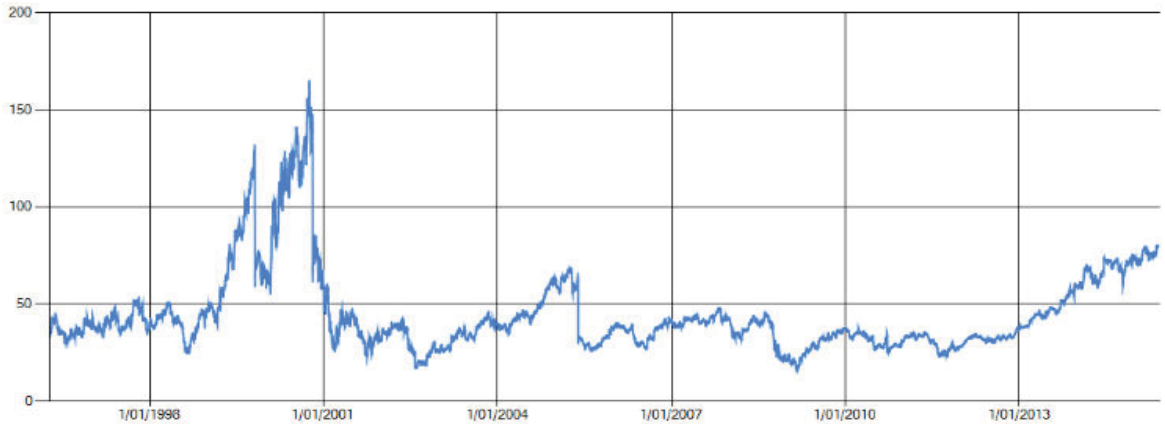
a. The first step is to collect and import the raw data of stock into the experimental platform;
b. Format and compute the changing value/rate of each stock as input for the experiments;
c. Build up weighted graph files;
d. Apply the *force-directed* algorithm on a given graph $G = (V, E)$ from step c;
e. Apply *clustering* method *DPCW* on the given graph *G;*
f. Apply the *force-directed* algorithm on clustered graph $C(G) = (G', T)$ with all its clusters 'close' as red dots in the layout until the (*force-directed* drawing) convergence process is completed and reaches the energy balance;
g. 'Open' all its clusters in the layout of *C(G);*
h. Apply the forces on the elements within the same clusters separately again to achieve the energy balance.
i. For any individual stock *x* details, apply forces on vertices connected to the specific vertex *x*, edge length needs to be adjusted based on weight on each edge, as well as the edge thickness.

IV. Case Study

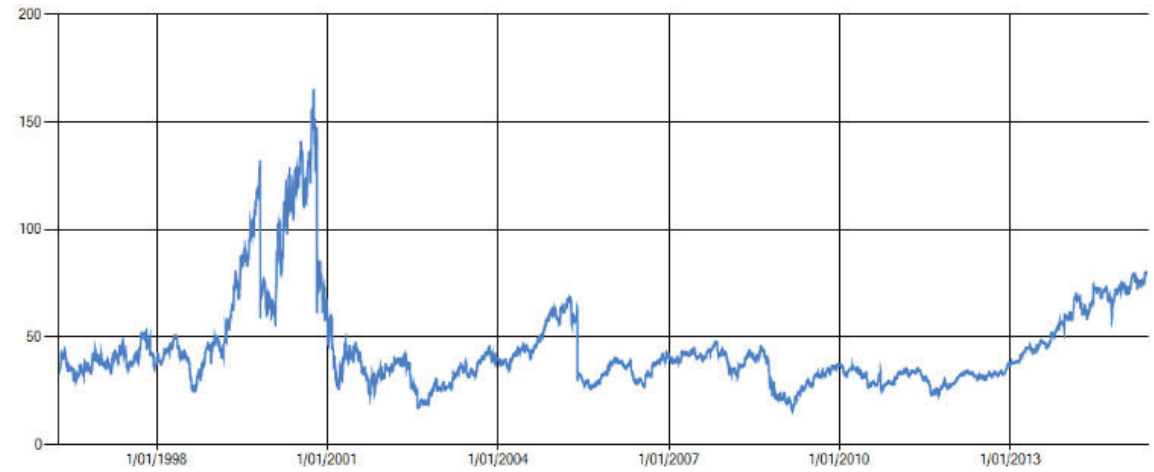
A case provided here to explain the details of our proposed methodology. For this case study, the raw data was collected from yahoo finance [20] for the period 1996 to 2015 (some stocks may be less) of 43 companies, which includes Adobe, Alibaba, Amazon, Apple, Facebook, HP, IBM, Lenovo, Microsoft, Oracle, Twitter, VNET, Weibo and Yahoo. The detailed data aspects of each stock include:

- Open Value,
- Close Value,
- High Value,
- Low Value,
- Adjusted Close Value,
- Volume and Average Value.

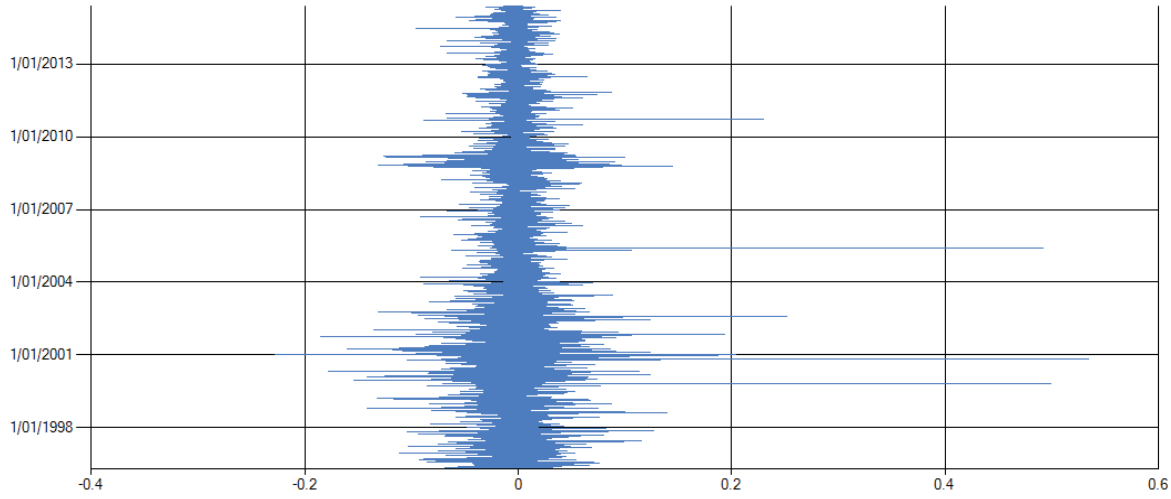
Roughly 150,000 data entries are used in our experiments. The three kinds of charts (Spline, Bar and Column) are created for each individual stock as illustrated in Fig. 2.



(a) Spline



(b) Column



(c) Bar

Fig. 2 Example of raw data changing trend on Open Value of Adobe

Fig. 2 provides investors with visualized representation of patterns and trends of individual stocks. This will aid investors make informed decisions regarding their portfolio.

Apple vs IBM		Apple vs Oracle	
OPEN:	1.040174	OPEN:	0.6769175
CLOSE:	1.009599	CLOSE:	0.8459578
HIGH:	0.7796444	HIGH:	0.5193473
LOW:	0.9062253	LOW:	0.6908079
ADJCLOSE:	0.6904746	ADJCLOSE:	0.7512918
VOLUME:	-6.701494	VOLUME:	1.361306
AVG:	0.8852233	AVG:	0.6968644

Fig. 3 Changing rate comparisons of Apple, IBM and Oracle

Based on the long-term stock value changing analytics, potential relationships between different stocks (companies) are provided depending on the final layouts of graphs. Bigger changing rate leads to weaker connection, as shown in Fig. 3, most changing rates such as open value, close value, high value, low value and average value between Apple and Oracle are lower than the relevant changing rates between Apple and IBM, which means that Apple may tends to have a closer connection to Oracle than IBM, since the differences of the value movements are bigger between Apple and IBM.

Given the results (Figs. 2 and 3) of the proposed method for each stock, investors may use the result to find stocks behaving in the same manner or may be affected by the same economic circumstances. Information such as these is important as it may help investors adjust their stock investments in order to reduce risk. In summary, benefits of applying the proposed method may include:

- Timely adjustment of investments portfolio to reduce risk;
- Reduce possible losses;
- Make better informed decisions.

A much detailed discussion of the results is provided in the next section.

V. Experimental Evaluation

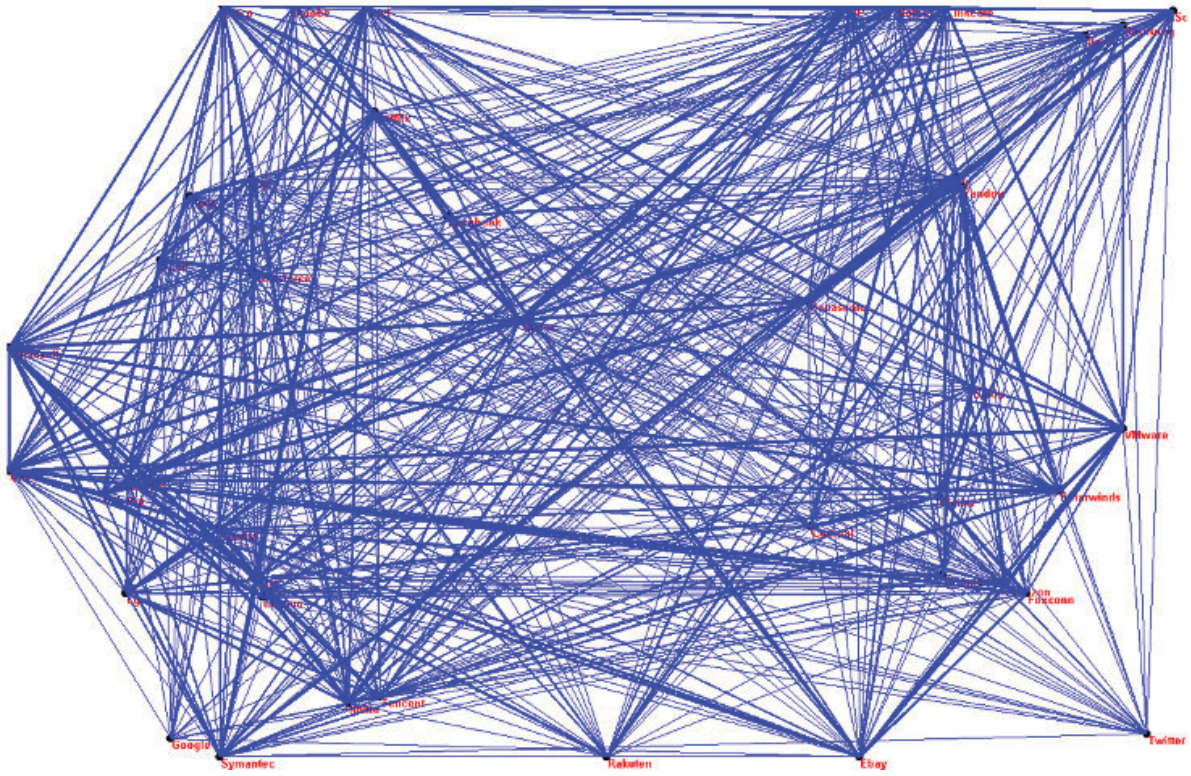
In our experiment, we created artificially seven 7 connected / undirected weighted graphs based on around 150,000 data

entries downloaded from yahoo finance [20] to test the proposed method. The data was used for evaluation and seven 7 types of graphs representing seven 7 different types of data information such as Open Value, Average Value etc. We then applied the *DPCW* and forces on each graph and then compared the edge length and thickness of each final layout. See Fig. 4 for the finalized layouts and Fig. 5 for the comparison result example.

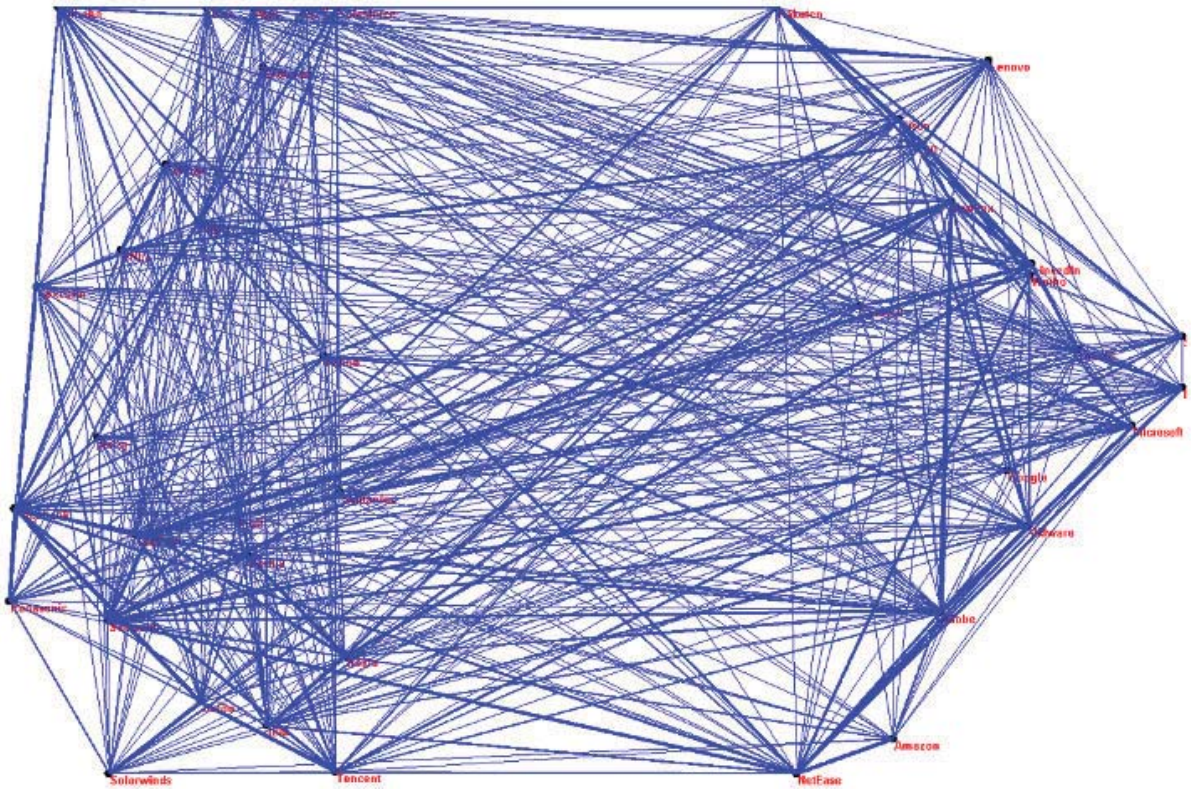
From the results of Fig. 4 (g), clusters have been identified as:

Cluster 1: {Acer, Asus, HTC, Salesforce};
Cluster 2: {Microsoft, Yahoo, Alibaba, HP, NetEase, LG, Adobe, Apple, Quanta, Tencent, VMware};
Cluster 3: {Cisco, Nokia};
Cluster 4: {IBM, Intel, Oracle, Symantec};
Cluster 5: {JD, Weibo, Rakuten, VNET};
Cluster 6: {Panasonic, Sony, Yandex};
Cluster 7: {Amazon, EBay, Samsung, Foxconn, Compal};
Cluster 8: {Facebook, Netflix, Baidu, Lenovo, Solarwinds, Benq};
Cluster 9: {Google, LinkedIn};
Cluster 10: {Groupon, Twitter}.

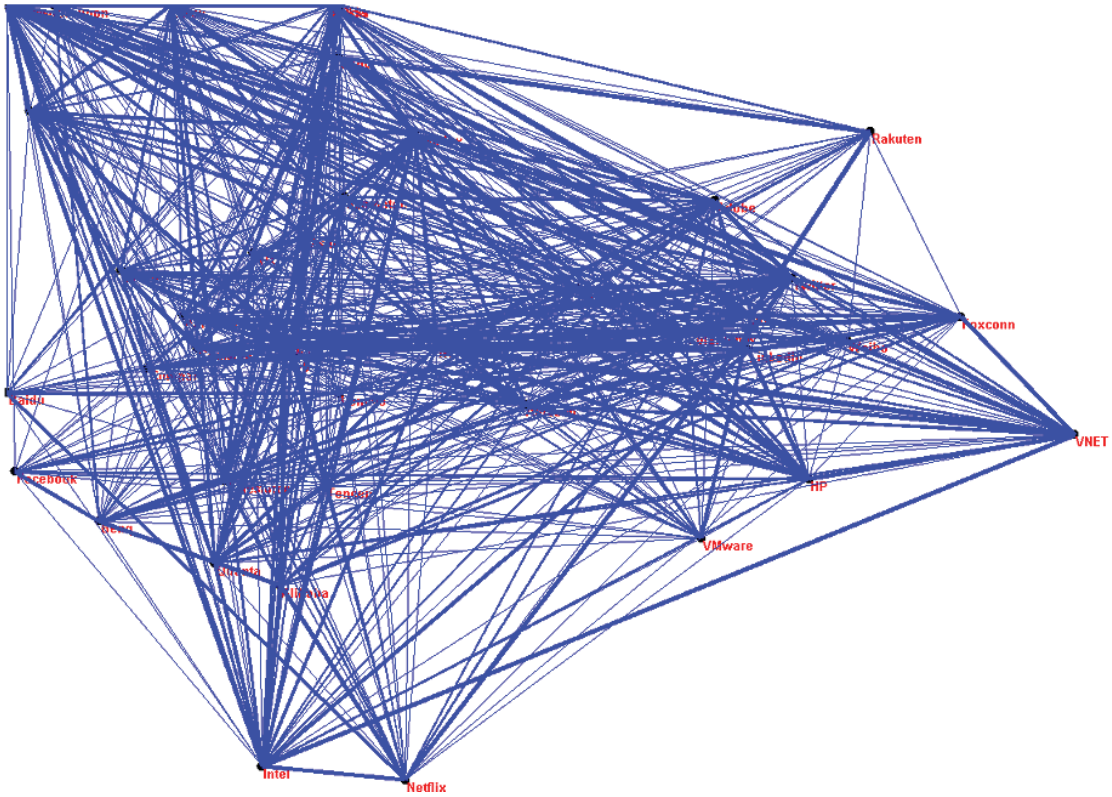
Based on the thickness of edges, we can see from Fig. 5 that Microsoft has closer connections to Oracle, Yahoo and IBM and weaker relationships to HP, Apple etc. The above results could be used in the future to make decisions on Microsoft stock related investments.



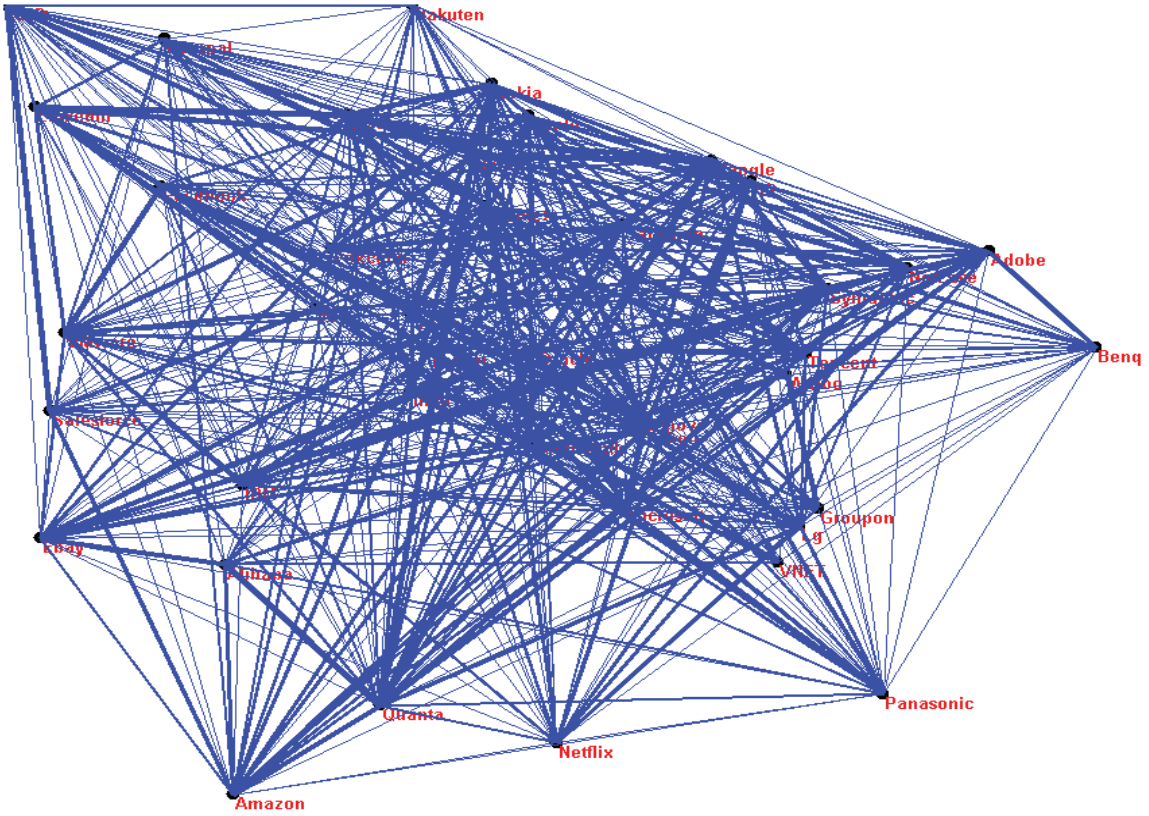
(a) Open Value



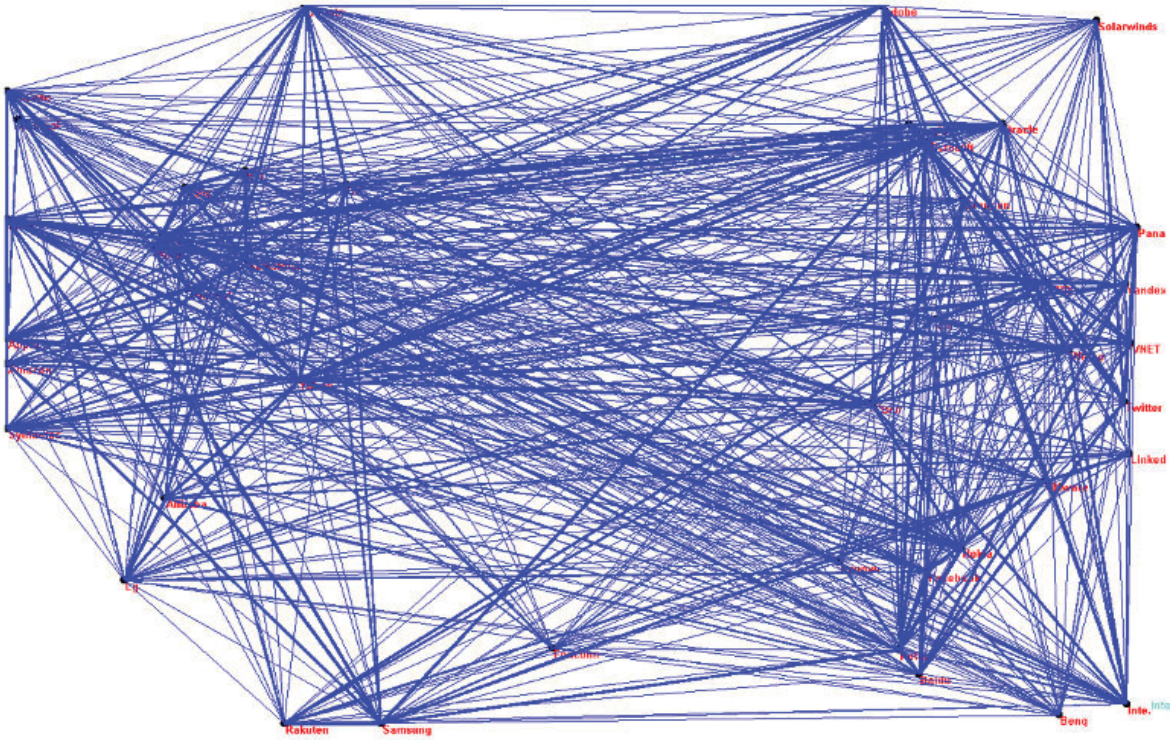
(b) Close Value



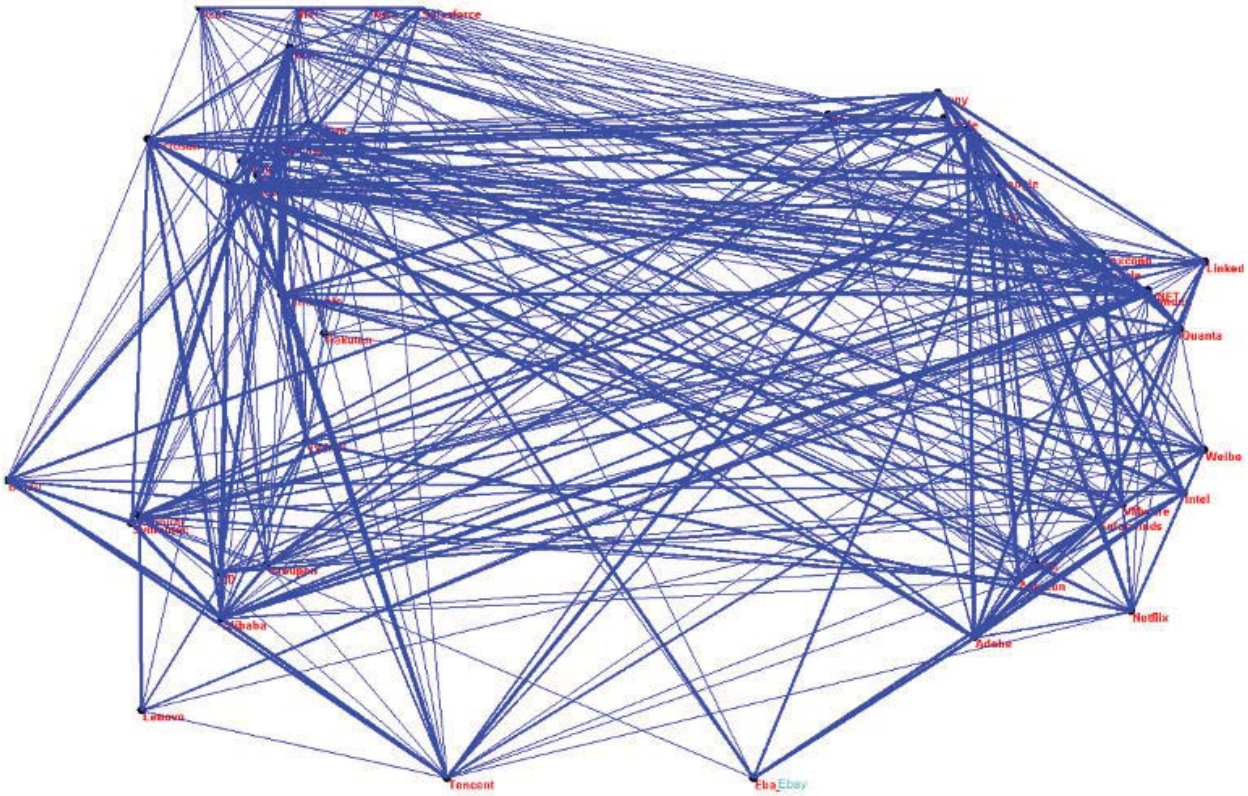
(c) High Value



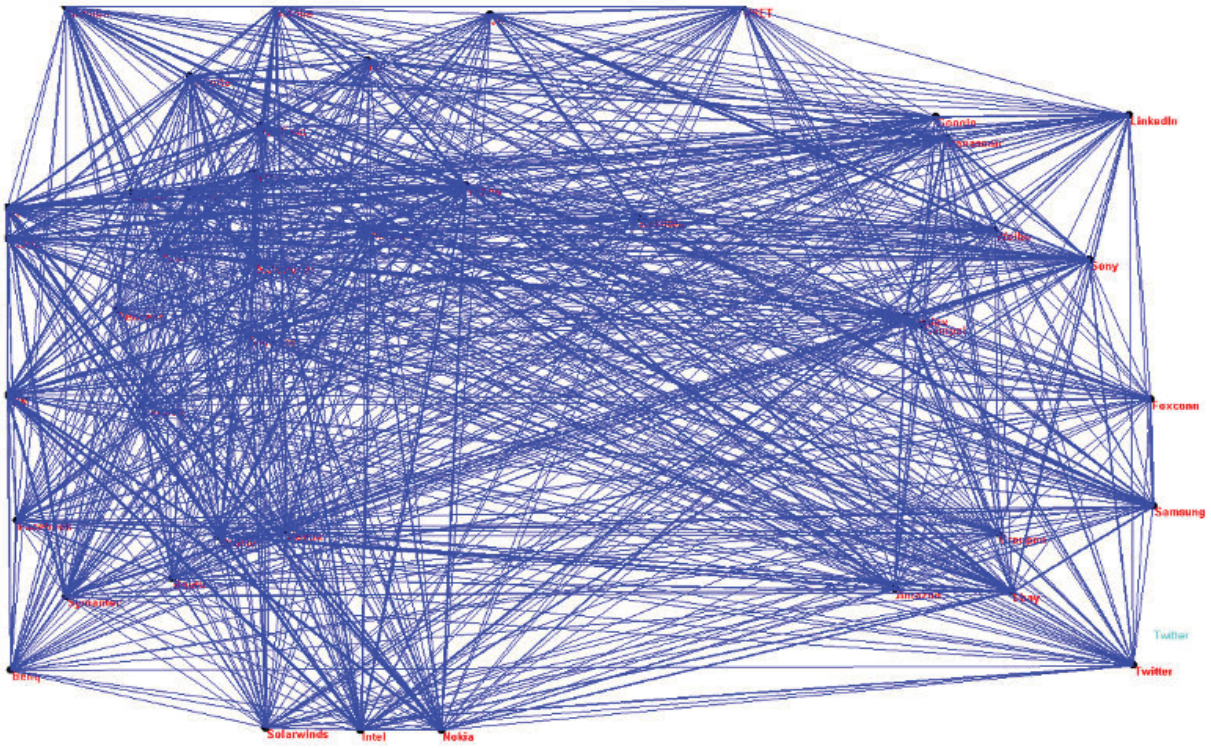
(c)Low Value



(e) Adjusted Value



(f) Volume Value



(g) Average Value

Fig. 4 Comparisons edge length and thickness

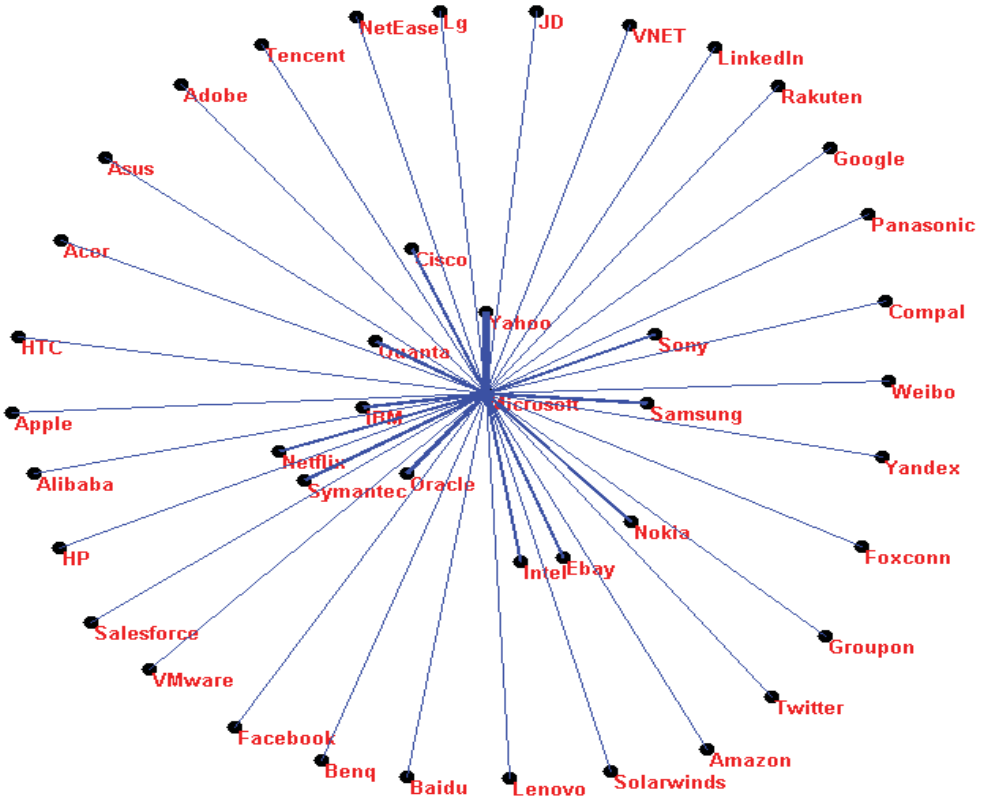


Fig. 5 Connection details of Microsoft

VI. Conclusion and Future Works

In this paper we have presented a new approach for potentially visualizing the relationships between stocks and presenting trends for individual stocks by combining *clustering* and *force-directed* algorithms based on long-term historical data analysis. The experimental results of the proposed method demonstrate their effectiveness in terms of providing investors reasonable visualized information to assist in their decision regarding stock investment; however, we have identified several issues or limitations that we need to address in our future works. These issues are listed below.

- Stock market analysis is complex, and affected by multiple factors, this methodology could only provide a view from another angle to assist investors to have a better understanding of potential relationships between each stock.
- The methodology is only applicable to stocks with a long history of available data (long-term) but cannot be adopted to those with limited data availability (short-term).
- The current *clustering* algorithm, although able to identify the connectivity between nodes, may cause wrong group divisions.

In our future works, we will apply the proposed revised method to a wider and larger set of data and applications. The future work will be addressing the limitations identified by:

- Carrying out further experiments that use more data.
- Revising the *clustering* algorithm to emphasis weight on each connection.
- Undertaking a study to formally evaluate the effectiveness of the proposed method.
- Working with experts in stock market to find more factors affect the finalized methodology.

References

[1] Cindy Wang. 2014. Investing 101: How to Analyze Stock Market Trends. (ONLINE) Available at: http://blog.sprinklebit.com/investing-101-how-to-analyze-stock-market-trends/. (Accessed 13 July 15).

[2] Omote, H. & Sugiyama, K. 2007, Force-Directed Drawing Method for Intersecting Clustered Graphs, APVIS 2007, 6th International Asia-Pacific Symposium on Visualization 2007, 5-7 February 2007, Sydney, Australia, pp.85-92

[3] Huang, X. & Lai, W., 2005, clustering graphs for visualization via node similarities. J. Vis. Lang. Comput. 17, 3 (June 2006), pp. 225-53. Elsevier Ltd.

[4] Huang, M., Nguyen, Q. V., A space efficient clustered visualization of large graphs Image and Graphics, 2007. ICIG 2007. Fourth International Conference on, pp. 920-27

[5] Huang, M.L., & Nguyen, Q.V. 2007, Navigating Large Clustered Graphs with Triple-Layer Display, 11th International Conference Information Visualization, 2-6 July 2007, Zürich, Switzerland, pp. 684-92.

[6] Sarkar, M. & Brown, M.H. 1994, 'Graphical fisheye views', Communications of the ACM, Volume 37 Issue 12, pp. 73–83.

[7] Huang, M.L., Eades, P. & Wang, J. 1998, On-line animated visualization of huge graphs using a modified spring algorithm, Journal of Visual Language and Computing, no. vl980093, pp. 623-45.

[8] Huang, M.L., Eades, P. & Cohen, R.F. 1998, Webofdav – navigating and visualizing the web on-line with animated context swapping, Computer Networks and ISDN Systems, 30(1), pp. 638-42.

[9] Nguyen, Q.V. & Huang, M.L. 2005: EncCon: an approach to constructing interactive visualization of large hierarchical data. Information Visualization 4(1): 1-21.
[10] Qiu, M., Zhang, K., Huang, M., 2006, Usability in mobile interface browsing, Web Intelligence and Agent Systems, Vol.4 (1), pp. 43-59, IOS Press.
[11] Nguyen, Q. V. and Huang, M. L., A focus+ context visualization technique using semi-transparency, The Fourth International Conference on Computer and Information Technology, 2004. CIT'04, pp. 101-08.
[12] Brandes, U., Delling, D., Gaertler, M., Gorke, R., Hoefer, M., Nikoloski, Z., Wagner, D., On Modularity Clustering, Knowledge and Data Engineering, IEEE Transactions on, pp. 172 - 88 Volume: 20, Issue: 2, Feb. 2008.
[13] Battista, G.D., Eades, P., Tamassia, R. & Tollis, I.G. 1999, Graph drawing algorithms for the viisualization of graphs, Prentice-Hall, New Jersey, USA.
[14] Eades, P., A heuristic for graph drawing. Congress Numerantium, 42:149-160, 1984.
[15] Lin, C.C., Yen, H.C. 2005, A New Force-Directed Graph Drawing Method Based on Edge-Edge Repulsion, Ninth International Conference on Information Visualisation, 6-8 July 2005, London, UK, pp.329-24
[16] Battista, G.D., Eades, P., Tamassia, R. & Tollis, I.G. 1999, Graph drawing algorithms for the visualization of graphs, Prentice-Hall, New Jersey, USA.
[17] Lin, C.C. & Yen, H.C. 2008, A new force-directed graph drawing based on edge-edge repulsion, 9th International Conference on Information Visualization IV2008, 6-8July, London, England, pp. 329-34.
[18] Hua, J. & Huang M.L. (2013). Improving the Quality of Clustered Graph Drawing through a Dummy Element Approach. In Computer Graphics, Imaging and Visualization (CGIV), 2013 10th International Conference. Macau, 6-8 Aug. pp. 88-92.
[19] Hua, J., Huang, M.L & Nguyen, Q.V, (2014). Drawing Large Weighted Graphs Using Clustered Force-Directed Algorithm. In Information Visualisation (IV), 2014 18th International Conference on. Paris, 16-18 July 2014. Paris: IEEE. pp. 13-17.
[20] Yahoo finance. 2016. Yahoo!7 Finance. (ONLINE) Available at: https://au.finance.yahoo.com/q. (Accessed 11 February 16).