Threat Identification and Examination using Graph Based Anomaly Detection

Soumok Dutta, Parvathi.R, Ganesan.R

Abstract*: The aim of this paper is to investigate the Graph Based Anomaly Detection (GBAD) systems to find anomalies or features in a graph that are inconsistent with the general or maximal substructures of the graph. A substructure miner approach was implemented. The Frequent Substructure Miner (FSM) was adopted to find the optimal substructure, which was then used to compare the normal GBAD and Minimum Description Length (MDL) approach that has been in use. The FSM approach uses graphs of size 10, 100 and 1000 nodes to determine the resulting efficiency and hence the runtime as well. The runtime determines how long the two systems require to find anomalies in each type of graph.

Keywords* : GBAD-FSM, GBAD-MDL, SUBDUE, GBAD, Minimum Descriptive Length, Probabilistic, MPS, Runtime Efficiency, Anomaly detection, Graph based, Insider threat , Cyber Crime.

I. INTRODUCTION

One of the major issues of the Cyber Crime industry is the insider approach, wherein a infiltrator with usage access in the system exudes information from inside the company to the outside parties from within the system. The task of mining complex data holding multiple attributes is important for several aspects, including that of detecting insider threats in an organization. GBAD system allows to achieve that using one of the two approaches used for implementing it. The humungous amount of data generated today from internet resources such as social networks, banking systems, or any other complex system is an example of the dataset that need to determine if any insider threats lurk within them. An attack from inside is extremely difficult to trace from an approach that is meant to detect only external intrusions and anomalous instances. Herein comes the GBAD or Graph based anomaly detection approach that aims at finding the threats inside the system using a pattern based search algorithm. This paper aims at understanding the idea of Graph based Anomaly Detection [5, 12]. The system has been implemented as such to comparatively analyze the two approaches of the Graph based anomaly detection system, the Minimum descriptive Length or the MDL approach and the Frequent Substructure Miner or the FSM approach. A study of the runtimes of both the systems, the size of the best found substructure and the no. of anomalous instances is done to find the best possible result of suspicious activities in the same.The earlier approaches where statistical or visual to mine and monitor illegal access. But as the complexity of data has risen with more interconnected and related data,so has the need to mine it for the relatedness that it possesses with respect to the other attributes of a network or a dataset,which has led to further studies into the much unexplored Graph based anomaly detection systems approach.

Revised Manuscript Received on October 30, 2019.
***** Correspondence Author
Soumok Dutta, School of Computing Science and Engineering, Vellore Institute of Technology, Chennai, Tamil Nadu, India.
Parvathi.R, School of Computing Science and Engineering, Vellore Institute of Technology, Chennai, Tamil Nadu, India.
Ganesan.R, School of Computing Science and Engineering, Vellore Institute of Technology, Chennai, Tamil Nadu, India.

II. BACKGROUND OF GRAPH BASED ANOMALIES

2.1 The idea behind graph based anomalies

The idea behind the approach is to find anomalies in graph-based data where anomalous patterns exist in the structured graph that has the non-anomalous pattern also referred to as normative pattern. The graph classification of an anomaly is unique for any such anomaly detection techniques, nevertheless it may be graph based or not.

Unlike the other approaches that determine the anomaly or in this case the bad or unrelated pattern, this approach determines the good pattern or the non- anomalous substructure that is then used to extract the illicit parts of the graph portrayed dataset. A graph substructure S' is anomalous if it is not isomorphic to the graph's normative substructure S, but is isomorphic to S within X%. [1, 8]

2.2. Category of Anomalies

The GBAD model in its presence entirety covers 3 types of anomalies: insertions, modifications and deletions [6].

Insertions anomalies are situations when upon checking the module finds the existence of a node in the network which has either been recently inserted or not recognized in the previous searches. Insertion anomalies in terms of insider threat can be hackers from external networks trying to get inside the network of an organization. Insertion anomalies are thus one of the most important types of anomalies.

Modification anomaly is the second most important type of anomalies, especially for insider threat security [4,9]. A situation where a particular node has gone through some modification such as disconnection from other nodes, editions in the label of the edges etc. Such a situation is very harmful for an organization with a huge network of computers because it is very hard to monitor the exact connections each of the nodes and a modification in these connections can cause huge losses as well [14]. Thus, using the GBAD module we can easily detect such modifications.

Deletion anomaly is the least important among the three anomalies mentioned here, but it can still cause a lot of damage to an organization. A deletion anomaly is basically when an already existing node is removed from the network. For an organization, every node present in its entire network stores some data or the other which is integral to the working of the organization. Thus it is

very important for the organization to immediately track the nodes that have been deleted so that the data can be recovered.

III. GRAPH BASED ANOMALY DETECTION

GBAD is a unsupervised approach [3,10], based on the SUBDUE graph based knowledge discovery method. The SUBDUE method basically, recognizes normative patterns that exist in the network. It uses a greedy beam search and Minimum Description length. SUBDUE is used to find the best substructure which is then given to MDL to find the anomalies. For modification anomalies, the GBAD MDL module is used which finds the best substructure that exists and sticks to it, thus making sure any new insertion is immediately detected. For insertion anomalies, the GBAD Probability module is used, which does not go for the similar patterns but check the extensions that take place. For deletion anomalies, the GBAD MPS module or the maximum partial substructure module is used, which first discovers the normative pattern and the checks for all the ancestral substructures that have some edges missing[7,13].

IV. THE RESULT AND DISCUSSION

4.1. Sampling and Execution Methods

The research sampling method that is used in this study is a size-based one - the sample graphs used are those of an increasing order from 10 to 1000 nodes. A few random samples were utilized. Both the systems were given the same parameters for implementing the algorithm and the same graph input file to find the relation between graph size and execution time for the same graphs.

This relation determines the general trend of how the FSM system works with the increasing size of the input graph.

4.2. GBAD-FSM Based Execution Procedure

The proposed method is to implement a substructure mining method instead of the previously implemented method of GBAD systems. One graph-based knowledge discovery approach that has shown to be expendable within the system configurations without losing any accuracy is the frequent subgraph miners. The possibility that the most effective approaches are the ones that convert graphs into a string in canonical form and then perform a canonical-string based graph match is high, as that is theoretically expected of it.

To verify the potential effectiveness of implementing anomaly detection algorithms to a frequent subgraph mining approach[2,11], GBAD algorithms are implemented into an Apriori algorithm based approach where the prior knowledge of frequent item-set properties are used to discover the substructures that are frequent, and called this new approach GBAD-FSM. Such a framework implementation for the Apriori based algorithm is the GASTON Framework. This well-known property provides a reduction in the search space, which can then be used to improve the performance for determining which substructures have an anomalous match.

4.3. Console Execution

The console execution of the system working on a graph with 1000 nodes and 903 edges is explained here. The parameters of MDL are set as 0.2 and the MST is set to 1 for keeping the minimum frequency to 1 as well . The Anomaly Detection method is Information Theoretic. Since the phase parameter of FSM is not mentioned, by default the system considers to execute both the phases that is it does both substructure mining in figure 2 to find the best structure and then detect the anomalous substructure and its anomalies as well. Then the results are compiled into a file and saved with the result data as Best_Sub.g and Anom_Sub.g in Table 1 and Table 2. The runtime of this instance is also printed as the process finishes in the console itself in Figure 2 shows the graph based anomaly detection

The Console Command: graph_fsm_2.1/gbad-fsm -prob 2 -mst 1 -graph graphs/run1000_mdl.g

The output generated in the console first describes or depicts the type of anomaly detection technique that is being used, as in the above mentioned command, the approach is probabilistic, with the iterations specified as 2 and the MST or minimum support threshold of 1 that is the minimum frequency, the graph attribute is set as run_1000_mdl.g which is the graph with the substructure with 1000 nodes in the structure. The best substructure is saved in the file best_sub.g and the anomalous substructure is saved in the file anom_sub.g after the execution phases of the system are completed.

The path processing is then iterated 2 times to find the probabilistic pattern in the graph. Then the frequent cyclic graphs are created based on the normative pattern that is generated during the processing the path.

After all the substructures are generated and the general anomalous value is calculated from the processing of the cyclic graphs (in case of the run1000_mdl.g, the anomalous value turns out to be 0.045455) using the average pattern deviation detected by the mining the substructures to find the general pattern. This anomalous value is then used to determine the anomalous substructures in the system. The resultant substructures are saved in the file anom_sub.g.

Similarly the best substructures are then found, that is the one with the least anomalous values which is the closest to the least deviation from the generic pattern of the system. The generated substructures as a result is then saved in the file best_sub.g.

Furthermore, since in the given case the user has not specified the phase of the execution of the system, it runs both the phases of creating the best substructure and finding the anomalous substructure. Also the GBAD system allows the user to specify an instance file where the normative pattern is already specified it runs the first phase to find the best substructure. It processes the paths for doing the same, by default the no of paths to be processed is 10. The 10 cycles of FSM determine the number of cyclic graphs. The number of real trees and the number of paths are used to generate the total number of cyclic graphs, total number of real trees and total number of paths. The runtime of the frequent substructure mining from cyclic graphs to generate the final anomalous score that is then used to classify the substructures that are anomalous and those that are non-anomalous is also displayed along with the total runtime that includes the time to classify the substructure. Along with the runtime, the number of instances of the best substructures and size of the best substructure also displayed. This info of the best substructure is used to find the anomalous substructures that are in fact the ones that the most distant from the best substructures.

4.4. Graphical Execution

The GUI based output is one that determines the general structure of data given, it converts the data into graphs of nodes and links in shown in Figure 3. The GUI allows the user to select the parameters of GBAD system and find the anomalous instances according to the thresholds provided by the user. In this case the anomaly detection method is MDL and the threshold for the same has been set to 0.2

The Figure 1 shows the visual output of the system is the one that shows the anomaly or the anomalous substructure with a color based objective. In the example used a red color represents the anomalous node and yellow color represents the anomalous link is used. The GUI based approach based is much easier to determine the anomalous edges and vertices in the graph, and is much easier on the eyes.

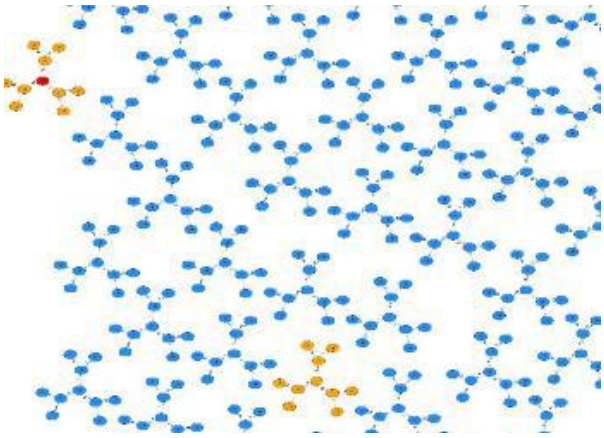

Figure 1. Anomaly , Anomalous Substructure Detection

4.4. Comparison Of Result

Table 3 describes the runtime comparison of different algorithms with respect to number of nodes with GBAD-MDL and GBAD-FSM. The FSM based algorithm approach is definitively faster than SUBDUE based algorithmic implementation at least in probabilistic anomalies, but in the larger size graphs both the systems are almost equally efficient in finding the anomalies.

Table -1 Runtime of systems (in seconds)

Algorithm	10 Nodes		10 Nodes		10 Nodes	
	GBAD MDL	GBAD FSM	GBAD MDL	GBAD FSM	GBAD MDL	GBAD FSM
MDL	3.07	2.296	0.01	41.786	0.08	0.0042
PROB	0.19	0.0013	0.06	0.012	0.14	0.1454
MPS	0.24	2.1969	0.02	40.466	0.14	0.0059

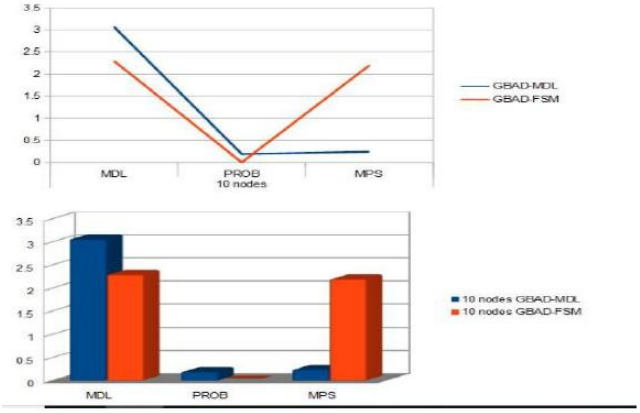

Figure 2: 10 Node comparison of GBAD-MDL and GBAD-FSM

The Figure 2 depicts how a SUBDUE based approach works compared to a FSM Based approach on a 10 node substructure graph. It compares the two techniques on their runtime on the three different parameters available on the GBAD system , that is MDL , PROB and MPS which detect the anomalous insertion, modification and deletion of nodes or edges in the graph.

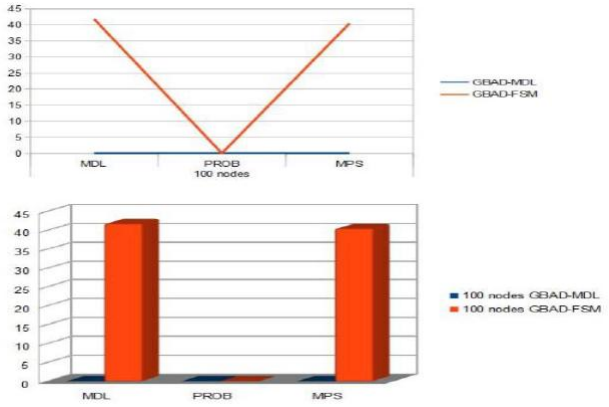

Figure 3. 100 Node comparison of GBAD-MDL and GBAD-FSM

The Figure 3 shows the 100 Node comparison of GBAD-MDL and GBAD-FSM is visualized in the aforementioned graph, with data comparing runtime in MDL, PROB and MPS for the 100 node graph with anomalies. The Figure 4 shows the 1000 Node comparison of GBAD-MDL and GBAD-FSM is visualized in the aforementioned graph,with data comparing runtime in MDL , PROB and MPS for the 1000 node graph with anomalies The line graph conveys the same story as the bar chart but more precisely with an exact depiction of the corresponding runtimes.

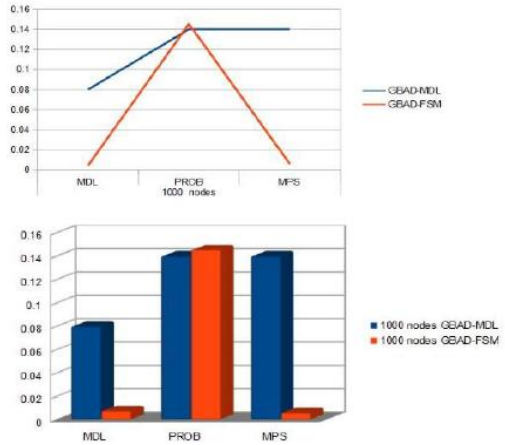

Figure 4. 1000 Node comparison of GBAD-MDL and GBAD-FSM

V. THE RESULT AND ITS INTERPRETATION

The graphs in figure 2,3, and 4 present that the FSM based algorithm approach is definitively faster than SUBDUE based algorithmic implementation at least in probabilistic anomalies,

but in the larger size nodes graphs both the systems are almost equally efficient in finding the anomalies. And in lesser nodes groups, the MDL and PROB algorithms have faster implementation in FSM system than the SUBDUE based system, the study also indicates that the SUBDUE is much faster compared to FSM for graphs with number of nodes in the order of 10 raised to power of 2. But in larger node graphs, such as ones with the number of nodes in the order of 10 raised to power of 3. The results are quite opposite where the MDL and MPS algorithmic implementation in FSM is much faster than that in SUBDUE based one. The authors have measured GBAD's accuracy by trying to identify the intended anomalies against the reported anomalies that were nonexistent and classified as false positives. The overall results indicated that GBAD showed a success rate of more than 95%, by detecting almost all the anomalies most of the time, with minimal or no false positives cases, in all the runtimes. It was indicative that the larger the graph, or the count of sub graphs analyzed for any anomalous structure, the greater was the runtime for the implementation of the algorithms. In general, the runtime of GBAD is of the form of polynomial relative to the size of the graph and varying as the parameters of the algorithm.

VI. CONCLUSION

The System's ability to detect the best subgraph and the anomalies is limited by the resources allocated to it. In a graph where the anomalous substructure has the minimal deviation from the normative pattern, that is it looks almost similar to the original best substructure to a great extent, given a sufficient amount of processing time and memory, it is assumed that the three algorithms will detect the anomalous substructure with a 100 % efficiency and thus no false positives. However, the amount of noise present in the graph also affects the discovery of anomalous substructure which creates the problem that if the noise in the graph generates a smaller deviation from the normative pattern compared to the actual anomalous substructure, it will create a scenario where it scores higher than the targeted anomaly. The results of the Graphs and the table indicate that in very small graphs and moderately large graphs, FSM based algorithmic Approach is equal or faster compared to SUBDUE based one.

The development of the GBAD system can be implemented to two or more algorithms together to determine 2 kinds of anomalies that cannot be determined by the current system. In the current system, if a deletion decreases the number of nodes and a same number of anomalous insertions are done, then the system might not detect any anomalies. Combining the two algorithms will solve this problem. Why the MPS algorithm reports the non-anomalous deviation which allows it to give false positive is a focus of future work. One way to address this issue is to handle the data as a stream, thereby building the graph as data "comes in", searching for normative patterns and anomalies. In addition, we will continue to analyze the effectiveness of this approach using data sources that are integrated together.

REFERENCES

1. Akoglu. L, Tong. H, Koutra. D 2014, Graph-based Anomaly Detection and Description: A Survey.
2. C.Jiang,F.Coenen, M.Zito,"A Survey of frequent sub graph mining algorithms", Knowledge Engineering Review, 2012
3. Eberle, W., Holder, L. B., &Massengill, B. (2012, May). Graph-Based Anomaly Detection Applied to Homeland Security Cargo Screening. In FLAIRS Conference.
4. Eberle, W., Graves, J., & Holder, L. (2010).Insider threat detection using a graph-based approach. Journal of Applied Security Research, 6(1), 32-81
5. Eberle, W., & Holder, L. (2013, December). Incremental Anomaly Detection in Graphs. In Data Mining Workshops (ICDMW), 2013 IEEE 13th International Conference on (pp. 521-528).
6. Mukherjee, M. and Holder, L. Graph-based Data Mining on Social Networks. Workshop on Link Analysis and Group Detection, KDD, 2004.
7. Noble, C. C., & Cook, D. J. (2003, August).Graph-based anomaly detection. In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 631-636).
8. Padmanabhan.K, Chen.Z, Sriram.L, Ramaswamy.S and Bryan Thomas.R. 2014. Graph Based Anomaly Detection in:Practical Graph Mining with R, CRC Press, Taylor and Francis Group,312367.
9. Rattigan, M. J., & Jensen, D. (2005). The case for anomalous link discovery. Acm Sigkdd Explorations Newsletter, 7(2), 41-47.
10. Shetty, J., & Adibi, J. (2005, August). Discovering important nodes through graph entropy the case of enron email database. In Proceedings of the 3rd international workshop on Link discovery (pp. 74-81). ACM.
11. Staniford-Chen, S., Cheung, S., Crawford, R., Dilger, M., Frank, J., Hoagland, J., ... & Zerkle, D. (1996, October). GrIDS-a graph based intrusion detection system for large networks. In Proceedings of the 19th national information systems security conference (Vol. 1, pp. 361-370).
12. Sun, J., Qu, H., Chakrabarti, D., & Faloutsos, C. (2005). Relevance search and anomaly detection in bipartite graphs.ACM SIGKDD Explorations Newsletter, 7(2), 48-55.
13. Thomas, L. T., Valluri, S. R., & Karlapalem, K. (2010). Margin: Maximal frequent subgraph mining. ACM Transactions on Knowledge Discovery from Data (TKDD), 4(3), 10.
14. Yan, X., & Han, J. (2002, December). gspan: Graph-based substructure pattern mining. In 2002 IEEE International Conference on Data Mining, 2002. Proceedings. (pp. 721-724). IEEE.