Araştırma Makalesi
Adıyaman Üniversitesi
Mühendislik Bilimleri Dergisi
8 (2021) 169-178

UNSUPERVISED MACHINE LEARNING ALGORITHMS TO FIND 3-KCP SOLUTION: MODULARITY, CLIQUE PERCOLATION, SPECTRAL, CENTRALITY, AND HIERARCHICAL CLUSTERING

Serkan GÜLDAL[1]*

[1]Department of Physics, Faculty of Arts and Sciences, Adiyaman University, Adiyaman, 02040, Turkey
Geliş tarihi: 25.03.2021 Kabul tarihi: 07.06.2021

ABSTRACT
Unsupervised learning algorithms are used in many engineering applications since they extract information by the minimum human interaction. Modularity is one of the well known unsupervised machine learning algorithms to classify the network of information, so the relational information is highlighted. The closely related data points gather together to create relatively dense subcommunities. Thus, the meaningful relationships between data points (a.k.a. nodes or vertices) could be extended to the common properties of the data points. In this study, we aim to solve 3-KCP which is inherently a combinatorics problem by the modularity classification. Additionally, we compared the result with the well-known clustering algorithms, namely Clique Percolation, Spectral, Centrality, and Hierarchical clustering. Our investigation by means of modularity includes extended resolutions from 0.1 to 2.1, and 1.0 is the optimum resolution to find all 3-KCP solutions for the Modularity algorithm. Comparison of the modularity with the specified clustering algorithms shows the superiority of the modularity algorithm because compared algorithm provides wrong clusters by means of N-KCP or identified clusters by modularity.

Keywords: *Knight Graph; Modularity; Unsupervised Learning; 3-KCP*

3-KCP ÇÖZÜM BULMAK İÇİN DENETİMSİZ ÖĞRENME ALGORİTMASI: MODÜLERLİK, KLİK SÜZME, SPEKTRAL, MERKEZİYET VE HİYERARŞİK KÜMELEME

ÖZ
Denetimsiz öğrenme algoritmaları, bilgiyi minimum insan etkileşimi ile çıkardıkları için birçok mühendislik uygulamasında kullanılırlar. Modülerlik verileri sınıflandırmak için iyi bilinen denetimsiz öğrenme algoritmalarından biridir, böylece ilişkisel bilgiler vurgulanır. Yakın ilişkili veri noktaları, nispeten yoğun alt topluluklar oluşturmak için bir araya gelir. Böylece, veri noktaları (yada düğümler) arasındaki anlamlı ilişkiler, veri noktalarının ortak özelliklerine genişletilebilir. Bu çalışmada, modülerlik sınıflandırması ile doğası gereği kombinatorik bir problem olan 3-KCP'yi çözmeyi amaçlıyoruz. Bununla birlikte sonuçlarımızı iyi bilinen kümeleme algoritmaları Klik Süzme, Spektral, Merkeziyet ve Hiyerarşik kümeleme ile karşılaştırdık. Araştırmamız, 0.1'den 2.1'e genişletilmiş çözünürlükleri içerir ve 1.0, tüm 3-KCP çözümlerini bulmak için en uygun çözünürlük olduğunu gösteriyor. Modülerliğin belirtilen kümeleme algoritmalarıyla karşılaştırılmamız modülerlik algoritmasının diğer metotlara göre daha avantajlı olduğu gösterdi çünkü karşılaştırılan algoritma, N-KCP anlamında yanlış kümeler veya modülerlikle tanımlanan kümeler olarak sonuçlandı.

Keywords: *At Çizelgesi; Modülerlik; Denetimsiz öğrenme; 3-KCP*

1. Introduction

Unsupervised Machine Learning (UML) methods are gaining momentum while the collected data increases. UML classifies the subjected data without instruction to extract the desired information. UML improves the results as it progresses, so the initial results are more error prone than previously generated outputs. For example, the fast unfolding modularity algorithm is one of the well-known UML to cluster

*e-posta: sguldal@adiyaman.edu.tr ORCID ID: https://orcid.org/0000-0002-4247-0786 (Sorumlu Yazar)

graphs [1]. The clustered graphs presents the close connections within the graph. Moreover, the UML is not limited with graph. The UML is utilized for Odometry [2], Big Data Analysis [3], face recognition [4], security [5], intelligent image analysis [6], video applications [7, 8], digital signal processing [9, 10], Voice Activity Detection [11], and energy systems [12] because of it is adaptability and superior understanding of the system and the material. We used UML to solve Knight Covering Problem (N-KCP).

The knight which is the main component of N-KCP is in a special place for the chess players because of surprising the opponent by its unique movement capability. The knight moves have been utilized in image encryption [13, 14] and many other algorithms [15-17]. Similarly, the knight's tour problem is studied solely by researchers because of its entertaining nature [18]. The knight move capability created a playground to the numerous versions of N-KCP which are investigated continuously by better algorithms [19-26]. There are many methods to solve the knight graphs to find Knight Covering Problems' solutions such as the independent set method [27, 28] and Girvan-Newman clustering algorithm [29]. Moreover, N-KCP has the potential to be used for permutation generation by the identified solutions.

In this study, we focused on 3-KCP by various clustering methods. 3-KCP has 9 cells to be occupied by the knights as shown in Figure 1. In Figure 2, the knight graph representation is presented. It is generated by the movements of the knights in the indexed board. For example, the knight which is placed in cell number 1 has ability to attack cells number 6 and 8. Thus, a knight in the cell 1 can cover 3 cells including occupied cell. This is true for all cells but cell 5. Similarly, the full capability of every cell is shown in the Figure 1.

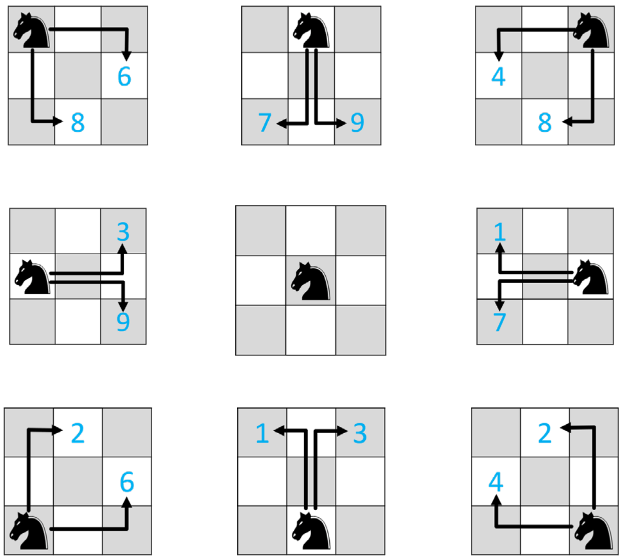
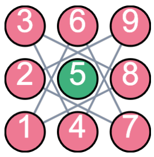


Figure 1. 3-KCP has 9 cells to position the knight

3-KCP graph is generated solely by reachability of the cells, and it is depicted by color-coded nodes in Figure 2. The colors of the nodes stand for the degrees of the nodes which are 2 and 0. The knights are on the corners and edges (dark pink) can attack to 2 nodes. In other words, they are 2-degree nodes. The other only node, node 5, has 0 degree because of the 3x3 board topology. The average degree is 1.778. The network diameter is 4 with a 2.2857 average path length. The density of the overall network is 0.222. Every knight can attack 2 cells since either corner or edge cell, but the knight is positioned in cell 5. Since the other cells too close, it is beyond the knight attacks' capability.



Color code	Degree	Percentage in the graph (%)
green	0	11.11
pink	2	88.89

Figure 2. 3-KCP graph is colored with respect to the degrees

There are developed algorithms to solve N-KCP by utilizing knight graphs. In Figure 3, all solutions of 3-KCP are found by the independent set algorithm introduced by Güldal et al. [27].

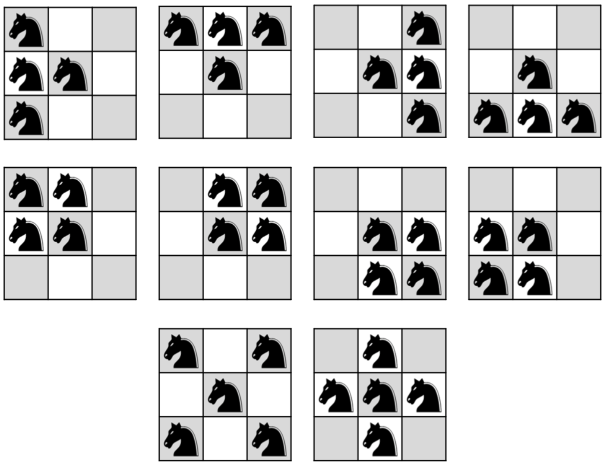


Figure 3. All 3-KCP solutions

In this study, the knight graph illustration of 3-KCP is analyzed by modularity phenomena with comparison of well known algorithms.

2. Materials and Methods

In our study, we used clustering algorithms, such as modularity, to extract subcommunities [30, 31]. Thus, extracted subcommunities present the knights which are the least likely to be in the same solution. The modularity is formulated as follows:

$$Q = \frac{1}{2}\sum_{i,j}\left(A_{ij} - \gamma\frac{k_i k_j}{M}\right)s_i s_j \qquad (1)$$

where $s_i s_j$ is 1 if i and j are in the same cluster. M stands for the number of edges in the graph. γ is the resolution. k_i is the degree of node i and k_j is the degree of node j. A_{ij} is the adjacency matrix.

Our application is done by the Gephi [1, 32, 33]. We applied the resolution from 0.1 to 2.1.

Additionally, the other clustering algorithms which are Clique Percolation, Spectral, Centrality, and Hierarchical are implemented to extract solutions of 3-KCP.

The application and analysis will be discussed further in the Results and Discussion section.

3. Results and Discussion

In this study, we investigate 3-KCP graph by means of graph modularity. Thus, we obtain all solutions of 3-KCP. There are extracted communities from 2 to 7 with respect to the resolutions.

In Figure 5, applications of modularity on 3-KCP (resolution = 0.1 - 2.1) graphs are shown. The resolution 0.1 is divided 3-KCP to 5 subcommunities in Figure 5.a. One of the communities has 1 node which degree is 0. Thus, no edge is connected it is a subcommunity by itself throughout the investigation. The other 8 nodes are divided equally, 2 nodes (1-8, 2-9, 3-4, and 6-7), per sub-community. For resolution 0.2, independent nodes appear as clusters; 4, 5, and 7 are single node subcommunities as seen in Figure 5.b. Node 3 is paired with node 8, and node 1 is paired with node 6. There is no change for 2-9 sub-community. Increasing resolution to 0.3 has 5 distinct subcommunities Figure 5.c. There is no effect on 1-6 and 3-8 pairs, but node 7 is paired with node 2, node 9 is paired with node 4. In Figure 5.d, the resolution increased to 0.4 causes bigger subcommunities while the number of communities is reduced to 4. In the new topology, 1-3-8 and 2-6-7 are the major subcommunities, 3-8 is paired, and node 5 is the single node sub-community. In Figure 5.e and f show the increase the resolution to 0.5 and 0.6, respectively, are divided 5 subcommunities. 4 of these communities have 2 nodes, and one has only one node. In Figure 5.g, h, and i, for the resolutions 0.7, 0.8, and 0.9 are divided 3-KCP graph to 4 subcommunities similar to resolution 0.4, so 2 subcommunities include 3 nodes each, and the other subcommunities have 2 nodes and 1 node. In Figure 5.j, the resolution for 1.0 divides the graph 5 subcommunities which have 2 nodes and 1 node. In Figure 5.k, resolution 1.1 is applied, the 3 subcommunities are extracted. One of them is a single node community and the other subcommunities are divided into 4 nodes evenly. In Figure 5.l, m, n, and o, resolutions are 1.2, 1.3, 1.4, and 1.5 are applied. By these resolutions, 4 subcommunities are extracted. 2 subcommunities have 3 nodes each, 1 subcommunity has 2, and the last community has 1 node. For resolution 1.6, 3 subcommunities are generated from the 3-KCP graph which is shown in Figure 5.p. The biggest subcommunity has 5 nodes, the next one has 3 nodes, and the smallest community has only 1 node. Increase the resolution to 1.7 changed topology as shown in Figure 5.q. Thus, this resolution distributed nodes evenly, 4 nodes for each. There is a similar subcommunities for the increase by means of resolution (1.8, 1.9, 2.0) as presented in Figure 5.r, s, and t. However, there is some change for the distribution of the clusters. Resolutions 2.1 and higher, there are 2 subcommunities. The small community has node number 5 which has 0 degrees. The big subcommunity has 8 nodes, the rest of the board. Thus, 2.1 is the biggest resolution for meaningful results. The clustering against to changing resolution is summarized in Figure 6.

There are a few comments on the overall modularity application. Node 5 has 0 degrees, so it is not affected by resolution changes. In other words, it is a part of the community which has a single node. Thus, it could be omitted while the new algorithms are developed and implemented. It is visibly seen for all solutions include a knight for cell 5 which is shown in Figure 3. The second comment is that there is a strong presence of symmetry while the 3-KCP graph is divided to subcommunities. The change in terms of resolutions causes pairing to node the other direct node by generating the same topology. It could be seen clearly by the circular layout in Figure 4. The other comment is that there is no direct correlation between the number of communities and the modularity resolution. Lastly, the direct connection plays a crucial role. Before secondary and tertiary connections affect the modularity, other nodes which have direct connection dominates the modularity.

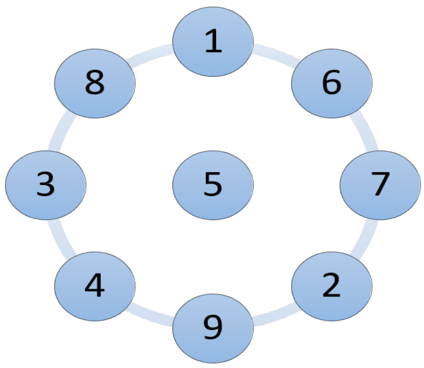
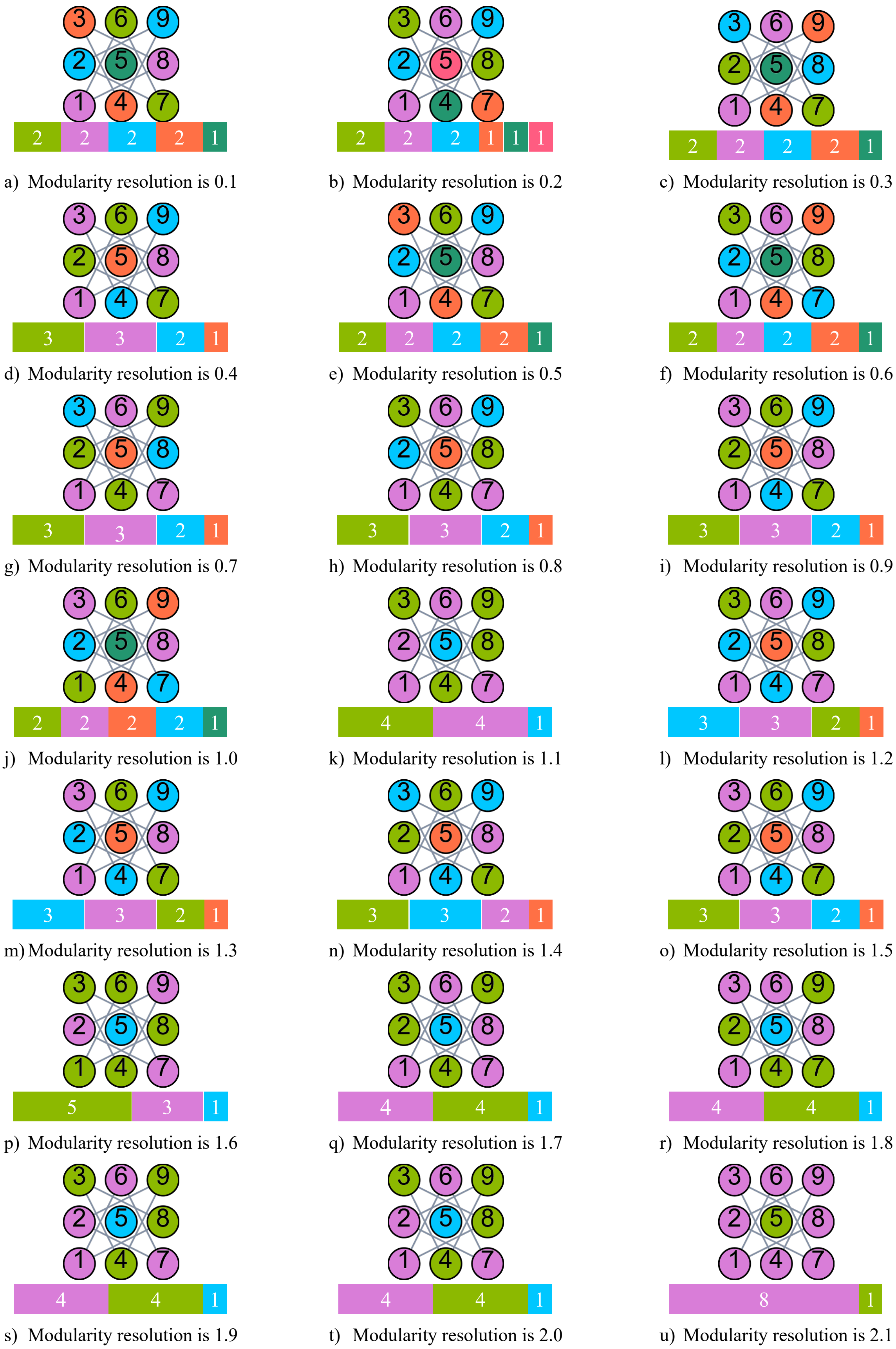


Figure 4. The circular layout of 3-KCP graph



a) Modularity resolution is 0.1
b) Modularity resolution is 0.2
c) Modularity resolution is 0.3
d) Modularity resolution is 0.4
e) Modularity resolution is 0.5
f) Modularity resolution is 0.6
g) Modularity resolution is 0.7
h) Modularity resolution is 0.8
i) Modularity resolution is 0.9
j) Modularity resolution is 1.0
k) Modularity resolution is 1.1
l) Modularity resolution is 1.2
m) Modularity resolution is 1.3
n) Modularity resolution is 1.4
o) Modularity resolution is 1.5
p) Modularity resolution is 1.6
q) Modularity resolution is 1.7
r) Modularity resolution is 1.8
s) Modularity resolution is 1.9
t) Modularity resolution is 2.0
u) Modularity resolution is 2.1

Figure 5. Modularity is applied 3-KCP graphs for various resolutions.

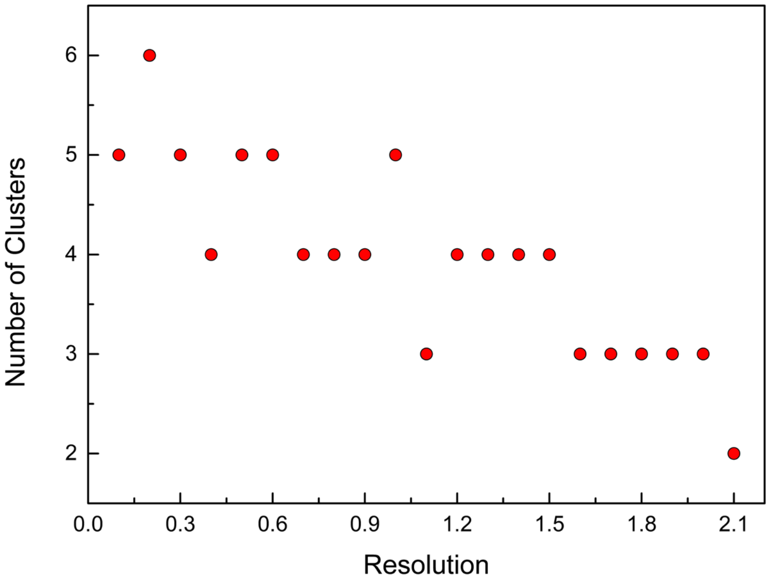
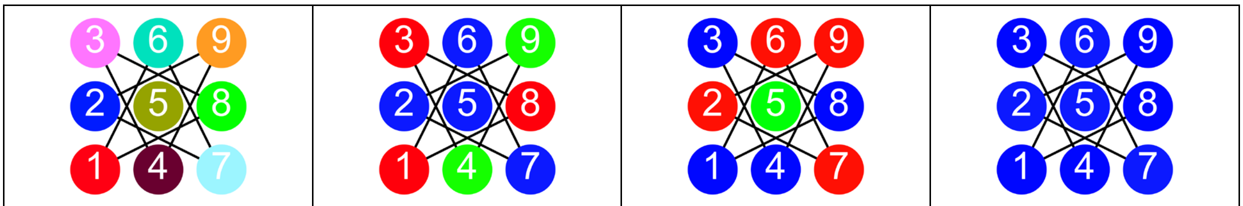


Figure 6. The resolution change had no explicit relation with number of clusters.

Modularity phenomena uses the strength of the connections between nodes. However, the N-KCP problem intents to place the knight which does not touch each other. Thus, after modularity is applied, the listed clusters show the list of cells that are the least likely to be in the same solution.

For the resolution 1.0, clusters are defined as follows: (1, 6) (2, 7) (3, 8) (4, 9) (5). In these groups, the nodes are mutually exclusive such as 1 and 6 cannot exist in the same solution. If we construct the possible solutions, there are three options (empty, 1, and 6) for the first step. There is a similar pattern for the following steps. All possible combinations are given in the appendix. In the appendix, 3-KCP solutions are highlighted which are all possible solutions.

The other clustering algorithms are compared with the modularity algorithm by means of 3-KCP solutions. In Figure 7, the clustered graphs are presented. In the figure, the nodes in the same color present the same clusters.



a) Clique Percolation	b) Spectral	c) Centrality	d) Hierarchical

Figure 7. Graph clustering algorithms, namely Clique Percolation, Spectral, Centrality, and Hierarchical, are applied to 3-Knight graph. The same colors show the same cluster.

The Clique Percolation divided to 9 clusters, so this provides no extra information. The spectral clustering algorithm is merged node 5 in another cluster although, it has no common edge with the nodes. Thus, the clusters are not meaningful for 3-KCP solutions. The centrality algorithm is identified the cluster which are extracted by modularity algorithm for resolution 1.7. The hierarchical clustering methods extracted only one cluster contrary to Clique Percolation clustering, so no node could be in the same solution. Therefore, hierarchical clustering method provides no solution.

4. Conclusion

In this study, we have applied the unsupervised machine learning algorithms to 3x3 knight graph for clustering purpose. Modularity has superiority over Clique Percolation, Spectral, Centrality, and Hierarchical clustering algorithms by means of 3-KCP graph. The extracted clusters by modularity corresponds to all 3-

KCP solutions. Thus, the modularity covers the meaningful clusters by the other algorithms. In this study, the investigation results showed 1.0 resolution divides the knight graph in such a way which includes all solutions. On the other hand, the other algorithms provide wrong clusters or not valuable clusters. Only Centrality algorithm provides useful clusters which are already identified by modularity.

Based on the promising results, the modularity algorithm provides incomputable results. In the future studies, we purpose to extend our investigation to different nifty problems similar to 3-KCP.

Acknowledgments:

The authors declare no conflict of interest.

References

[1] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, E. J. J. o. s. m. t. Lefebvre, and experiment, "Fast unfolding of communities in large networks," vol. 2008, no. 10, p. P10008, 2008.

[2] Y. Tian and M. Compere, "A Case Study on Visual-Inertial Odometry using Supervised, Semi-Supervised and Unsupervised Learning Methods," in *2019 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, 9-11 Dec. 2019 2019, pp. 203-2034, doi: 10.1109/AIVR46125.2019.00043.

[3] H. Yan, P. Yu, and D. Long, "Study on Deep Unsupervised Learning Optimization Algorithm Based on Cloud Computing," in *2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*, 12-13 Jan. 2019 2019, pp. 679-681, doi: 10.1109/ICITBS.2019.00168.

[4] M. S. Bartlett, J. R. Movellan, and T. J. Sejnowski, "Face recognition by independent component analysis," *IEEE Transactions on Neural Networks,* vol. 13, no. 6, pp. 1450-1464, 2002, doi: 10.1109/TNN.2002.804287.

[5] M. A. Kabir and X. Luo, "Unsupervised Learning for Network Flow Based Anomaly Detection in the Era of Deep Learning," in *2020 IEEE Sixth International Conference on Big Data Computing Service and Applications (BigDataService)*, 3-6 Aug. 2020 2020, pp. 165-168, doi: 10.1109/BigDataService49289.2020.00032.

[6] Q. Li, J. Zhao, and X. Zhu, "An Unsupervised Learning Algorithm for Intelligent Image Analysis," in *2006 9th International Conference on Control, Automation, Robotics and Vision*, 5-8 Dec. 2006 2006, pp. 1-5, doi: 10.1109/ICARCV.2006.345232.

[7] Y. Wang and Y. Xu, "Unsupervised Learning of Accurate Camera Pose and Depth From Video Sequences With Kalman Filter," *IEEE Access,* vol. 7, pp. 32796-32804, 2019, doi: 10.1109/ACCESS.2019.2903871.

[8] Y. He, N. Sang, C. Gao, and J. Han, "Online Unsupervised Learning Classification of Pedestrian and Vehicle for Video Surveillance," *Chinese Journal of Electronics,* vol. 26, no. 1, pp. 145-151, 2017, doi: 10.1049/cje.2016.08.011.

[9] C. Teng and Y. Chen, "Syndrome-Enabled Unsupervised Learning for Neural Network-Based Polar Decoder and Jointly Optimized Blind Equalizer," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems,* vol. 10, no. 2, pp. 177-188, 2020, doi: 10.1109/JETCAS.2020.2992593.

[10] T. C. Clancy, A. Khawar, and T. R. Newman, "Robust Signal Classification Using Unsupervised Learning," *IEEE Transactions on Wireless Communications,* vol. 10, no. 4, pp. 1289-1299, 2011, doi: 10.1109/TWC.2011.030311.101137.

[11] D. Ying, Y. Yan, J. Dang, and F. K. Soong, "Voice Activity Detection Based on an Unsupervised Learning Framework," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 19, no. 8, pp. 2624-2633, 2011, doi: 10.1109/TASL.2011.2125953.

[12] Y. Chang, X. Yuan, B. Li, D. Niyato, and N. Al-Dhahir, "A Joint Unsupervised Learning and Genetic Algorithm Approach for Topology Control in Energy-Efficient Ultra-Dense Wireless Sensor Networks," *IEEE Communications Letters,* vol. 22, no. 11, pp. 2370-2373, 2018, doi: 10.1109/LCOMM.2018.2870886.

[13] A. Philip, "A Generalized Pseudo-Knight?s Tour Algorithm for Encryption of an Image," *IEEE Potentials,* vol. 32, no. 6, pp. 10-16, 2013, doi: 10.1109/MPOT.2012.2219651.

[14] J. Kumar and S. Nirmala, "Securing the contents of document images using knight moves and genetic approach," in *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 10-13 Aug. 2015 2015, pp. 1091-1095, doi: 10.1109/ICACCI.2015.7275755.

[15] J. Delei, B. Sen, and D. Wenming, "An Image Encryption Algorithm Based on Knight's Tour and Slip Encryption-Filter," in *2008 International Conference on Computer Science and Software Engineering*, 12-14 Dec. 2008 2008, vol. 1, pp. 251-255, doi: 10.1109/CSSE.2008.1142.

[16] S. Bai, G. Zhu, and J. Huang, "An Intelligent Algorithm for the (1,2,2)-Generalized Knight's Tour Problem," in *2013 Ninth International Conference on Computational Intelligence and Security*, 14-15 Dec. 2013 2013, pp. 583-588, doi: 10.1109/CIS.2013.129.

[17] H. Jian and B. Sen, "An Efficient Algorithm for the Generalized (1,k)-Knight's Tours Problem," in *2009 First International Workshop on Education Technology and Computer Science*, 7-8 March 2009 2009, vol. 1, pp. 697-701, doi: 10.1109/ETCS.2009.161.

[18] S. Bai, X. Liao, X. Qu, and Y. Liu, "Generalized Knight's Tour Problem and Its Solutions Algorithm," in *2006 International Conference on Computational Intelligence and Security*, 3-6 Nov. 2006 2006, vol. 1, pp. 570-573, doi: 10.1109/ICCIAS.2006.294200.

[19] D. C. Fisher, "On the n x n Knight Cover Problem," *Ars Comb.,* vol. 69, / 2003.

[20] F. Rubin, "Improved Knight Coverings," *Ars Combinatoria,* vol. 69, / 2003.

[21] F. Rubin, "Knight Covers for the 50x50 Chessboard," presented at the Mathfest 2004, Providence RI, 2004.

[22] F. Rubin, "A Family of Efficient Knight Covering Patterns," *Journal of Recreational Mathematics,* Article vol. 33, no. 3, pp. 165-175, 2005.

[23] F. Rubin, "An Improved Method for Finding Knight Covers," *Ars Combinatoria,* vol. 82, / 2007.

[24] B. Lemaire, "Knights Covers on NxN Chessboards," *J. Recr. Math.,* vol. 31, pp. 87-99, 2003.

[25] A. H. Jackson and R. P. Pargas, "Solutions to the NxN Knights Covering Problem," *Journal of Recreational Mathematics* vol. 23, pp. 255-267, 1991.

[26] F. Wei, "Research on Knight Covering Based on Breadth First Search Algorithm," (in English), *Applied Mechanic and Materials,* vol. 686, pp. 377-380, 2014.

[27] S. Güldal, M. Lipscomb, and M. M. Tanik, "Solving Knights Covering Problem: Backtracking, Permutation, Bipartite Graph, and Independent Set," presented at the Nineteenth Annual Early Career Technical Conference, Birmingham, Alabama USA, 2019.

[28] S. Güldal, M. M. Tanik, and M. M. Lipscomb, "Solving Knights Covering Problem by a Hybrid Algorithm," presented at the IEEE SouthEastConn, Huntsville, Alabama, April 11 - 14 2019, 2019.

[29] S. Güldal, "Connectives of Knights Covering Problem By Girvan-Newman Clustering," presented at the SDPS 2019 Workshop, Madrid, Spain, November 25-26 2019, 2019.

[30] U. Brandes *et al.*, "On Modularity Clustering," *IEEE Transactions on Knowledge and Data Engineering,* vol. 20, no. 2, pp. 172-188, 2008, doi: 10.1109/TKDE.2007.190689.

[31] R. Rotta and A. Noack, "Multilevel local search algorithms for modularity clustering," vol. 16, p. Article 2.3, 2011, doi: 10.1145/1963190.1970376 %J J. Exp. Algorithmics.

[32] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: an open source software for exploring and manipulating networks," in *International AAAI Conference on Weblogs and Social Media*, 2009.

[33] R. Lambiotte, J.-C. Delvenne, and M. J. a. p. a. Barahona, "Laplacian dynamics and multiscale modular structure in networks," 2008.

APPENDIX

Table 1. Empty cells mean no knight is placed from the particular cluster. For resolution 1.0: (1, 6) (2, 7) (3, 8) (4, 9) (5)

First cluster	Second cluster	Third cluster	Fourth cluster	Fifth cluster
				5
			4	
			4	5
			9	
			9	5
		3		
		3		5
		3	4	
		3	4	5
		3	9	
		3	9	5
		8		
		8		5
		8	4	
		8	4	5
		8	9	
		8	9	5
	2			
	2			5
	2		4	
	2		4	5
	2		9	
	2		9	5
	2	3		
	2	3		5
	2	3	4	
	2	3	4	5
	2	3	9	
	2	3	9	5
	2	8		
	2	8		5
	2	8	4	
	2	8	4	5
	2	8	9	
	2	8	9	5
	7			
	7			5
	7		4	
	7		4	5
	7		9	
	7		9	5
	7	3		
	7	3		5
	7	3	4	
	7	3	4	5
	7	3	9	
	7	3	9	5
	7	8		
	7	8		5
	7	8	4	
	7	**8**	**4**	**5**
	7	8	9	
	7	**8**	**9**	**5**
1				
1				5
1			4	
1			4	5
1			9	
1			9	5
1		3		
1		3		5
1		3	4	
1		3	4	5
1		3	9	
1		3	9	5
1		8		
1		8		5
1		8	4	
1		8	4	5
1		8	9	
1		8	9	5
1	2			
1	2			5
1	2		4	
1	**2**		**4**	**5**
1	2		9	
1	2		9	5
1	2	3		
1	**2**	**3**		**5**
1	2	3	4	
1	2	3	4	5
1	2	3	9	
1	2	3	9	5
1	2	8		
1	2	8		5
1	2	8	4	
1	2	8	4	5
1	2	8	9	
1	2	8	9	5
1	7			
1	7			5
1	7		4	
1	**7**		**4**	**5**
1	7		9	
1	7		9	5
1	7	3		
1	7	3		5
1	7	3	4	
1	7	3	4	5
1	7	3	9	
1	**7**	**3**	**9**	**5**
1	7	8		
1	7	8		5
1	7	8	4	
1	7	8	4	5
1	7	8	9	
1	7	8	9	5
6				
6				5
6			4	
6			4	5
6			9	
6			9	5
6		3		
6		3		5
6		3	4	
6		3	4	5
6		3	9	
6		**3**	**9**	**5**
6		8		
6		8		5
6		8	4	
6		8	4	5
6		8	9	
6		**8**	**9**	**5**
6	2			
6	2			5
6	2		4	
6	2		4	5
6	2		9	
6	2		9	5
6	2	3		
6	**2**	**3**		**5**
6	2	3	4	
6	2	3	4	5
6	2	3	9	
6	2	3	9	5
6	2	8		
6	2	8		5

6	2	8	4	
6	2	8	4	5
6	2	8	9	
6	2	8	9	5
6	7			
6	7			5
6	7		4	
6	7		4	5
6	7		9	
6	7		9	5
6	7	3		
6	7	3		5
6	7	3	4	
6	7	3	4	5
6	7	3	9	
6	7	3	9	5
6	7	8		
6	7	8		5
6	7	8	4	
6	7	8	4	5
6	7	8	9	
6	7	8	9	5