Graph-Based Anomaly Detection using MapReduce on Network Records

Brandon Joyce[1] | Enyue Lu[2] | Matthias K. Gobbert[3]

1. The University of North Carolina at Greensboro | 2. Salisbury University | 3. University of Maryland, Baltimore County

Abstract

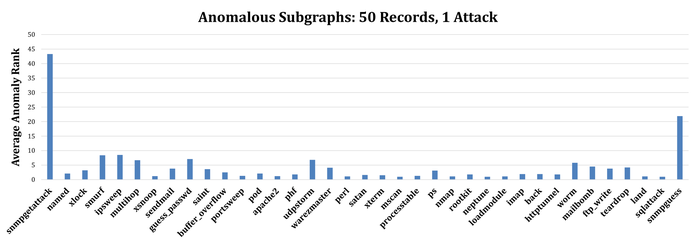
It has become increasingly complex to detect intrusion through network records due to large volumes of network traffic data. In order to detect anomalies and do it quickly, parallel strategies such as MapReduce can be used to improve the performance. In this project, we modeled network data as a graph in order to highlight the interconnected nature of data records. Our star configuration graph model allows for the use of a pattern recognition algorithm to identify network intrusions. We have developed a MapReduce algorithm that uses a SUBDUE graph compression method. We tested our algorithm on the 1999 KDD Cup network intrusion dataset. It has shown that our algorithm is capable of processing large amounts of data with high specificity and accuracy.

MapReduce Algorithm

Our proposed MapReduce algorithm basically contains three different types of Map and Reduce jobs. By modeling each network record as a star graph with outer nodes representing the record attributes, we were able to implement the Apache Hadoop MapReduce paradigm.

- The first Map accepts an arbitrary key with an individual record as its value and creates combinations of record attributes with a length of two. Each combination is used as an output key. The first Reduce counts the combinations so that the most common pattern can be found. To improve performance, we specify that each reducer returns the maximum pattern locally found.
- The second Map accepts the most common pattern as its key and an individual record as its value. If a record contains both attributes in the pattern, then they are removed and replaced with the full pattern; no Reduce is needed.
- Repeat above until there are no more patterns with length > 1.
- Finally, the third Map job is used to calculate anomaly scores.

Detection Results



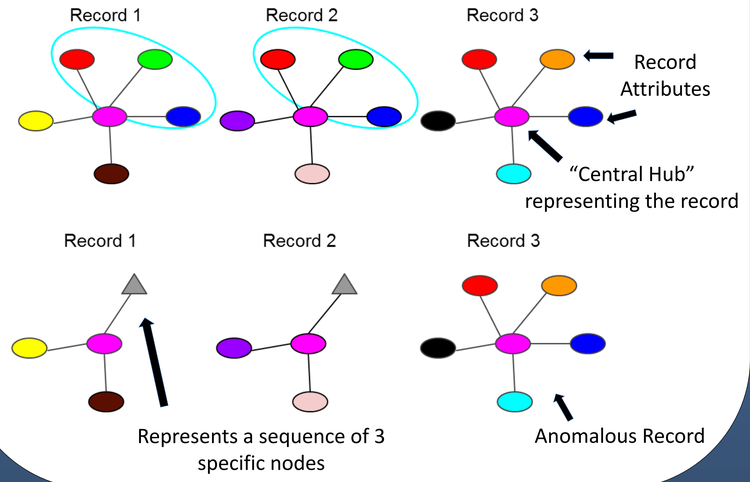
The average anomaly rank was calculated by sorting records based on their anomaly score after algorithm termination. Rank 1 means the highest likelihood for the anomaly. A lower average indicates better accuracy. Our average anomaly rank was 4.6 (slightly lower than in [1]). In addition, the average dropped as low as 3.6 at earlier iterations.

Graph Compression

SUBDUE is a graph-based knowledge discovery system. It uses graph compression to identify structural patterns in a given graph [1].

Graph Compression on Star Graphs that Represent KDD Records

- Examine network record attribute-nodes to find common patterns
- Compress common attribute-nodes into a single node
- After several compressions, anomalous records will have more nodes and consequently, a higher anomaly score
- In this way, we can identify common network patterns and isolate anomalous network records



Calculating Anomaly Scores

Anomaly Score is calculated by the formula in [1]:

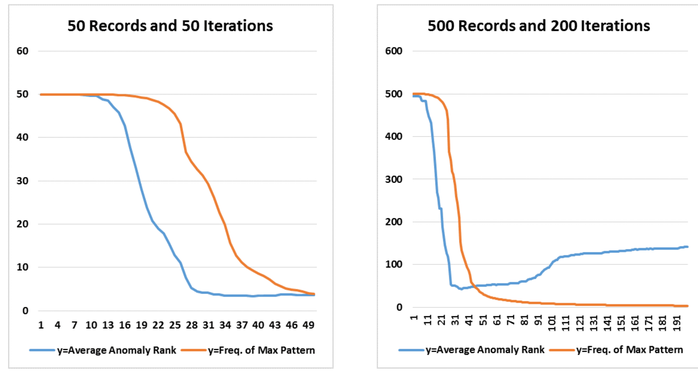
$$A = 1 - \frac{1}{n}\sum_{i=1}^{n}(n - i + 1) * c_i$$

n is the number of iterations, i is the current iteration, and c_i is the percentage of the star graph that was compressed/reduced on the ith iteration. The $n - i$ term in the above equation has the effect of weighting common patterns discovered at earlier iterations higher than the patterns discovered at later iterations. This is because patterns that are discovered at later iterations will become increasingly anomalous since the most common patterns will be removed at earlier iterations.

Since we only compress two nodes at a time, a star graph will only be reduced by a maximum of one node. For that reason, we define c_i to be dichotomous:

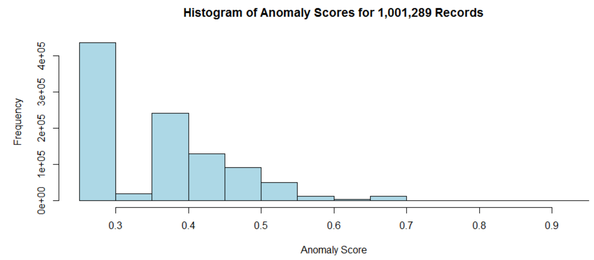
$$c_i = \begin{cases} \frac{1}{number\ of\ attributes}, & \text{graph was compressed} \\ 0, & \text{graph was not compressed} \end{cases}$$

Iterations vs. Accuracy



Attack rate for both graphs was 2%. Our algorithm performs better at early iterations than at later iterations. This intuitively makes sense because once we have eliminated all patterns with a certain percentage of occurrence, the remaining patterns are (in theory) anomalous by virtue of their infrequent occurrence.

Calculating Accuracy with Fewer Iterations



We used a sample from KDD [2] with all records labeled as "normal", "back", "satan", and "portsweep." We set the repetition parameter for the first two MapReduce jobs as 32 in our algorithm. By observing the above histogram, we labeled all records with an anomaly score ≥ 0.5 as anomalous. This yielded a true positive rate of 99.2%, a true negative rate of 94.5%, and an overall accuracy of 94.7%.

Acknowledgments

This research is funded by National Science Foundation CCF-1460900 under the Research Experience for Undergraduates Program. We would like to thank Professor Matthias K. Gobbert at University of Maryland, Baltimore County (UMBC) for allowing us to use the UMBC High Performance Computing Facility .

References

1. Noble, Caleb C., and Diane J. Cook. "Graph-Based Anomaly Detection." Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2003.
2. KDD dataset, 1999; http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html