Community Detection Based on Isomorphic Subgraph Analytics in Criminal Network

Theyvaa Sangkaran, Azween Abdullah, NZ. Jhanjhi
School of Computing and IT (SoCIT) Taylor's University, Malaysia, School of Computing and IT (SoCIT) Taylor's University, Malaysia, School of Computing and IT (SoCIT) Taylor's University, Malaysia

Abstract
Community detection using graph theory allows us to detect a community within an organized criminal syndicate that has network orientated structures. Classical community detection methods will have problems to detect communities with different network orientated structures even though they have similar nodes. Studying the inter-connections between the nodes by employing isomorphic subgraph analytics allows the researchers and law enforcement agencies to understand and to determine the key participants and the criminals' modus operandi of illicit operations. One of the domains which we have selected to work on is criminal network analysis as there is a lack of new perspective in the Criminal Network Analysis (CNA), which is urgently required as the modus operandi behind crimes are considerably complex now. We studied community detection in criminal networks using graph theory and formally introduced an algorithm that opened a new perspective of community detection compared to the traditional methods used to model the relations between objects using the isomorphic graph-based analytics. Community structure is an important property of complex networks, which is generally described as densely connected nodes and similar patterns of links. Our method differed from the traditional methods because our method allowed the law enforcement agencies to compare the detected communities, and this would allow a different point of view of the criminal network. This research allowed and assisted enforcement agencies and researchers to detect the same community from different patterns and structures by employing isomorphism. This would allow the detection of the communities that may not have been found using the traditional methods.

Key words
Cybersecurity, Graph Theory, Criminal Network, Ad-hoc networks, Social Network

1. Introduction

As mentioned earlier, the origin of graph theory is a historically critical problem in mathematics known as The Seven Bridges of Königsberg.[1] It has laid the foundation for graph theory. Since then, there has been a significant improvement in the graph theoretical concepts which have become widely used to study and model various applications, in different areas. One of the domains and one of the applications of the graph theory is Criminal Network Analysis (CNA), which is a subset of cybersecurity. There were a few research gaps in this particular domain in relation to isomorphism and isomorphic graph matching. [2]. Isomorphic graph matching has been a useful, enjoyable, and an excellent challenge for the research community as it has been used and collaborated in theoretical findings, and being a facilitating factor in the process of successful analysis of graphs with the combination of other analytical tools and methods. Table 1 presents the name of its components in different fields of study:

Table 1: Graph variable quantities in different fields stUdy. Adopted from [3]

Computer Science	Mathematics	Physics	Sociology	Computer Science
Node	Vertex/vertices	Site	Agent/actor	Node
Link /connection	Edge	Bond	Relational tie	Link /connection

2. Isomorphic Graphs

Graphs can be found in different forms but the graph that is related to the current research is the 'isomorphic' graphs. Two graphs, S1 and S2 are considered as isomorphic, when they meet the requirement where (a) the number of components contained in each graph is equivalent and if (b) the edge connectivity is preserved [4]. Furthermore, the isomorphic eligibility of two or more subgraphs can possibly be confirmed by using the criteria mentioned above. The finding of an isomorphic graphs is normally done by allowing certain algorithms to analyze the graph structure by keeping in mind the conditions mentioned above. Fundamentally, an isomorphic algorithm is a graph-matching method. The significance of pinpointing the isomorphic graphs is associated with deciding and connecting akin graphs or subgraphs among social networks, criminal networks or even within the network itself.

3. Community Detection

A complex system possesses constituents that are interrelated between themselves which is able represented as nodes (the composing elements of the system) and links (the known interconnection between nodes [5][6]. The structure of these systems holds topological information that contains viable information which are processed into solutions to solve the inherent problems of the complex system being represented as a graphical network.

Manuscript received May 5, 2020
Manuscript revised May 20, 2020

In a network, communities are also referred to as clusters, though with no quantitative definition. They are described as nodes with denser intra-group connections than inter-group connections [7]. These communities, also known as sub-graphs in graph analysis, are inherent and usually hidden, and hence the need for methods, algorithms or schemes to adequately detect them. Even more so, the definitions of the communities vary from discipline to discipline, ranging from extracting the most available number of subgraphs by having high enough density within a graph, to creating partitions in a given network in order to minimize interconnections between parts [8]. This characteristic of real networks is called community structure or community clustering [9][10].

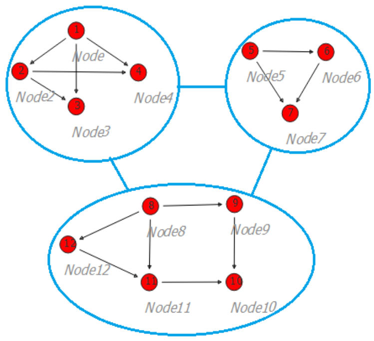

Fig. 1 A simple graph with three communities, enclosed by the circles

The goal of community detection is to partition the nodes of a given network with respect to the relationships between them, In other words, it is to create a group of nodes within a given network that represent strongly linked sub-network from the entire network. For the purpose of getting meaningful, usable, valuable and actionable information from the complex networks, the community structure of the given network possess important information which is being mined through community detection mechanisms. This is currently a dominant research areas since many real-world problems can be depicted, formulated, and solved through graph and its analyses [11]. By means of community detection, the internal network organization of a given network is being uncovered and it also provide better understanding of the characteristics inherent to the dynamic processes that are peculiar to the network.

4. Community Detection using Isomorphic Graphs in Criminal Network

Much research is being undertaken in community detection, visualization, dynamics in social networks, social/criminal network analysis. However, some of the areas are overlooked by the researchers, such as community detection using isomorphism of graph involving criminal network. Moreover, Criminal Network Analysis (CNA) as a whole is less addressed compared to Social Network Analysis (SNA). New perspective of Criminal Network Analysis (CNA) is highly required as the modus operandi behind crimes are much more complex now. By incorporating Community Detection, Isomorphic Graph Analytics with the use of Criminal Networks Analysis, which is still at an infant stage, it would enable us and others to easily analyse different structures of the graph and it would also enable us to extract isomorphic graph and analyze them further. This research will expedite the community detection and analyzing process by reducing the complexity in analyzing the data in bulk with the help of data visualization. Notwithstanding, most of the analysis and approaches applied to the criminal data considered the criminal networks to be a static graph and overlooked the fact that in reality these networks continue to evolve. Our algorithm can be used even with time evolving datasets to compare more than one community at a time. Unraveling the problem and experimenting with the matrices applied over big criminal network graphs would give us the chance to offer the networks a different structure. This would lead to the examination of different patterns, relationship within an organization or network, and also predicting future interactions and preferential attachments between groups of individuals within a criminal network. Previous studies showed no much attention to study the structural complexity of criminal and comparing them in a timely manner for further understanding. Experimenting with the model and metrics with similar nature will greatly contribute to the domain of cybersecurity.

A. Isomorphic Graph Algorithm

Our previous research undertaking [12] contributed an extensive survey of isomorphic graph algorithms usable for graph analytics. Basically, isomorphic (sub)graph algorithms were used for matching two graphs by ascertaining the mapping associated with the nodes of the involved graphs. The existence of different types of mapping is the result of how different types of constraints can be applied on the mapping, further leading to graph-subgraph isomorphism, monomorphism, automorphism, or strict isomorphism. Over the years, various algorithms to determine the isomorphism between graphs have been designed and implemented as well as comparatively evaluated. Some of these algorithms included the Ullman Algorithm [13], VF [14] and VF 2 [15] algorithm, Schmidt and Druffel method [16], Corneil and Gotlieb [17] method and Merging Lexicographic Chain (MLC) [18] algorithm. These algorithms have several varying underlying mathematical theories and concepts for determining isomorphism between the graphs. Moreover, as we uncovered in our previously published study, each of the highlighted isomorphic graph algorithms possesses its strengths and weaknesses.

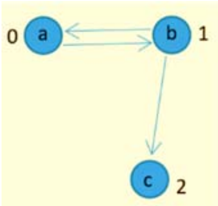

Fig. 2 An example of a query graph

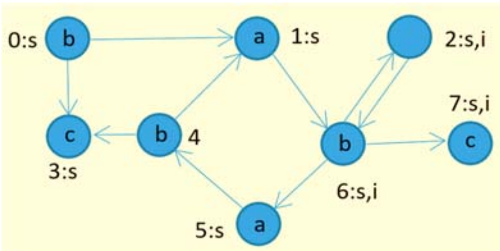

Fig. 3 An example of matched nodes of the query graph

Figure 2 and Figure 3: An example of a query graph (above) and its matches via simple, dual and isomorphism in a data graph (bottom). Simple matches are denoted by an 's', dual matches by a 'd' and isomorphic matches by an 'i'.

B. Community Detection Algorithm

Community detection is an important aspect of Social Network Analysis (SNA) as it decomposes a complex graphical network into smaller units for deeper analysis of relationships between the nodes. Though the communities are without quantitative definition, they are often referred to as a group of nodes with densely connected intra-group modes. The hidden nature of communities within a given network drives the continuous development of community detection algorithm, having optimal speed for community detection completion as well as efficiency in the area of computational cost reduction. In simple words, discovering and splitting up a group of complex nodes of a graphical network into groups of smaller densely connected intra-group nodes is referred to as community detection.

It is worth noting that how communities in a graphical network is detecting is an interesting area of research, and several quality researches had been conducted and published accordingly. The novelty of this research work lies within this scope. Community detection in a graphical network is carried out using several algorithms including, but not limited to, Random Walk, Clique Percolation Method (CPM), Link Graph and Link Partitioning, Simulated Annealing Fuzzy Detection, Local Expansion and Optimization, Statistical Inference based methods, Extremal Optimization, NMF and PCA based methods, Spin Models, Partitional Clustering, Graph Partitioning, Hierarchical Clustering, Spectral Clustering, Greedy Optimisation, Genetic Algorithms and Spectral Optimization as revealed, by [19]. The various community detection algorithms highlighted above are used to either detect overlapping communities or disjoint communities. The community detection algorithms used for the detection of disjoint communities are usually categorized into three (3) vis–a–vis traditional, dynamic, modularity-based algorithms. Each method is based on different mathematical theories and used in different practical applications and always outperform each other. Many of the aforementioned community detection algorithms have yet to be implemented, and evaluated on criminal network. Thus, this research work is a pivotal work on the topic of community detection involving a typical criminal graphical network.

Popular community detection algorithms used for SNA included the Factorised Asymptotic Bayesian (FAB) inference with Belief Propagation, an algorithm whose strengths lies in having no parameters, effecting consistent results over sparse and dense graph even as its extended version is known to lack consistency based on the usage of the Belief Propagation [20]. Another popular algorithm used for detecting community via SNA is the modularity optimization and maximum likelihood equivalence algorithm. This algorithm is known to produce consistent result in detecting communities when compared against its maximum-likelihood method though it heavily relies on, and is only applicable when, the resolution parameter value is correct [21]. Yet another popular SNA community detection algorithm is the Minimum Description Length (MDL) principle and multilevel Monte Carlo algorithm. This algorithm is capable of producing simple, unbiased, efficient and fully nonparametric analysis of large-scale features of a large network and, thus, is able to detect a reasonable number of communities within a given large network [22]. Lastly, Matrix Blocking community detection algorithm is popular in SNA mainly because it does not require the specification of the clusters nor the clustering of all network nodes. Matrix Blocking is also a popular choice due to its lowered computational cost as well as its applicability on different graph types such as undirected, directed and/or bipartite graph. Also, the ability of Matrix Blocking method to output singletons as a community serves as the chief limitation of this method.

5. Experimental Setup

The researchers planned to use one of the state-of-art isomorphism algorithms to complete the comparison with their new proposed method. Although some influential isomorphism algorithms, such as VF2 [23], VF3 [24], LAD [25], RI [26] and Nauty algorithm [27], are optimal and can yield perfect results in graph isomorphism detection, their computational complexity is not polynomial-time. On the other, the Girvan-Newman community detection model drastically reduces the running time of the calculation even though the resulting estimate of the betweenness necessarily suffers from the statistical fluctuations inherent in random sampling methods. Even though Girvan-Newman is computationally intensive, it is very suitable for small graphs, which happened to be the focus of our research due to the speed with which it can detect a community. As for the datasets used, we utilized a covert network dataset from UCINET, namely Cocaine Dealing Natarajan, which resulted from an investigation into to a large cocaine trafficking organization in New York City. Both experiments were conducted in the same environment. Examples of the metrics used to match the isomorphic graph were degree centrality, closeness centrality and common neighbours.

ROC Score

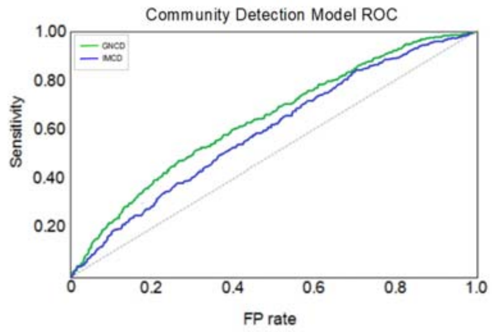

Fig. 4 ROC Score

The number of correct and incorrect communities detected within the network were also measured using the confusion matrix rate, and a receiver operating characteristic (ROC) curve was plotted. The true positive (TP) rate or also known as sensitivity rate, is the proportion of the actual missing links that are predicted as such by the models. The FP rate (1—specificity) represents the proportion of the non-existing links that have been identified as such by the models. However, the experimental results also seemed to indicate that the Girvan-Newman Community Detection (GNCD) model has a higher level of accuracy compared to the Isomorphic Community Detection (IMCD) model. Still, the GNCD takes a longer time to train with the same number of iterations. Furthermore, the IMCD model was able to identify probable communities that were not detected by the GNCD model.

AUC Score

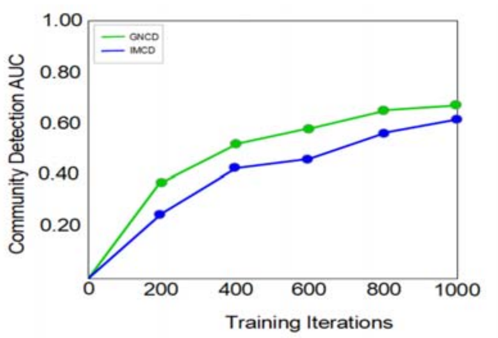

Fig. 5 AUC Score

The Area Under Curve (AUC) metric of a statistical learning metric indicates the predictive accuracy of the model and it has a value that lies between 0 and 1, where a higher value indicates a better predictive accuracy.
The AUC scores seem to indicate as the number of iteration of training had been performed, the IMCD model seems to achieve a level of accuracy nearer to the GNCD model. This could partly be affected by the initial base sub-graph by using just three nodes in the training of the IMCD model within the limit of 1000 iterations. Based on the results in Table 4.1, the IMCD model seems to use significantly less time for training purposes of achieving relatively similar AUC scores. The IMCD may provide complementary perspectives regarding the communities detected compared with the models developed using classical algorithms such as Girvan-Newman, which uses more computing time.

Table 1: AUC SCore

Dataset	AUC	Time-score(Hr)	Iterations
IMCD	0.62	0.77	1000
GNCD	0.68	1.61	1000

Furthermore, research may need to be conducted to confirm if the complexity of the IMCD can be increased by the number of nodes in the base matching sub-graph without a significant increase in the computing time required for training.

II. Proposed 'Heuristics Algorithm for Isomorphic Subgraph Matching' (HAISM) Modelled From Ullmann Algorithm

In this section, we explained the principle of the proposed HAISM algorithm.

Below is the pseudocode of HAISM algorithm. The refined pseudocode for the algorithm is as follows:
Name: Heuristics Algorithm for Isomorphic Subgraph Matching

START
Step 1: procedure simulation (G; Q; s);
Step 2: changed ←true;
Step 3: while changed do
Step 4: changed ←false
Step 5: for u←Vq do
Step 6: for u'←Q.adj(u) do
Step 7: for v←s(u) do
Step 8: if G.adj(v) ∩ s(u') = null; then
Step 9: remove v from s(u)
Step 10: if s(u) = null; then
Step 11: return empty s
Step 12: end if
Step 13: changed ←true
Step 14: end if
Step 15: end for
Step 16: end for
Step 17: end for
Step 18: end for
Step 19: return
Step 20:end procedure
STOP

Our algorithm is different from the traditional community detection techniques due to its ability to detect communities that the traditional algorithms fail to find by incorporating subgraph isomorphic matching features

Because $|s(0)| + \ldots + |s(|Vq|-1)| \leq |V||Vq|$, the outer while loop in Line 3 may execute at most $|V|\,|Vq|$ times. This would correspond to a scenario in which only one vertex is removed with each iteration of the loop. The next two for loops (lines 5–6) execute a total of $|Eq|$ times, and there are at most $|V|$ vertices in $|s(u)|$ for any $u \in Vq$, creating a bound on the innermost for loop (line 7). The intersection operation in line 8 takes at most $|V|$ steps, and so the total time complexity is O (|Eq||Vq||V |3). Note that vertex 0 is in s(1) despite the fact that it does not have a vertex in s(0) as a parent.

Firstly, given a Graph G, and a graph pattern Q, the objective of the isomorphic subgraph matching is to identify all the sub-graphs mappings, P from G, that are isomorphic to Q.

Input: pattern Q and graph G

Output: all isomorphic mappings P from Q to G

Matching procedure (P)

a) if P covers all nodes in Q then output P;

b) or else compute the set S(P) of all candidate pairs for inclusion in P

c) for each pair p = (u, v) in S(P)

  i) if p passes feasibility check

  ii) then P' ← P ∪ {p}; call Match(P');

d) Restore data structures

Reiteration refinement

For each pair p = (u, v) in S(P):

i) enumerate all possible extensions, for refinement

ii) if the feasibility test is not successful, drop it and try the next.

Table 2: This is a simplification of the results of the experiment

Algorithm Used	Figure No	Community Detected	Isomorphic Community Detected (P')	Remarks
Proposed HAISM algorithm	4	Input graph (Q')	Nodes- (6P, 28M, 22L)	Base sub graph (Q')
Proposed HAISM algorithm	5	YES	Nodes- (32L, 40L, 34L)	Matched Isomorphic Subgraph (Community 1)
Proposed HAISM algorithm	6	YES	Nodes- (35L, 14S, 31M)	Matched Isomorphic Subgraph (Community 2)
Proposed HAISM algorithm	7	YES	Nodes- (3L, 25S, 27M)	Matched Isomorphic Subgraph (Community 3)
Girvan-Newman algorithm	8	YES	Nodes- (35L, 14S, 31M)	Matched Isomorphic Subgraph (Overlap with Community 3)

As summarised and concluded in Table 2, the experimental results demonstrated that HAISM can detect communities that overlap with the well-studied Girvan-Newman algorithm. To explain this further, the proposed HAISM managed to identify a matching Community 3 as represented in Figure 7 (nodes 3L,25S,27M) as the community represented by red nodes. The experimental results show that HAISM is very effective in community detection problems and is superior to traditional community detection techniques in its ability to detect communities that the traditional algorithms fail to find and that some of the detected communities overlap. This gives assurance that the detected communities are correct and the one that does not overlap opens up a new perspective in the domain of communities by incorporating subgraph isomorphic matching features.

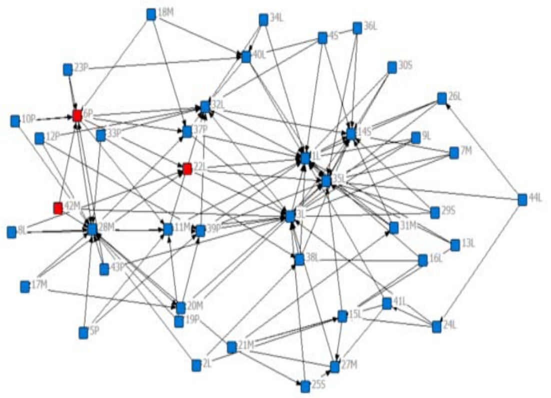

Fig. 6 . Represents the base subgraph, which is also the input graph for the isomorphic subgraph detection

The general term of pruning refers to the pruning cuts made in a certain order, so that the total number of cuts is greatly reduced. The pruning procedure used for our subgraph isomorphism algorithm is based on a concept known as graph simulation. The conditions in Ullmann's equation for checking the pruning process of the search space can be adapted to create an algorithm, as seen in the pseudocode.

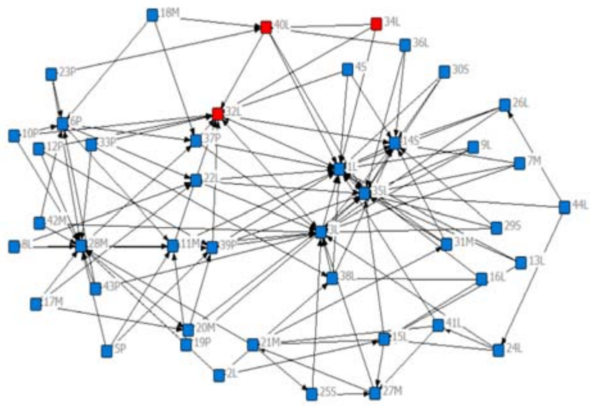

Fig. 6 Represents one of the communities (Community 1) detected using the new heuristics community detection algorithm

The community detection model is based on the notion of edge betweenness centrality that was investigated in the domain of SNA. The betweenness centrality value of an edge is defined as the number of shortest paths between the node-pairs traversing through that edge. The betweenness of edges is therefore a measurement of the flow of network activities traversing along the edges of a graph.

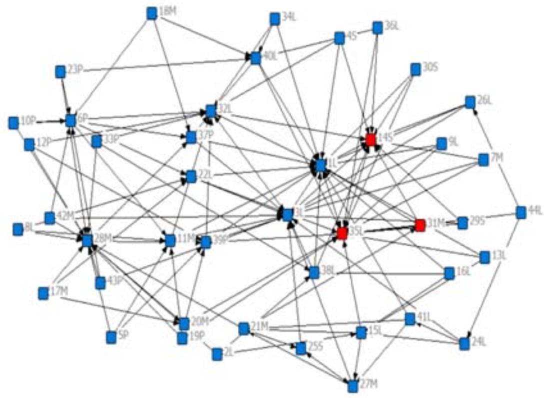

Fig. 8 Represents one of the communities (Community 2) detected using the new heuristics community detection algorithm

HAISM is effective in community detection problems and is superior to traditional community detection techniques because it is able to detect communities that the traditional algorithms miss The fact that some of the detected communities may overlap connotes that the detected communities are correct and the one that does not overlap opens up a new perspective in the domain of community detecting by the incorporation of the subgraph isomorphic matching features.

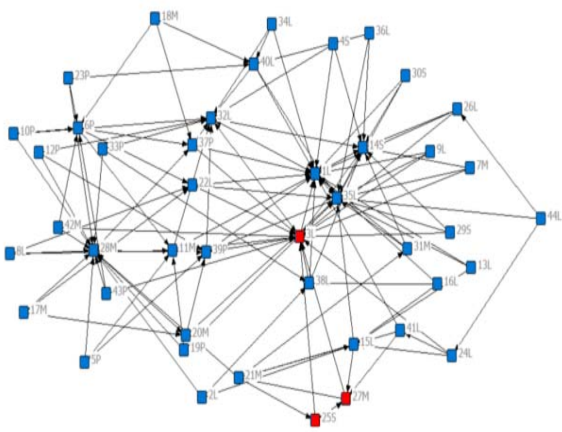

Fig. 9 Represents one of the communities (Community 3) detected using the new heuristics community detection algorithm.

Our algorithm falls into the category of the tree-search based algorithms described earlier, with the primary difference lying in the pruning algorithm, an index used for mapping each label to every vertex in G with that label, thus allowing fast initial access to feasible matches. Our algorithm begins by obtaining feasible matches: for each given query vertex u, it creates the set s(u) to contain all vertices in the data graph with the same label as u. This is the same as Ullmann's algorithm except for the fact that it ignores the degree constraint. On the whole, during the experimentation, it turned out that this constraint did not help our algorithm and that in the cases where response times were exceptionally fast, it actually slowed down the algorithm (if only by a few tens of milliseconds). This is attributable to how the pruning process rapidly eliminates many of the vertices that would have been eliminated by an initial degree constraint, nevertheless .

6. Results & Discussion

In this section we conducted an experiment on a well-studied algorithm, the Girvan-Newman community detection model, for benchmarking purposes using the same dataset. We wanted to accomplish three goals, namely to 1) compare the effectiveness of the proposed algorithm HAISM; 2) prove our hypothesis that it can detect community or communities that other algorithms miss and 3) open a new perspective by comparing the results of HAISM and traditional community detection methods, specifically the Girvan-Newman algorithm.

The Girvan-Newman community detection model is based on the notion of edge betweenness centrality that was investigated in the domain of SNA by Wasserman et al. [24]. The betweenness centrality value of an edge in a graph network is defined as the number of shortest paths between the node-pairs traversing through that edge. Therefore, the betweenness of edges is a measurement of the flow of network activities that traverse along the edges of the graph. The fundamental notion of this model is that two tightly knitted communities tend to be linked by edges where the network traffic traversed through, should be relatively high. The technique used in the Girvan-Newman's algorithm involves the analysis and detection of community structure, depending upon the iterative elimination of the edges, based on the highest number of the shortest paths that goes through them. By pruning the edges, the said network becomes divided, clustered or segmented into smaller networks, i.e. communities. The logic behind this algorithm is to find which edges in a network occur most frequently between the other pairs of nodes by finding their edges betweenness. The edges interlinking the communities are expected to have high edge betweenness. The fundamental community structure of the network will be considerably fine-grained once we eliminate the edges with high edge betweenness.

Algorithm Girvan-Newman adopted from [28]

START

(Graph: G = (N, A),
Number of Clusters: c, Edge lengths: [eij])

Begin
Calculate betweenness value of all edges in graph G;

Repeat
Identify edges of highest betweenness of all edges in G;
Remove edge (i, j) of highest betweenness value from G;

Recalculate the betweenness value of edges after the removal of edges (i, j);

Until only c clusters remain in G;
Return G connected components;

END

The edge betweenness b(i,j) for edge e(i,j):

$$b(i,j) = \sum_{i,k\neq s} \frac{\sigma ik\ e(i,j)}{\sigma ik}$$

Where σik is the total number of shortest paths from node i to node k and $\sigma ik\ e(i,j)$ is the number of those paths that pass through e(i, j) and s is the source node of the path traversal.

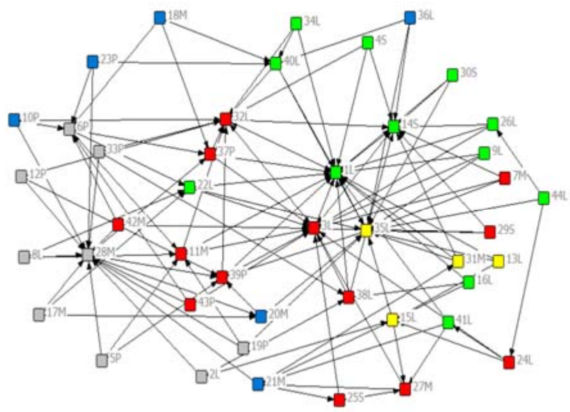

Fig. 10 The 4 communities of the Cocaine Dealing Natarajan network based on the Girvan-Newman community detection model with a search depth of 20.

Table 3: represents the 4 key communities or cliques identified

Four(4) key communities	
	Grey
	Red
	Yellow
	Green
	Blue (Outliner Nodes)

Community 1 (Figure 5), Community 2 (Figure 6) and Community 3 (Figure 7) represent some of the isomorphic subgraphs P that match the base subgraph pattern Q (Figure 4). They have been identified from our isomorphic-based community detection algorithm. (Figure 8) is the result of the use of the Girvan-Newman algorithm, utilized for benchmarking purposes against the community detected

using HAISM. As concluded in Table 2, HAISM can detect communities that overlapped with the well-studied Girvan-Newman algorithm and can additionally detect isomorphic graphs of the community that may have gone previously undetected using traditional algorithm.

7. Conclusion

In this paper, a community detection algorithm namely, HAISM has been proposed and its effective in detecting communities using isomorphic subgraph matching has been tested. The HAISM algorithm can be very practical in real-life scenario, such as in law enforcement to engage in different ways of detecting a community. We also compared it to the well-studied algorithm Girvan-Newman, with the results demonstrating that HAISM can detect communities, which overlaps with the Girvan-Newman algorithm. This verifies the detection ability of the HAISM algorithm, with the detected isomorphic graphs opening up a new perspective in the domain of detecting community by incorporating subgraph isomorphic matching feature(s).
Future works on criminal network analysis can include machine learning and deep reinforcement learning for detecting criminal communities more comprehensively and accurately, making it a valid research area.

Acknowledgment

"This work was supported by Taylor's University through its Scholarship Master Program." I would personally thank Taylor's for providing this opportunity and giving me the trust to conduct the study.
I would like to thank my supervisors, Associate Prof. Dr Azween Abdullah & Dr. Noor Zaman Jhanjhi , for their patient guidance, encouragement and continuous support provided throughout my study. I have grateful to my supervisors, who had supported me in my work, and professional growth.

References

[1] Shirinivas, S. G., Vetrivel, S., & Elango, N. M. (2010). Applications of graph theory in computer science an overview. International journal of engineering science and technology, 2(9), 4610-4621.
[2] Lyzinski, V., Fishkind, D. E., Fiori, M., Vogelstein, J. T., Priebe, C. E., & Sapiro, G. (2015). Graph matching: Relax at your own risk. IEEE transactions on pattern analysis and machine intelligence, 38(1), 60-73.
[3] Sangkaran, T., Abdullah, A., JhanJhi, N. Z., & Supramaniam, M. (2019). Survey on Isomorphic Graph Algorithms for Graph Analytics. INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND NETWORK SECURITY, 19(1), 85-92.
[4] Hopcroft, J. E., & Wong, J. K. (1974, April). Linear time algorithm for isomorphism of planar graphs (preliminary report). In Proceedings of the sixth annual ACM symposium on Theory of computing (pp. 172-184). ACM.
[5] Yang, Z., Algesheimer, R., & Tessone, C. J. (2016). A comparative analysis of community detection algorithms on artificial networks. Scientific reports, 6, 30750
[6] Calderoni, F., Brunetto, D., & Piccardi, C. (2017). Communities in criminal networks: A case study. Social Networks, 48, 116-125
[7] Wang, M., Wang, C., Yu, J. X., & Zhang, J. (2015). Community detection in social networks: an in-depth benchmarking study with a procedure-oriented framework
[8] Chen, J., & Saad, Y. (2010). Dense subgraph extraction with application to community detection. IEEE Transactions on knowledge and data engineering, 24(7), 1216-1230
[9] Mucha, P. J., Onnela, J., & Porter, M. A. (2009). Communities in networks. Notices of the American Mathematical Society, 56, 1082-1097.
[10] Lim, M., Abdullah, A., Jhanjhi, N. Z., & Supramaniam, M. (2019). Hidden Link Prediction in Criminal Networks Using the Deep Reinforcement Learning Technique. Computers, 8(1), 8.
[11] Liu, C., & Liu, Q. (2018). Community Detection Based on Differential Evolution Using Modularity Density. Information, 9(9), 218
[12] Sangkaran, T., Abdullah, A., JhanJhi, N. Z., & Supramaniam, M. (2019). Survey on isomorphic graph algorithms for graph analytics. International Journal of Computer Science and Network Security, 19(1), 85-92
[13] Ullmann, J. R. (1976). An algorithm for subgraph isomorphism. Journal of the ACM (JACM), 23(1), 31-42.
[14] 5. Cordella, L. P., Foggia, P., Sansone, C., & Vento, M. (1999, September). Performance evaluation of the VF graph matching algorithm. In Proceedings 10th International Conference on Image Analysis and Processing (pp. 1172-1177). IEEE.
[15] 6. Cordella, L. P., Foggia, P., Sansone, C., & Vento, M. (2004). A (sub) graph isomorphism algorithm for matching large graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(10), 1367-1372.
[16] 7. Schmidt, D. C., & Druffel, L. E. (1976). A fast backtracking algorithm to test directed graphs for isomorphism using distance matrices. Journal of the ACM (JACM), 23(3), 433-445.
[17] 8. Corneil, D. G., & Gotlieb, C. C. (1970). An efficient algorithm for graph isomorphism. Journal of the ACM (JACM), 17(1), 51-64.
[18] 9. Fu, L., & Chandra, S. (2012). Optimized backtracking for subgraph isomorphism. International Journal of Database Management Systems, 4(6), 1.
[19] Javed, M. A., Younis, M. S., Latif, S., Qadir, J., & Baig, A. (2018). Community detection in networks: A multidisciplinary review. Journal of Network and Computer Applications, 108, 87-111.
[20] 11. Li, J., Lear, M. J., Kawamoto, Y., Umemiya, S., Wong, A. R., Kwon, E., ... & Hayashi, Y. (2015). Oxidative amidation of nitroalkanes with amine nucleophiles using molecular oxygen and iodine. Angewandte Chemie International Edition, 54(44), 12986-12990.
[21] 12. Newman, M. E. (2016). Community detection in networks: Modularity optimization and maximum likelihood are equivalent. arXiv preprint arXiv:1606.02319.
[22] Peixoto, T. P. (2013). Parsimonious module inference in large networks. Physical Review Letters, 110(14), 148701

[23] Cordella, L. P., Foggia, P., Sansone, C., & Vento, M. (2004). A (sub) graph isomorphism algorithm for matching large graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(10), 1367-1372.
[24] Carletti, V., Foggia, P., Saggese, A., & Vento, M. (2017). Challenging the time complexity of exact subgraph isomorphism for huge and dense graphs with VF3. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(4), 804-818
[25] Omidvar, M. N., Li, X., Mei, Y., & Yao, X. (2013). Cooperative co-evolution with differential grouping for large scale optimization. IEEE Transactions on Evolutionary Computation, 18(3), 378-393.
[26] Huang, Q., White, T., Jia, G., Musolesi, M., Turan, N., Tang, K., ... & Yao, X. (2012, September). Community detection using cooperative co-evolutionary differential evolution. In International Conference on Parallel Problem Solving from Nature (pp. 235-244). Springer, Berlin, Heidelberg.
[27] McKay, B. D., & Piperno, A. (2014). Practical graph isomorphism, II. Journal of Symbolic Computation, 60, 94-112.
[28] Krzakala, F., Moore, C., Mossel, E., Neeman, J., Sly, A., Zdeborová, L., & Zhang, P. (2013). Spectral redemption in clustering sparse networks. Proceedings of the National Academy of Sciences, 110(52), 20935-20940

Theyvaa Sangkaran received his Bachelor's in Computer Security from UNITAR International University a in 2016. Currently, he is pursuing his Master of Computer Science at Taylor's University, Malaysia.

Azween Abdullah is a professional development alumni of Stanford University and MIT and his work experience includes thirty years as an academic in institutions of higher learning and as director of research and academic affairs at two institutions of higher learning, vice-president for educational consultancy services, fifteen years in commercial companies as Software Engineer, Systems Analyst and as a computer software developer and IT/MIS consultancy and training.

NZ Jhanjhi has completed his PhD. in IT from University Technology Petronas (UTP) Malaysia. He has 18 years of teaching and administrative experience internationally. He has an intensive background of academic quality accreditation in higher education besides scientific research activities, he had worked for academic accreditation for more than a decade and earned ABET accreditation twice for three programs at College of computer sciences and IT, King Faisal University Saudi Arabia.