International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 7, pp: 418-424 www.ijaem.net ISSN: 2395-5252

Scalable Unsupervised Ensemble Algorithm for Effective Insider Threat Detection

Ajayi Adebowale[1], Ajayi Olutayo[2], Idowu Sunday[3], Ogbonna A.C[4], Ajayi Oluwabukola[5]

[1]*Department of Computer Science, Babcock University*
[2]*Department of Information Communication Technology, FUNAAB.*
[3]*Department of Computer Science, Babcock University*
[4]*Department of Computer Science, Babcock University*
[5]*Department of Computer Science, Babcock University*

Date of Submission: 25-09-2020 Date of Acceptance: 08-10-2020

ABSTRACT: Insider threats remain one of the oldest and notorious threats to information security. Early detection remains key to preventing insider attacks on an information system. The vast amount of enterprise data and the little data points pertaining to insider threats calls for techniques to handle the rare class problem.

This study conceptualised insider threat data as a streaming data problem. Scalability of insider threat detection systems represents a gap in knowledge in this disposition. Building on existing unsupervised ensemble stream mining techniques, this study proposed an insider threat detection algorithm and evaluated it using Centre for Analysis of Internet Data (CAIDA) Anonymized trace dataset for 2015. CAIDA datasets was used to ascertain the scalability of quantised dictionary construction by applying a distributive approach to graph based anomaly detection (GBAD). Pattern learning anomaly detection system processes GBAD in a streaming approach. Dictionary construction was done using Apache Spark on top of the Hadoop stack.

PLADS enhanced GBAD successfully discovered the same anomalous substructure within a streaming approach in a fraction of the time (642 seconds) it took to process the entire graph (59,743 seconds) when applied on the CAIDA Anonymised 2015 dataset. Application of Apache Spark as the distributed computing framework for construction of quantised dictionaries of user command data depicted a reduction in processing time under varying input sizes and number of reducers

In conclusion, scalability of Insider Threat Detection systems is essential and a complexity analysis of proposed algorithms showed it scales to increased number of users of the system. The implemented prototype system using Apache Spark scaled to increasing workloads showing its usefulness for early detection of insider threats. This study recommends the use of unsupervised learning ensembles and distributed frameworks for effective detection of insider threats

KEYWORDS: Gbad, Plads, Caida, Quantised Dictionary, Apache Spark, Hadoop

I. INTRODUCTION

Insider threat detection is fast gaining the attention of the information security community given its persistence and notorious difficulty partly due to its adversarial nature. Ancient Greeks had an aphorism for it: Who will watch the watchers! As insider threats are authorised users of an information system whose operations lie within the boundaries of their permissions and closely mimic legitimate user behaviour. Who watches the disgruntled system administrator, who decides to delete essential system files or perform masquerade attacks on the system using other user's credentials? The insider threat scenario is further hardened by the fact that only a few data points out of the vast amount of enterprise data are related to insider threat detection, constituting a rare class problem. Traditionally, supervised learning methods require well balanced labelled datasets (containing representative numbers of benign and anomalous instances) to build effective classification models so as to avoid overfitting. This poses a problem in the use of supervised learning in identifying anomalous data points corresponding to malicious insiders since they belong to the minority class. This is especially so in previous researches where proposed insider threat detection models are evaluated on static data streams. This stationarity assumption deviates from reality as data pertaining to insider threat is a

International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 7, pp: 418-424 www.ijaem.net ISSN: 2395-5252

continuous data stream of infinite length with the inherent concept drift and feature evolution characteristics of continuous stream data.

The rare class problem is better handled with unsupervised learning which builds detection models on non-anomalous data hereby creating a profile of normal behaviour and classifies instances as anomalous based on their geometric differences from the established model of normalcy. The problem with such approaches as seen in previous research is the high rate of false positives as previously unseen albeit normal behaviour is erroneously classified as anomalous. The high amount of false alerts distracts the analyst, allowing genuine insider attacks evade detection while the analyst chases shadows. Previous unsupervised learning approaches also make the stationarity assumption leading to high positive rates as the concept drift characteristic of stream data is not put into consideration in the development and evaluation of proposed algorithms.

Ensembles of algorithms have been trained on user command sequence data in a bid to achieve improved detection accuracy in the presence of concept drift and feature evolution characteristics of continuous stream data. However, developing quantised dictionaries of user command sequence data remains a computationally expensive endeavour given the variability and velocity of collected user command sequence data.

This study elucidates the use of a streaming approach to graph based anomaly detection (GBAD) to enable faster detection of normative substructures in user command sequence data.

II. RELATED WORKS

Insider threat detection research has applied ideas from both intrusion detection and external threat detection [1,2] Hybrid high order Markov chain model detects anomalies by identifying a signature behaviour for particular used based on their sequence of command [3]. The probabilistic Anomaly Detection (PAD) algorithm [4] is a general purposed algorithm for anomaly detection in windows environment that assumes anomalies as a rare event in the training data. A number of detection methods have been applied to data set of "truncated" UNIX shell commands for 70 users [5]this commands were collected using the UNIX acct auditing mechanism, for each user a number of commands were gathered over a period of time. The detection methods used was supervised using multi-step Markovian model and combination of Bayes and Markov approaches. [6] argues that the data set were not appropriate for the masquerade task pointing out that the period of the data gathering varied greatly from each user and also the commands were not logged in the order in which they were typed instead they were coalesced when the application terminated the audit mechanism. This led to the unfortunate consequence of possible faulty analysis of strict sequence data. Therefore, in this work the dataset were not considered.

These approaches differ from the approach in this work in that the learning approaches are static and do not learn evolving streams. In other words, stream characteristics of data were not explored further, hence static learning performance may degrade over time. Past works explored unsupervised learning for insider threat detection using static streams [7].

There are two basic unsupervised approaches to adaptation: incremental learning [8, 1] and ensemble-based learning [9]. Past work demonstrated that ensemble-based approaches are the more effective of the two [10]).

Ensembles has been used in the past to bolster the effectiveness of positive/negative classification [8], by maintaining an ensemble of K models that uniformly vote on the final classification, the number of false negatives (FN) and false positives (FP) for a test set can be reduced. As better models are created, poorer models are discarded to maintain an ensemble of size exactly K [10], this helps the ensemble evolve with the changing characteristics of the stream and keeps the classification task tractable.

Researchers have explored unsupervised learning [11] for insider threat. However, this learning algorithm is static in nature, although unsupervised approach will be used in this work, more data will be used for it and it will learn from evolving stream over time.

Few researchers have conceptualised insider threat detection in stream mining area [12, 3, 4]. Unsupervised learning has been applied to detect insider threat in a data stream [5, 8, 9], but this works did not consider sequence data for threat detection. Instead it considered data as graph or vector, finding normative patterns and applying ensemble based technique to handle changes.

User's repetitive activities may constitute user profiles, to find a normative pattern over dynamic data streams of unbounded length is challenging due to the requirement of one pass algorithm. For this unsupervised learning approach is used by making use of compressed/ quantized dictionary to model common behaviour sequences. This unsupervised approach needs to identify

International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 7, pp: 418-424 www.ijaem.net ISSN: 2395-5252

normal user behaviour in a single pass [13]. The major challenge with these repetitive sequences is their variability in length. To tackle this problem a dictionary can be generated which will contain any possible combination of patterns existing in the gathered data stream.

Recently, research has attempted to mine frequent closed subgraphs in non- stationary data streams. One of such approach is the AdaGraphMiner, which maintains only the current frequent closed graph, utilizing estimation techniques with theoretical guarantees. [14]. Empirical experiments showed the effectiveness of this approach on graph streams representing chemical molecules and structural representations of cancer data. Moreover there has been recent attempts to discover outliers in massive network stream using structural connectivity model. Some researchers also attempted to handle the issue of sparseness in massive networks by dynamically partitioning the network [15]. Applying techniques such as reservoir sampling methods that compress a graph stream, one can search for structural summaries of the underlying network. The aim of this type of outlier detection is to identify graph objects which contains unusual bridging edges or edges between regions of a graph that rarely occur together. However, all of these approaches have not addressed the issue of scalability associated with performing graph bases anomaly detection.

Some approaches have detected outliers in graph streams, their objective was to identify unusual clusters of subgraphs in the graph by analyzing the statistical nature of the existence of edges as opposed to discovering anomalies in the structure of a graph or graph pattern. Besides, while some works attempted to discover anomalous subgraphs using ensembled based approach [16, 17] based on the GBAD approach [13, 12], that type of approach does not address the issue of scalability.

III. DATASET

The data sets used for training and testing sequence data is UNIX dataset from the University of Calgary project. In the dataset, 168 trace files were collected from 168 different users of Unix csh. The Calgary dataset [18] was modified by [3] for masquerade detection. This study use the guidelines given by [3] to inject masquerades commands. From the list of users, those users who have executed more than 2400 commands (that did not result in an error) were filtered out to form the valid pool of users. These 2400 commands were split into 12 chunks of 200 commands each. The first chunk was used as the training chunk. The other 11 chunks were the testing chunks. In the testing chunks, a number of masquerade data blocks comprising of 10 commands were inserted at random positions.

From the list of invalid-users, 25 of them were chosen at random and a block of one hundred commands from the commands that they had executed were extracted and put together to form a list of 2500 (25 * 100) commands. This commands were brought together as 250 blocks of 10 commands each. Out of these 250 blocks, 30 blocks were chosen at random as the list of masquerade commands (300 commands). Table 1 shows the description of the dataset used.

Table 1.0 Description of Dataset

Description	Number
Number of valid users	37
Number of invalid users	131
Number of valid commands per user	2400
Number of anomalous commands in testing chunks	300

Normal user profiles are considered to be repetitive daily or weekly activities which are regular sequences of commands. These repetitive command sequences are called normative patterns and these patterns reveals the regular behavior of a user. When a user suddenly demonstrates unusual activities an alarm is flagged for potential insider threat.

Therefore, in order to identify an insider threat, a need to find normal user behavior is required. For that, sequences of commands are collected and patterns are observed within these command sequences are identified in an unsupervised fashion. The unsupervised approach needs to identify normal user behavior in a single pass, one major challenge is the variability in length with these repetitive sequences. To combat this issue, a dictionary is generated which to contain combination of possible normative patterns existing in the gathered data stream.

IV. SCALABLE UNSUPERVISED LEARNING FOR SEQUENCE DATA

Unsupervised learning techniques for non-sequence data, and formulas necessary for understanding the inner workings of unsupervised learning are discussed. It provides adequate details as to exactly how each of the methods arrived at detecting insider threats and how the ensemble models are built, modified and discarded.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 7, pp: 418-424 www.ijaem.net ISSN: 2395-5252

Algorithm 1 uses three varieties of pattern learning anomaly detection system (PLADS) [7] to infer potential anomalies using each model. PLADS is a pattern learning algorithm in GBAD which is a graph-based approach to finding anomalies in data by searching for three factors: modifications, insertions, and deletions of vertices and edges. Each factor runs its unique algorithm that determines a normative substructure and attempts to find the substructures that are similar but not entirely identical to the discovered normative substructure. A normative substructure is a recurring subgraph of vertices and edges that, when coalesced into a single vertex, most compresses the overall graph [13]. The rectangle in Figure 1 identifies a typical example of normative substructure for the depicted graph.

SUBDUE [19] was used to find normative substructures. The best normative substructure can be characterized as the one with minimal description length (MDL):

$$L(S,G) = DL(G \mid S) + DL(S)(1.0)$$

where G is the entire graph, S is the substructure being analyzed, DL(G | S) is the description length of G after being compressed by S, and DL(S) is the description length of the substructure being analyzed. Description length DL (G) is the minimum number of bits necessary to describe graph G [7].

Insider threats appear as small percentagedifferences from the normative substructures, this is because insider threats attempts to mimic legitimate system operations closely except for small variations embodied by illegitimate behavior. Three different approaches for identifying such anomalies was applied.

V. GBAD-MDL

For determining the best compressing normative substructure, GBAD-MDL searches for deviations from normative substructure in subsequent substructures by analyzing substructures of the same size as the normative one, differences in the edges and vertices' labels and in the direction or endpoints of edges are then identified. The most anomalous of these are those substructures for which the fewest modifications are required to produce an isomorphic substructure to the normative one.

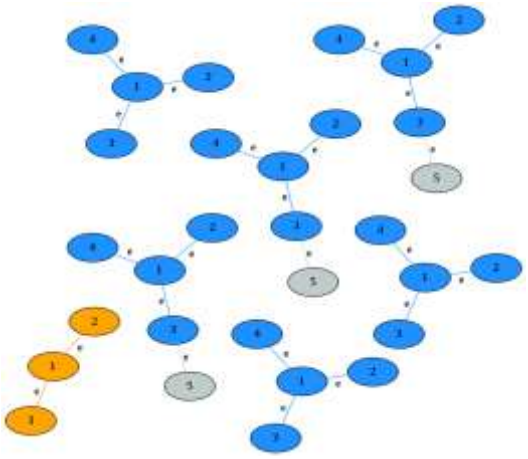

Figure 1. Graphical depiction of the normative pattern and anomaly reported by GBAD-MDL.

a. GBAD-P

GBAD-P searches for insertions that when deleted, yields the normative substructure. Insertions made to a graph are seen as extensions of the normative substructure. GBAD-P calculates the probability of each extension based on edge and vertex labels and hence exploits label information to discover anomalies [13]. The probability is given by

$$P(A=v) = P(A=v \mid A)P(A) \quad (2.0)$$

where A represents an edge/vertex attribute and v represents its value. Probability P(A=v | A) can be generated by a Gaussian distribution:

$$\rho(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp - \frac{(x-\mu)^2}{2\sigma^2} \quad (3.0)$$

where μ is the mean and σ is standard deviation. A higher value of ρ(x) corresponds to more anomalous substructures.

GBAD-P therefore ensures that malicious insider behavior that is reflected by the actual data in the graph rather than merely its structure can be reliably identified as anomalous with the algorithm.

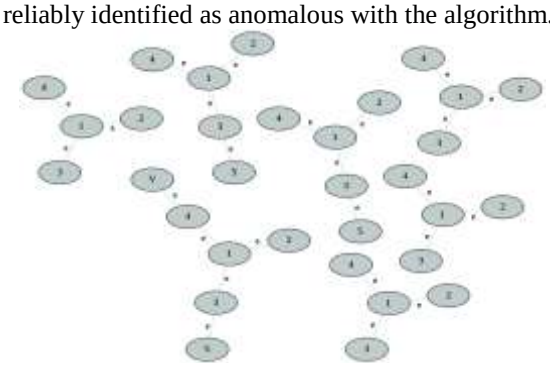

Figure 2: Graphical picture of graph input file containing an anomalous insertion

International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 7, pp: 418-424 www.ijaem.net ISSN: 2395-5252

b. GBAD-MPS

GBAD-MPS considers deletions that when re-inserted yields the normative substructure. For these, GBAD-MPS examines the parent structure. Changes orientation and size in the parent structure signifies deletions amongst the subgraphs. Figure 3 shows a graphical picture of graph input file containing anomalous deletion.

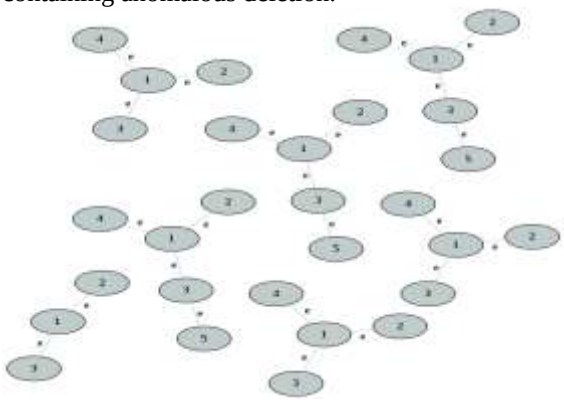

Figure 3: Graphical picture of graph input file containing anomalous deletion (Eberle and Holder, 2014)

c. Pattern Learning Anomaly Detection System (PLADS)

PLADS is a pattern learning algorithm in GBAD which uses a graph based approach to finding anomalies in data.PLADS is an algorithm for enhancing GBAD by evaluating anomalous substructures across partitions and reporting it. It receives as input a set of N graph partitions by partitioning a static graph.

1. Process N partitions in parallel

a. Each partition discovers top M normative patterns

b. Each partitions waits for all partitions to discover their normative patterns.

2. Determine best normative pattern, P among NM possibilities.
3. Each partition discovers anomalous substructures based upon P.
4. Evaluates anomalous substructures across partitions and report most anomalous substructure (s).
5. Process new partition

a. if oldest partition(s) has exceeded a threshold T (based upon criteria such as the number of available partitions or the time-stamped-age of the partition), remove partition(s) from further processing.

b. Determine top M normative patterns from new partition

c. Determine best normative pattern, P' among all active partitions

d. If (P' ≠P), each partition discovers new anomalous substructures based upon P'.

e. Else, only new partition discovers anomalous substructure(s).

f. Evaluate anomalous substructures across partitions and report most anomalous substructure(s).

g. Repeat.

d.**Unsupervised Adaboost ensemble Classification and Update Algorithm**

1. .Input: E (ensemble), t (test graph), and S (chunk)
2. Output: A (anomalies), E^0 (updated ensemble) and Strong classifier H(x)
3. $M^0 \leftarrow$ NewModel(S)
4. $E^0 \leftarrow E \cup \{M^0\}$
5. for each model M in ensemble E^0 do
6. $c_M \leftarrow 0$
7. for each q in model M do
8. $A_1 \leftarrow PLADS_P(t,q)$
9. $A_2 \leftarrow PLADS_{MDL}(t,q)$
10. $A_3 \leftarrow PLADS_{MPS}(t,q)$
11. $A_M \leftarrow ParseResults(A_1,A_2,A_3)$
12. end for
13. end for
14. for each candidate a in $^{S}{}_{M\in E}0$ A_M do
15. Given: $(x_1,y_1),...,(x_m,y_m)$; $x_i \in X, y_i \in \{-1,1\}$
16. Initialize
17. weights $D_1(i) = 1/m$ For t = 1,...,T:
18. (Call WeakLearn), which returns the weak classifier $h_t: X \rightarrow \{-1,1\}$ with minimum error w.r.t. distribution D_t;Choose $\alpha_t \in R$,
19. Update
20. $D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$
21. where $Z_t$is a normalization factor chosen so that D_{t+1} is a distribution
22. Output the strong classifier:

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

23. end for.

VI. EXPERIMENTAL SETUP

Proposed unsupervised adaboost ensemble approach was evaluated on a synthetic dataset obtained from randomising and duplicating Calgary dataset using a modification of [7] framework for masquerade detection to inject concept drift into the duplicated dataset.

VII. RESULTS

Table 2 gives a summary of results obtained on unsupervised learning approach.

Table 2: Non supervised learning approach on dataset

Accuracy	65%
False Positive Rate	54%
False Negative Rate	42%

VIII. CONCLUSION

Experimental results from this study showed that a streaming approach for insider threat detection outperforms traditional static stream mining. Proposed PLADS ensemble algorithm scales to increased workload and therefore aids early detection of insider threats.

REFERENCES

[1]. Eskin, E., Miller, M., Zhong, Z., Yi, G., Lee, W., and Stolfo, S. (2000). Adaptive model generation for intrusion detection systems. In Proc. ACM CCS Workshop on Intrusion Detection and Prevention (WIDP).

[2]. Eskin, E., Arnold, A., Prerau, M., Portnoy, L., and Stolfo, S. (2002). A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. In D. Barbara and S. Jajodia (Eds.), Applications of Data Mining in Computer Security, Chapter 4. Springer.

[3]. Liu, A., Martin, C., Hetherington, T., and Matzner, S. (2005). A comparison of system call feature representations for insider threat detection. In Proc. IEEE Information Assurance Workshop (IAW), pp. 340-347.

[4]. Liao, Y. and Vemuri, V. (2002). Using text categorization techniques for intrusion detection. In Proc. 11th USENIX Security Symposium, pp. 51-59.

[5]. Parveen, P., McDaniel, N., Evans, J., Thuraisingham, B., Hamlen, K., and Khan, L. (2013). Evolving insider threat detection stream mining perspective. International Journal on Artificial Intelligence Tools (World Scientific Publishing) 22 (5), 1360013-1-1360013-24.

[6]. Bifet, A., Holmes, D., Pfahringer, B., and Gavaldà, R. (2011). Mining frequent closed graphs on evolving data streams, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge discovery and Data Mining (KDD'11).

[7]. Davison, B., and Hirsh, H. (1998). Predicting sequences of user actions. in working notes of the joint workshop on predicting the future: Ai approches to time series analysis. In 15th National Conference on Arti_cial Intelligence and Machine, pp. 5-12. AAAI Press.

[8]. Eberle, W., and Holder, L. (2007). Mining for structural anomalies in graph-based data. In Proc. International Conference on Data Mining (DMIN), pp. 376-389.

[9]. Eberle, W., and Holder L. (2015), "Streaming Data Analytics for Anomalies in Graphs," IEEE International Symposium on Technologies for Homeland Security.

[10]. Parveen, P., Weger, Z., Thuraisingham, B., Hamlen, K., and Khan, L. (2011). Supervised learning for insider threat detection using stream mining. In Proceedings of the 23rd IEEE International Conference on Tools with Artificial Intelligence, Nov. 7-9, 2011, Boca Raton, Florida, USA (acceptance rate 30%) (Best Paper Award).

[11]. Eberle, W., and Holder, L. (2015). "Scalable Anomaly Detection in Graphs," Intelligent Data Analysis, an International Journal, Volume 19(1).

[12]. Masud, M. M., Al-Khateeb, K., Khan, L., Aggarwal, C., Gao, J., Han, J., and Thuraisingham, B. (2011). Detecting recurring and novel classes in concept-drifting data streams. In ICDM, pp. 1176-1181.

[13]. Gao, D., M. K. Reiter, and D. Song (2004). On gray-box program tracking for anomaly detection. In Proc. USENIX Security Symposium, pp. 103-118.

[14]. Hofmeyr, S., Forrest, S., and Somayaji, A. (1998). Intrusion detection using sequences of system calls. Journal of Computer Security 6 (3), 151-180Ju, W,. and Vardi, Y. (2001). A hybrid high-ordermarkov chain model for computer intrusion detection. Journal of Computational and Graphical Statistics.

[15]. Ajayi, A., and Idowu, S. (2013b). "AnEnhanced Data Mining Based Intrusion Detection System (Ids) Using Selective Feedback". International Journal of Computer and Information Technology

[16]. Ketkar, N. S., L. B. Holder, and D. J. Cook (2005). Subdue: Compression-based frequent pattern discovery in graph data. In Proc. ACM KDD Workshop on Open-Source Data Mining.

[17]. Schonlau, M., DuMouchel, W., Ju, H., Karr, A., Theus, M., and Vardi, Y. (2001). Computer intrusion: Detecting masquerades. Statistical Science 16 (1), 1-17.

[18]. Maxion, R. (2003). Masquerade detection using enriched command lines. In Proc. IEEE International Conference on

Dependable Systems & Networks (DSN), pp.5-14.

[19]. Parveen, P., Evans, J., Thuraisingham, B., Hamlen, K., and Khan, L. (2011). Insider threat detection using stream mining and graph mining, Privacy, Security, Risk and Trust (PASSAT), IEEE Third International Conference on and 2011 IEEE Third International Conference on Social Computing (SocialCom), 1102–1110.