Detecting Insider Threats Using a Graph-Based Approach

William Eberle, *Tennessee Tech University*, and Lawrence Holder, *Washington State University*

Abstract – This paper presents the use of graph-based approaches to discovering anomalous instances of structural patterns in data that represent insider threat activity. The approaches presented search for activities that appear to match normal transactions, but in fact are structurally different. We show the usefulness of applying graph theoretic approaches to discovering suspicious insider activity in domains such as e-mail correspondences, business processes, and cybercrime. The paper then concludes with some future research that deals with the handling of dynamic graphs, as well as the implementation of these graph-based anomaly detection algorithms in a frequent subgraph miner.

Index terms - anomaly detection, graph-based, insider threat

I. INTRODUCTION

The ability to mine data for nefarious behavior is difficult due to the *mimicry* of the perpetrator. If a person or entity is attempting to participate in some sort of illegal activity, they will attempt to convey their actions as close to legitimate actions as possible. Recent reports have indicated that approximately 6% of revenues are lost due to fraud, and almost 60% of those fraud cases involve employees [1]. The Identity Theft Resource Center recently reported that 16.9 percent of the security breaches in 2009 came from insiders, up from 15.8 percent in 2008 [2]. Various insider activities have threatened our nation's security, such as violations of system security policy by an authorized user, deliberate and intended actions such as malicious exploitation, theft, or destruction of data, the compromise of networks, communications, or other IT resources, and the difficulty in differentiating suspected malicious behavior from normal behavior. Organizations responsible for the protection of their company's valuable resources require the ability to mine and detect internal transactions for possible insider threats. Yet, most organizations spend considerable resources protecting their networks and information from the outside world, with little effort being applied to the threats from within.

Cybercrime is one of the leading threats to company confidential data and resources. A recent study by the Ponemon Institute surveyed 577 IT practitioners, who rated the issue of cybercrime as the top trend in their industry for the next few years, over such hot topics as cloud computing, mobile devices, and peer-to-peer sharing [3]. The U.S. Department of Justice, in its Computer Crime & Intellectual Property Section, reported nine incidences in the last month alone (June 2010), ranging from hacking into a wireless network to counterfeiting [4]. News stories detail how insiders have bilked corporations out of millions due to their ability to access sensitive information – sometimes after they have resigned from a company that did not immediately remove their confidential access [5]. There have even been studies that suggest that the economy has impacted, or will impact, the surge in cybercrime [6][7].

For the last several years, companies have been analyzing their IT operations and processes for the purpose of uncovering insider threats and cybercrime. Most approaches have been either statistical in nature, leading to various data mining approaches, or a visualization of their resources where they can monitor for illegal access or entry. However, recently, the ability to mine relational data has become important for detecting *structural* patterns. The complex nature of heterogeneous data sets, such as network activity, e-mail, payroll and employee information, provides for a rich set of potentially interconnected and related data. A potential avenue for detecting threats from insiders in structurally complex data can be found in g*raph-based anomaly detection*.

II. RELATED WORK

Much of the information related to insider threats resides in the *relationships* among the various entities involved in an incident. Recently there has been an impetus towards analyzing multi-relational data using graph theoretic methods. Not to be confused with the mechanisms for analyzing "spatial" data, graph-based data mining approaches are an attempt at analyzing data that can be represented as a graph (i.e., vertices and edges).

In 2003, Noble and Cook used the SUBDUE system to look at the problem of anomaly detection from both the anomalous substructure and anomalous subgraph perspective [8]. They were able to provide measurements of anomalous behavior as it applied to graphs from two different perspectives. *Anomalous substructure* detection dealt with the unusual substructures that were found in an entire graph. In order to distinguish an anomalous substructure from the other substructures, they created a simple measurement whereby the value associated with a substructure indicated a degree of anomaly. They also presented the idea of *anomalous subgraph* detection which dealt with how anomalous a subgraph (i.e., a substructure that is part of a larger graph) was to other subgraphs. The idea was that subgraphs that contained many common substructures were

William Eberle is with Tennessee Tech University (e-mail: weberle@tntech.edu).

Lawrence Holder is with Washington State University (e-mail: holder@wsu.edu).

generally less anomalous than subgraphs that contained few common substructures.

Several approaches employ statistical measures to identify individual node or edge anomalies. Lin and Chalupsky [9] took the approach of applying what they called rarity measurements to the discovery of unusual links within a graph. The AutoPart system presented a non-parametric approach to finding outliers in graph-based data [10]. Part of this approach was to look for outliers by analyzing how edges that were removed from the overall structure affected the minimum descriptive length (MDL) of the graph [11]. The idea of entropy was used by Shetty and Adibi in their analysis of the famous Enron e-mail data set [12]. Using bipartite graphs, Sun et al. [13] presented a model for scoring the normality of nodes as they relate to other nodes. Rattigan and Jensen went after anomalous links using a statistical approach [14].

In Priebe et al.'s work, they used what are called "scan statistics" on a graph of the e-mail data that is represented as a time series [15]. While their approach detects statistically significant events (excessive activity), without further analysis, they are unable to determine whether the events are relevant (like insider trading). Martin et al. examined what they called "behavioral features" of a particular user's network traffic in order to discover abnormal activity [16]. Through various clustering approaches, and comparisons to methods such as Support Vector Machines and Naives Bayes Classification, they group sets of users into single behavioral models. Wang et al. used attack graphs to measure and quantify potential network attacks [37]. In their model, an attack graph is a directed graph representing prior knowledge about vulnerability, where the harder it is to attack some vulnerability, the less likely that path will be taken by an insider. Diesner et al. applied various network analytic techniques in their exploration of the structural properties of the Enron network. They used various graph structural metrics, such as betweenness centrality, eigenvectors and total degree in order to identify key players across time [17]. In 2007, Kurcz et al. used hierarchical spectral clustering to evaluate weighted call graphs [18]. They analyzed several heuristic approaches using phone calls made over an eight-month period. In Swayne et al.'s work, they used graph techniques to explore AT&T phone records [19]. While their approach was able to provide for the analysis of phone traffic, it was entirely based upon graph visualization, rather than any graph theoretic approaches. In fact, when it comes to generating graphs of information, much research has dealt with only the visual aspects of what is represented, rather than the structural aspects of the graphs themselves.

The advantage of graph-based anomaly detection is that the relationships between elements can be analyzed, as opposed to just the data values themselves, for structural oddities in what could be a complex, rich set of information.

III. GRAPH-BASED ANOMALY DETECTION (GBAD)

The idea behind the approach used in our work is to find anomalies in graph-based data where the anomalous substructure in a graph is part of (or attached to or missing from) a *normative pattern,* which in our implementation is a substructure that minimizes the description length (MDL) of a graph.

Definition: *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%.*

X signifies the percentage of vertices and edges that would need to be changed in order for *S'* to be isomorphic to *S*. The importance of this definition lies in its relationship to any deceptive practices that are intended to illegally obtain or hide information. The United Nations Office on Drugs and Crime states the first fundamental law of money laundering as "The more successful money-laundering apparatus is in imitating the patterns and behavior of legitimate transactions, the less the likelihood of it being exposed" [20].

GBAD (Graph-based Anomaly Detection) is an *unsupervised* approach, based upon the SUBDUE graph-based knowledge discovery method [21]. Using a greedy beam search and MDL heuristic, each of the three anomaly detection algorithms in GBAD uses SUBDUE to find the best substructure, or normative pattern, in an input graph. In our implementation, the MDL approach is used to determine the best substructure(s) as the one that minimizes the following:

$$M(S,G) = DL(G \mid S) + DL(S)$$

where *G* is the entire graph, *S* is the substructure, *DL(G|S)* is the description length of *G* after compressing it using *S*, and *DL(S)* is the description length of the substructure.

There are three general *categories of anomalies*: insertions, modifications and deletions. Insertions would constitute the presence of an unexpected vertex or edge. Modifications would consist of an unexpected label on a vertex or edge. Deletions would constitute the unexpected absence of a vertex or edge. We have developed three separate algorithms: GBAD-MDL, GBAD-P and GBAD-MPS. Each of these approaches is intended to discover one of the corresponding possible graph-based anomaly categories as set forth earlier. The reader should refer to Eberle and Holder's work for a more detailed description of the actual algorithms [22].

IV. EXPERIMENTS WITH GBAD

A. *E-mail Correspondences*

One of the more recent domains that have become publicly available is the data set of e-mails between employees from the Enron Corporation. In addition to Deisner et al.'s work

[17] and Shetty and Adibi's work [12], both of which were mentioned in the previous section, others have attempted to analyze this data set using graph-based approaches. Wan et al. use a link-based event detection method that clusters similar vertices together and then considers deviations from each vertex's individual profile [38]. Whereas, Huang uses probabilities to generate models that predict the likelihood of links with the topology of a graph [39]. Also, Akoglu et al. present an algorithm called OddBall that searches weighted graphs based upon a set of rules to determine whether or not an anomaly exists [40].

The Enron e-mail dataset consists of not only messages, but also employee information such as their full name and work title. By limiting our graph to the Enron employees and their correspondences, we are able to not only create a "social network", but also discover anomalous behaviors among *classes* of individuals. Thus, we generated graphs based upon the social aspect and company position of employees that start a "chain" of e-mails, where a chain consists of the originating e-mail and any subsequent replies or forwards to that corresponding e-mail. Each graph consists of the substructures shown in Figure 1.

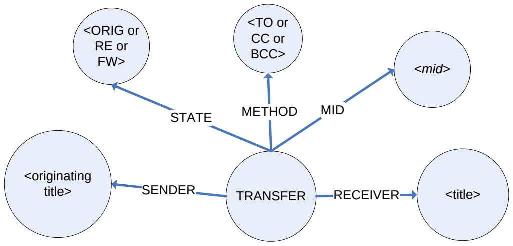

Figure 1. Graph substructure of e-mail data set.

In this representation, a graph consists of individual, disconnected substructures that represent the "flow" of each e-mail that originates from someone with a specified employment title (e.g., Director). An e-mail can be sent by one or more TRANSFERs to one or more individuals with varying employment titles (represented by a directional arrow to show who sent the message to whom), and can either be sent back (as a reply or forward) to the <originating title>, or forwarded/replied on to other <title> entities. There is no limit to the number of times a message can be replied/forwarded.

There are many different employee titles within Enron (i.e., Managers, Directors, CEOs, etc.), and each of the GBAD algorithms were able to show different structural anomalies in the chains of e-mails that originated along people's company titles. For instance, running GBAD on the graph that consists of e-mails originating from Directors, the anomalous instance shown in Figure 2 is discovered.

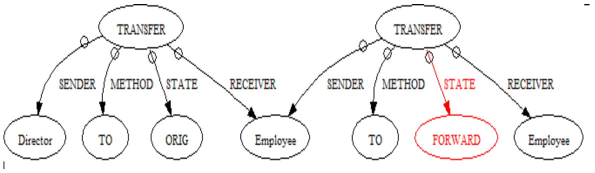

Figure 2. Anomalous instance (portion) of e-mail being forwarded.

This anomalous instance consists of a message being sent from a Director to an Employee (i.e., non-management personnel), that was then forwarded to another non-management Employee. What is interesting about this anomaly is that the data set consists of many e-mails that are sent "TO" "Employee"s from "Director"s, but this is the only situation where the Employee FORWARDed the e-mail onto another "Employee", who was not privy to the original e-mail. Specifically, the e-mail started with Hyatt (director) regarding "Oasis Dairy Farms Judgement", who sent it to Watson (employee), who then forwarded it to Blair (employee). In addition, applying GBAD to the graph of e-mails originating from personnel with the title of "Trader" produces two anomalous instances. In the first anomalous instance, from an e-mail entitled "Financial Disclosure of $1.2 Billion Equity Adjustment", out of only four e-mails sent to a CEO, this was the only example of an e-mail being sent TO a CEO - the other 3 e-mails are CCed to the CEO. In the case of the second anomalous instance, an e-mail entitled "Fastow Rumor", this was the only time that an e-mail was sent by a Trader to a President.

B. Business Processes

In order to demonstrate the potential effectiveness of GBAD for detecting insider threats in business processes, we simulated a passport processing scenario that was motivated by two real-world sources of information. One source is the incidents reported in the CERT Insider Threat documents [23][24][25] that involve privacy violations in a government identification card processing organization and fraud in an insurance claim processing organization. The other model we used is based on the process flow associated with a passport application [26]. The outline of this process flow, depicted in Figure 3, is as follows:

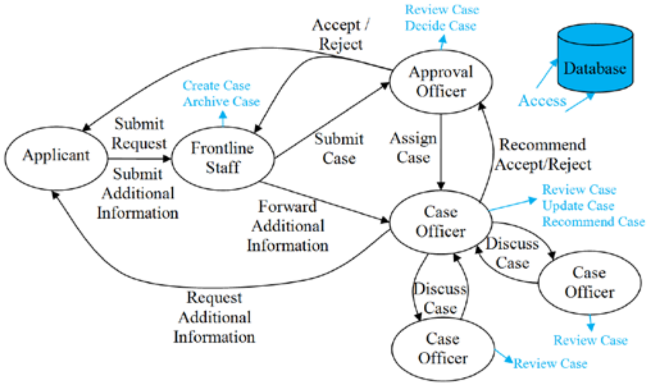

Figure 3. Information flow in application scenario.

1. The applicant submits a request to the frontline staff of the organization.
2. The frontline staff creates a case in the organization's database and then submits the case to the approval officer.
3. The approval officer reviews the case in the database and then assigns the case to one of the case officers. By default, there are three case officers in this organization.
4. The assigned case officer reviews the case. The assigned case officer may request additional information from the applicant, which is submitted to the frontline staff and then forwarded to the assigned case officer. The assigned case officer updates the case in the database based on this new information. The assigned case officer may also discuss the case with one or more of the other case officers, who may review the case in the database in order to comment on the case. Ultimately, the assigned case officer will recommend to accept or reject the case. This recommendation is recorded in the database and sent to the approval officer.
5. Upon receiving the recommendation from the assigned case officer, the approval officer will make a final decision to accept or reject the case. This decision is recorded in the database and sent to both the frontline staff and the applicant.
6. Finally, upon receiving the final decision, the frontline staff archives the case in the database.

There are several scenarios where potential insider threat anomalies might occur, including:

1. Frontline staff performing a Review Case on the database (e.g., invasion of privacy).
2. Frontline staff submits case directly to a case officer (bypassing the approval officer).
3. Frontline staff recommends or decides case.
4. Approval officer overrides accept/reject recommendation from assigned case officer.
5. Unassigned case officer updates or recommends case.
6. Applicant communicates with the approval officer or a case officer.
7. Unassigned case officer communicates with applicant.
8. Database access from an external source or after hours.

Representing the processing of 1,000 passport applications, we generated a graph of approximately 5,000 vertices and 13,000 edges, and proceeded to replicate the scenarios described above.

For scenarios 1, 3 and 6, while the GBAD-MDL and GBAD-MPS algorithms do not discover any anomalous structures, GBAD-P is able to successfully discover the single anomalous cases out of 1,000 where staff is violating the process. For scenario 2, the GBAD-MPS algorithm successfully discovers all three instances where the frontline staffer did not submit the case to the approval officer.

For Scenario 4, we randomly modified three examples by changing the recommendation that the "CaseOfficer" sends to the "ApprovalOfficer". This scenario tests GBAD's ability to handle *multiple normative patterns*. Potentially, there are two types of prevalent patterns in this type of data: (1) The ApprovalOfficer and CaseOfficer both accept a passport application, and (2) The ApprovalOfficer and CaseOfficer both reject an application. Therefore, potentially anomalous scenarios could exist where the ApprovalOfficer overrides the accept/reject recommendation from the assigned CaseOfficer. We generated a graph consisting of these two normative patterns, although these patterns were not among the top-ranked most normative substructures. We then randomly inserted an anomalous instance of the first type (case officer accepts, approval officer rejects) and two anomalous instances of the second type (case officer rejects, approval officer accepts). Configuring the GBAD-P algorithm to analyze the top *N* normative patterns, where *N* is set arbitrarily to 20, all three anomalous examples are reported as the most anomalous. Other experiments showed that the size of *N* was not important. For instance, in this example, when we increase *N* to 100, the top three anomalies reported are still the same ones. In addition, no other substructures are reported as anomalous along with these top three anomalies (i.e., no false positives).

For scenario 5, we randomly inserted into two examples the situation where a "CaseOfficer" recommends to accept a case for which they were not assigned. In this scenario, GBAD-MDL does not report any anomalies, while both GBAD-MPS and GBAD-P each discover both anomalous instances. GBAD-MPS discovers the anomalies because the "CaseOfficer" has assigned himself to the case without any corresponding recommendation back to the "ApprovalOfficer" or "Database", while GBAD-P uncovers the extra "CaseOfficer" and his unauthorized assignment to the case. Figure 4 shows the normative pattern and the anomalous structures from one of these examples. Also, while not shown, this same structural anomaly can be found in scenario 7. Scenario 7 consists of an extra edge going from the unauthorized "CaseOfficer" node to the "Customer" node, and as such is only different from Scenario 5 by the label on the edge and the targeted node.

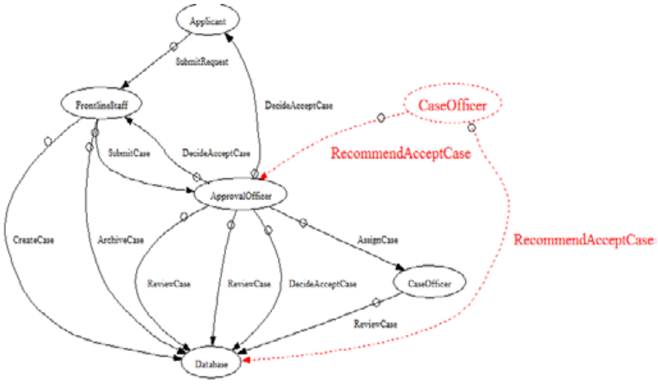

Figure 4. Scenario showing unauthorized CaseOfficer's handling of case.

Finally, for scenario 8, we represented time in the graph as the number of hours since midnight, and we enhanced GBAD to use a simple statistical analysis of numerical attributes as part of its evaluation of the graph structure. In this case, we randomly inserted two anomalies into the graph, and the GBAD-P algorithm was able to successfully discover both anomalies where access to the company database was during unexpected hours, with no false positives reported. While the structure was the same, the time information (represented as a number), provides extra information that aides in the insider threat detection. Also, it is important to note that no false positives are reported with this scenario.

C. *Cybercrime*

Another example of insider threat is the leaking of information by employees with access to confidential and sensitive information. For example, in April of 2010, a federal grand jury indicted a former senior NSA executive for leaking classified information to a reporter [27]. As part of the 2009 IEEE Symposium on Visual Analytics Science and Technology (VAST 2009), we applied our approaches to one of their mini challenges that involved various aspects of a fictional insider threat scenario where someone is leaking information [28]. The goal of these challenges is to allow contestants to apply various visual analysis techniques so as to discover the spy and their associated actions. While our GBAD approaches are not "visually based", we chose to apply our algorithms to the mini-challenge that consists of badge and network IP traffic. The data set is comprised of employee "badge swipes" during the month of January in 2008, and the IP log consists of all network activity to and from the facility. One of the goals of this mini-challenge was to determine which computers the "spy" used to send the sensitive information.

We can separate the cybercrime discovery process into three separate tasks:

1. Discover the anomalous network activity,
2. Create *targeted* graphs for just those days and people that might be involved in the anomalous activity, and
3. Use GBAD to discover which employees participate in anomalous activity.

The first stage of this process is to discover the network activity that is unusual – or the source of illegal transmissions. Rather than apply a graph-based approach to the discovery of what would be numerical/statistical anomalies (i.e., non-structural anomalies), we can do a simple analysis of the actual records. Sorting the IP logs by amount of traffic, one discovers that the top five transmissions are all to the same destination IP, 100.59.151.133 on port 8080:

```
…
Synthetic Data   37.170.100.31   2008-01-15T17:03   100.59.151.133  8080      9513313     14324
Synthetic Data   37.170.100.20   2008-01-24T17:07   100.59.151.133  8080      9732417     42347
Synthetic Data   37.170.100.13   2008-01-22T08:50   100.59.151.133  8080      9984318     42231
Synthetic Data   37.170.100.56   2008-01-29T15:41   100.59.151.133  8080     10024754     29565
Synthetic Data   37.170.100.8    2008-01-31T16:02   100.59.151.133  8080     13687307    485421
```

In the IP log file, the first column is the type of data, the second column is the source IP, the third column is the date and time, the fourth column is the destination IP, the fifth column is the destination port, the sixth column is the size of the transmission, and the last column is the size of the response record. In fact, 17 of the 32 highest-transmission records have this same destination IP - clearly an unusual volume of traffic to a single, external destination. In addition, with our graph-based approach, we can *verify* the anomalousness of the traffic based upon the relationship of the activity within the graph. For example, knowing that employee 31's computer is one of the computers that sent the supposedly illegal transmissions (see the top record above), we can analyze the subgraph of that employee's activity on that day.

In order to discover an insider committing this form of cybercrime, we can make two assumptions:

1. The insider never uses their own computer (for fear of their actions being traced back to them), and
2. The insider only uses someone else's computer when they are in the classified area (as that is the only time we know that they are not in their office).

Using these two assumptions (which were correct assumptions made by others as part of this competition), we can then focus on the generation of graphs that (1) exclude people whose computer was compromised from being considered as suspects, and (2) reduce the graph search space to only those days where the illicit transmissions took place. In this data set, 10 employees are removed from being considered as suspects, and only the activity of other employees during the anomalous network activity are represented in the graph. This will enable us to analyze abnormal structure in the graph during the times of the crimes.

So first we create graphs consisting of subgraphs that represent employee movements for *each* targeted day (i.e., the days when the illicit transmissions took place), as well as graphs that represent the movements for each employee over *all* of the targeted days. Each subgraph will contain a "backbone" of movement vertices. Attached to the movement vertices will be two vertices representing where the person was before entering the current location and the current location (i.e., outside, building, classified). The edges will be labeled start and end, respectively. Then, if network traffic is sent before the person moves again, a network vertex will be created and linked to the movement vertex via a sends edge. The network vertex will also be linked to a vertex with a numerical label, representing how many messages are sent before the next movement occurs. The result is a graph topological representation as shown in Figure 5.

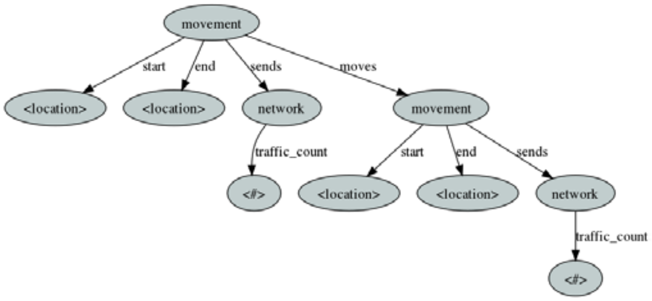

Figure 5. Graph topology of movement and activity.

In the partial example shown in Figure 6, a person enters from the outside, transfers some data across the network, and then moves into the classified area.

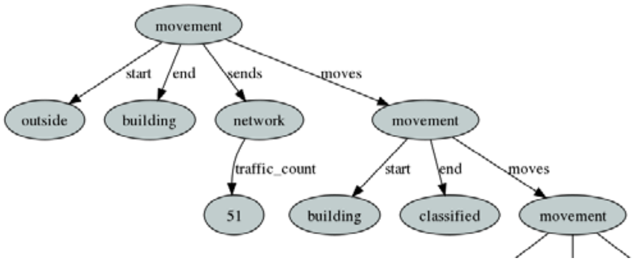

Figure 6. Example movement and activity (partial graph shown).

We created a tool to process the comma-delimited proxy log and IP log files and output graph files for use with GBAD. Once the graph files are created, GBAD can then be used to obtain (1) the normative pattern discovered in the specified graph input file and (2) the top-N most anomalous patterns.

Using this graph representation, GBAD discovers the normative pattern shown in Figure 7.

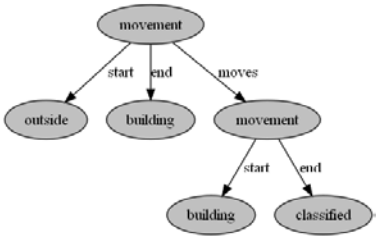

Figure 7. Normative pattern.

After uncovering the normative pattern, GBAD can then use its three algorithms to discover all of the possible structural changes that can exist in a graph (i.e., modification, deletions, and insertions).

The VAST data set consists of the activities of 60 employees at an embassy over the month of January in 2008. As stated earlier, there are 17 transmissions to the suspect IP. Based upon our first assumption, we can remove 10 employees from the list of suspects (some employees' computers were compromised more than once). We can also reduce our data set down to just the days where the anomalous transmissions took place, which consists of 8 of the 31 available days worth of information. This subset of the data is then the baseline for our GBAD analysis.

Using these targeted graphs (8 day graphs and 50 people graphs), we ran the GBAD algorithms using default parameter settings, where it would report only the most anomalous instances, rather than the top-K instances. On the graphs that represent individual people and their movements and network activities across all targeted days, the GBAD-MDL algorithm discovers 12 employees as having anomalous movements and activities, and the GBAD-MPS algorithm reports 8 employees as anomalous. On the graphs that represent all movements and activities for each targeted day, GBAD-MDL reports 6 employees as anomalous while GBAD-MPS reports 2 employees. However, there is an interesting commonality across all four experiments. If you take the overlap (intersection) between them, in other words which employees are reported in ALL of the experiments, one discovers that there are only 2 employees that are very suspicious: employee 49 and employee 30.

We can further distinguish a difference between these two employees by analyzing the graphs and GBAD results. From the GBAD results, employee 30 is reported as the most anomalous (score-wise) on 6 of the 8 days, with employee 49 being the most anomalous on the other 2. Also, employee 30 is the only employee with the structural anomaly shown in Figure 8.

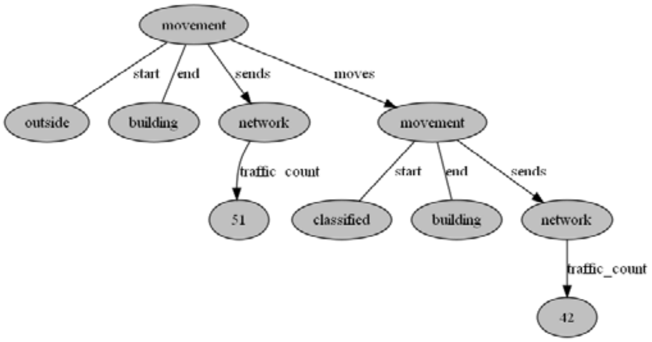

Figure 8. Anomalous structure (in the graph).

In Figure 8 (only the parts of the graph necessary for this observation are shown), one will notice that the employee initially moves from the outside into the building. However, their next move is from the classified area into the building – with no movement into the classified area before that. This is called "piggybacking", where an employee does not use their badge, but instead follows on the heels of another employee. Employee 30 is not only the only employee to piggyback into the classified area, but they do it several times. Perhaps their intent is to gather classified information without a trace of ever entering the area. Unfortunately (for them), they had to badge-out of the area – resulting in a structural anomaly in their movement.

D. *Tests and Performance*

We have also performed a variety of synthetic tests on each of the algorithms using graphs of various sizes and connectivity. In each experiment, we randomly modified, inserted or deleted (as well as a combination of multiple changes) vertices and edges. We measured GBAD's ability to correctly identify the intended anomalies (true positives) versus reporting anomalies we did not introduce into the data (false positives). The overall results were that GBAD never found less than 95% of the anomalies, with minimal (none in most cases) false positives reported.

The average running times of the algorithms is anywhere from < 1 second to ~45 minutes, depending upon the size of the graphs. The larger the graph, as well as the number of subgraphs one wants to analyze for anomalous structure, the greater the runtime for the algorithms. In general, the running time of GBAD is polynomial in the size of the graph and the parameters of the algorithm. Figure 9 shows the performance of the GBAD algorithms on our synthetic experiments as a linear log-log plot.

The ability to discover the anomalies is sometimes limited by the *resources* allocated to the algorithm. Given a graph where the anomalous substructure consists of the *minimal* deviation from the normative pattern, if a sufficient amount of processing time and memory is provided, all of these algorithms will discover the anomalous substructure with no false positives. However, the ability to discover anomalies (per our definition) is also hampered by the amount of *noise* present in the graph. The issue is that if noise is a smaller deviation from the normative pattern than the actual anomaly, it may score higher than the targeted anomaly (depending upon the frequency of the noise).

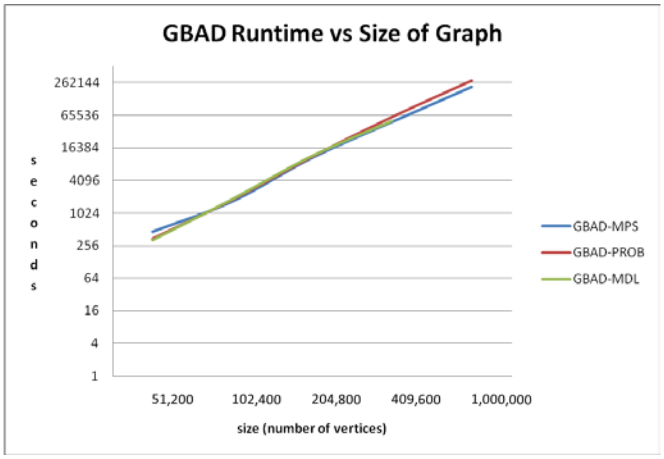

Figure 9. GBAD running times on synthetic graphs.

The reader should refer to [22] for more information regarding the experimental results and analysis of GBAD.

V. FUTURE WORK

A. *Data Changing Over Time*

So far, GBAD has only been applied to static graphs. However, many domains in which we desire to detect anomalies are dynamic; that is, the information is changing over time. One solution to this scenario is to collect data over a time window, build a graph from this data that may or may not explicitly represent time, and then apply GBAD to the graph. While this solution will find anomalies to patterns within the time window, any dynamic component to the patterns and anomalies will rely on a proper representation of time and a sufficiently long time window in which to observe the patterns' regularity.

One approach to detecting patterns of structural change in a dynamic graph, which has been successfully applied to the domain of biological networks [29], is called DynGRL [30]. In this approach, DynGRL first learns how one graph is structurally transformed into another using graph rewriting rules, and then abstracts these rules into patterns that represent the dynamics of a sequence of graphs. The goal of DynGRL is to describe how the graphs change over time, not merely whether they change or by how much.

Graph rewriting rules represent topological changes between two sequential versions of the graph, and transformation rules abstract the graph rewriting rules into the repeated patterns that represent the dynamics of the graph. Figure 10 shows the framework of this approach. The dynamic graph contains a sequence of graphs that are generated by sampling snapshots from a continuously-changing graph. First, the approach learns graph rewriting rules including removals (R_i) and additions (A_{i+1}) between two sequential graphs G_i and G_{i+1} (Figure 10 (B)), and generates a list of all graph rewriting rules (Figure 10 (C)). The final step is to learn the transformation rules to abstract the structural change of the dynamic graph based on the repeated patterns in the graph rewriting rules. If some structural changes are repeated in the dynamic graph, there exist common subgraphs in the R's and A's.

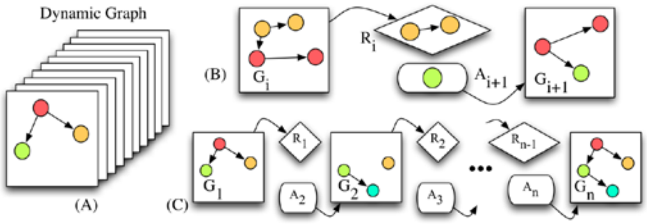

Figure 10. Framework of dynamic graph analysis.

In order to detect anomalies in the change of dynamic graphs, we must first learn how one graph is structurally transformed into another, and then abstract patterns that represent the dynamics of a sequence of graphs. In order to detect anomalies, the goal is to describe how the graphs change over time, and discover those changes that are structurally anomalous. Specifically, we want to (1) look for structural modifications, insertions and deletions to nearby instances of the transformation rules as potential anomalies to the normative pattern, and (2) detect anomalies in the temporal application of the transformation rules, e.g., when in some cases the structure does not appear exactly four times after it was last removed.

Using this approach, we coupled DynGRL with GBAD to produce a system for discovering anomalies in dynamic graphs which we call DynGBAD. First, DynGBAD produces a sequence of difference graphs for each pair of graphs in the time-slice sequence, searching for recurring patterns in these difference graphs. DynGBAD then analyzes these difference graphs using the normative recurring patterns discovered by the relational learner (DynGRL) and identifies anomalies to these patterns (GBAD). A dynamic anomaly may be a change in the pattern at some point in time (similar to what GBAD already does), but also may consist of a change in the period of recurrence of the pattern. Our hypothesis is that a representation that links the difference graphs together will allow DynGBAD to detect such anomalies.

B. Speed and Efficiency

While the GBAD approach has been effective on various datasets, its use of the computationally complex graph matching operations hinders its application to large datasets. One graph-based knowledge discovery approach that has shown to be expedient without losing any accuracy can be found in the many frequent subgraph miners. Existing approaches such as GASTON [31], gSpan [32], GBI [33] and Grew [34] use canonical graph representations to efficiently return frequent substructures in a database that are represented as a graph. In addition, each of these approaches has demonstrated significant improvements in processing time when applied to complex data sets.

In order to demonstrate the potential effectiveness of implementing anomaly detection algorithms into a frequent subgraph mining approach, we have initially tried to implement the GBAD-MDL algorithm into GASTON. The way GASTON works is that embeddings (substructures) are analyzed for potential refinements, where the refinement (or extension) could lead to a potentially frequent substructure. The advantage lies in its *Apriori* approach whereby prior knowledge of frequent itemset properties is used to discover those substructures that are frequent [35]. 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.

To test the potential of such an approach, we used actual cargo shipping manifest data from the Customs and Border Protection agency [36]. Taking a sampling of shipments received at the port in Norfolk, Virginia, three graphs of ~10,000 vertices and edges were generated: one with an anomalous vertex, one with an anomalous edge, and one with both an anomalous vertex and an anomalous edge. For all three graphs, GASTON-GBAD, with a threshold of 0.1 (i.e., anomalies that consist in less than 10% of the substructure), was able to correctly identify the anomalous instance, with no false positives, and in less than ~7 minutes. Even though no algorithmic improvements were made to the baseline GASTON algorithm, this still was a significant improvement over the SUBDUE-GBAD implementation which required ~54 minutes to discover these same anomalies.

VI. Conclusions

Results from running the GBAD algorithms on e-mails, business processes and movement activities show how these graph-theoretic approaches can be used to identify insider threats. While we have been able to achieve some minimal successes when applying graph-theoretic algorithms to dynamic graphs that change over time, clearly we have only begun to scratch the surface. In addition, initial results from implementing GBAD algorithms in frequent subgraph miners have demonstrated a potential for a significant speed-up in performance.

VII. References

[1] AFCE, "2006 AFCE Report to the Nation on Occupational Fraud & Abuse," *Association of Certified Fraud Examiners*, 2006.

[2] Identity Theft Resource Center, "Data Breaches: The Insanity Continues," *ITRC*, January 8, 2010.

[3] L. Ponemon, "Cyber Crime: The 2009 MegaTrend," *CSO*, 2009.

[4] United States Department of Justice, "Computer Crime & Intellectual Property Section", http://www.justice.gov/criminal/cybercrime/, 2010.

[5] J. Vijayan, "Insider at Cal Water steals $9M and runs," *Computerworld Security*, May 22, 2009.

[6] J. Kirk, "In poor economy, IT pros could turn to e-crime," *IDG News Service*, March 25, 2009.

[7] J. Bush, "Survey suggests economy could lead to cybercrime increase," *Purdue University News Service*, March 19, 2009.

[8] C. Noble and D. Cook, "Graph-Based Anomaly Detection," *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 631-636, 2003.

[9] S. Lin and H. Chalupsky, "Unsupervised Link Discovery in Multi-relational Data via Rarity Analysis," *Proceedings of 3rd IEEE ICDM Intl. Conf. on Data Mining*, pp. 171-178, 2003.

[10] D. Chakrabarti, "AutoPart: Parameter-Free Graph Partitioning and Outlier Detection," *PKDD 2004, 8th European Conference on Principles/Practices of KDD*, pp. 112-124, 2004.

[11] J. Rissanen, *Stochastic Complexity in Statistical Inquiry*, World Scientific Publishing Company, 1989.

[12] J. Shetty and J. Adibi, "Discovering Important Nodes through Graph Entropy: The Case of Enron Email Database," *KDD, Proceedings of the 3rd international workshop on link discovery*, pp. 74-81, 2005.

[13] J. Sun et al., "Relevance search and anomaly detection in bipartite graphs," *SIGKDD Explorations* 7(2), Pp. 48-55, 2005.

[14] M. Rattigan and D. Jensen, "The case for anomalous link discovery," *ACM SIGKDD Exploration News*, 7(2):41-47, 2005.

[15] C. Priebe et al., "Scan Statistics on Enron Graphs," *Computational and Mathematics Organization Theory*, Volume 11, Number 3, p229 – 247, 2005.

[16] S. Martin et al., "Analyzing Behavioral Features for Email Classification," *CEAS 2005: Conference on Email and Anti-Spam*, July 21-22, 2005.

[17] J. Diesner and K. Carley, "Exploration of Communication Networks from the Enron Email Corpus," *Computational and Mathematical Organization Theory*, Vol 11,Issue 3, p. 201-228, 2005.

[18] M. Kurcz et al., "Spectral Clustering in Telephone Graphs," *Joint 9th WEBKDD and 1st SNA-KDD Workshop '07*, August 12, 2007.

[19] D. Swayne et al., "Exploratory Visual Analysis of Graphs in GGobi," *Proceedings of the 3rd International Workshop on Distributed Statistical Computing*, DSC 2003, March 20-22, 2003.

[20] M. Hampton and M. Levi, "Fast spinning into oblivion? Recent developments in money-laundering policies and offshore finance centres," *Third World Quarterly*, Volume 20, Number 3, June 1999, pp. 645-656, 1999.

[21] D. Cook and L. Holder, "Graph-based data mining," *IEEE Intelligent Systems* 15(2), 32-41, 1998.

[22] W. Eberle and L. Holder, "Anomaly Detection in Data Represented as Graphs," *Intelligent Data Analysis, An International Journal*, Volume 11(6), 2007.

[23] M. Randazzo, M. Keeney, E. Kowalski, D. Cappelli and A. Moore, "Insider Threat Study: Illicit Cyber Activity in the Banking and Finance Sector," http://www.cert.org/insider_threat/, 2004.

[24] E. Kowalski, D. Cappelli and A. Moore, "Insider Threat Study: Illicit Cyber Activity in the Information Technology and Telecommunications Sector," http://www.cert.org/insider_threat/, 2008.

[25] E. Kowalski, T. Conway, S. Keverline, M. Williams, D. Cappelli and A. Moore, "Insider Threat Study: Illicit Cyber Activity in the Government Sector," http://www.cert.org/insider_threat/, 2008.

[26] A. Chun. "An AI framework for the automatic assessment of e-government forms," *AI Magazine*, Volume 29, Spring 2008.

[27] "Former Senior NSA Exec Charged With Leaking Classified Info", Security Dark Reading, April 15, 2010.

[28] W. Eberle, L. Holder and J. Graves, "Detecting Employee Leaks Using Badge and Network IP Traffic," *IEEE Symposium on Visual Analytics Science and Technology*, October 2009.

[29] C. You, L. Holder and D. Cook, "Learning Patterns in the Dynamics of Biological Networks," *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, June 2009.

[30] C. You, L. Holder and D. Cook, "Graph-based Data Mining in Dynamic Networks: Empirical Comparison of Compression-based and Frequency-based Subgraph Mining," *IEEE International Conference on Data Mining (ICDM) Workshop on Analysis of Dynamic Networks*, December 2008.

[31] S. Nijssen and J. Kok, "A Quickstart in Frequent Structure Mining Can Make a Difference," *International Conference on Knowledge Discovery and Data Mining, SIGKDD*, pp. 647-652, 2004.

[32] X. Yan and J. Han, "gSpan: Graph-Based Substructure Pattern Mining," *Proceedings of International Conference on Data Mining, ICDM*, pp. 51-58, 2002.

[33] T. Matsuda, H. Motoda, T. Yoshida and T. Washio, "Knowledge Discovery from Structured Data by Beam-Wise Graph-Based Induction," *PRICAI 2002: Trends in Artificial Intelligence*, Vol 2417, Pp. 123-141, 2002.

[34] Kuramochi, M. and Karypis, G. Grew – A Scalable Frequent Subgraph Discovery Algorithm. *IEEE International Conference on Data Mining, ICDM*, 2004.

[35] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," *Proceedings of the International Conference of Very Large Databases (VLDB)*, Pp. 487-499, September, 1994.

[36] W. Eberle and L. Holder, "Detecting Anomalies in Cargo Shipments Using Graph Properties," *Proceedings of the IEEE Intelligence and Security Informatics Conference*, 2006.

[37] L Wang, A. Singhal, and S. Jajodia, "Measuring the Overall Security of Network Configurations Using Attack Graphs", *Data and Applications Security*, LNCS, pp. 98-112, 2007.

[38] X. Wan , E. Milios , J. Janssen , and N. Kalyaniwalla, "Link-Based Event Detection in Email Communication

Networks", *ACM Symposium on Applied Computing*, 2009.

[39] Z. Huan, "Link Prediction Based on Graph Topology: The Predictive Value of the Generalized Clustering Coefficient", *Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (LinkKDD2006)*, 20 August, 2006.

[40] L. Akoglu, M. Mcglohon, and C. Faloutsos, "OddBall: Spotting Anomalies in Weighted Graphs", *Pacific-Asia Conference on Knowledge Discovery and Data Mining* (PAKDD), June 23, 2010.