Distributed Graph-Based Clustering for Network Intrusion Detection

Corbin McNeill ◦ Department of Mathematics and Computer Science ◦ Wheaton College, IL
Enyue Lu ◦ Department of Mathematics and Computer Science ◦ Salisbury University, MD
Matthias Gobbert ◦ Department of Mathematics and Statistics ◦ University of Maryland Baltimore County, MD

Abstract

In order to process large volumes of network traffic data and quickly detect intrusions, we use parallel computation frameworks for Network Intrusion Detection Systems (NIDS). Additionally, we model network data as graphs to highlight the interconnected nature of network traffic and apply unsupervised graph-based clustering to flag anomalies as network intrusions. We particularly examine the effectiveness of barycentric clustering on graph network traffic models for intrusion detection. The clustering algorithms have been implemented in Hadoop MapReduce for the purpose of rapid intrusion detection. Furthermore, k-nearest neighbor graphs (kNNGs) are used to optimize the clustering process. We find that across various data sets, unsupervised graph based clustering is able to exceed a 92% intrusion detection accuracy and that kNNGs can effectively optimize the network traffic graphs for larger data sets.

Data and Graph Creation

Network Data

- Data sets are obtained as subsets of the KDD 1999 cups dataset.
- Data sets are filtered to only contain network connection where a login was completed.

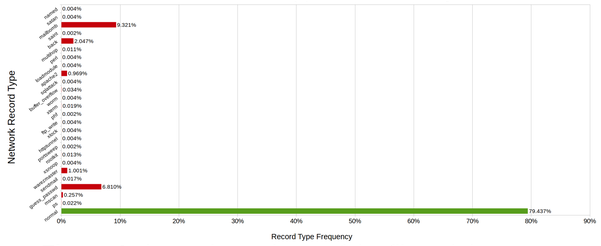


Figure 1 shows the makeup of the filtered data set.

Similarity Graph Creation

- Edge weights are calculated by:

$$w_{i,j} = \frac{1}{\exp\left(\left\|\vec{v}_i - \vec{v}_j\right\|\right)}$$

where $\vec{v}_x$ is a data vector associated with the x^{th} network record.

Graph Clustering & Intrusion Classification

Barycentric Clustering

- Graphs of network records can be clustered using Barycentric Clustering implemented in Apache Hadoop's MapReduce framework. [1]
- Barycentric clustering forms the clusters using the following process:
  1) Assigning random positions to each graph vertex.
  2) Iteratively adjusts each vertex's position based on the weights of all adjacent edges.
  3) Edges of longer than the average length are cut.
  4) Repeat.
- Barycentric Clustering requires $O(E)$ time to run, i.e. time proportional to the number of edges [1].

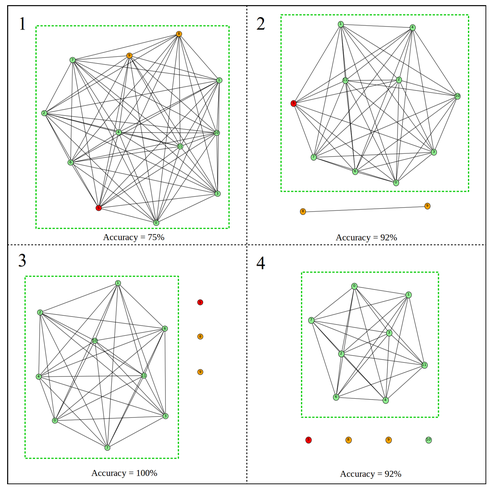
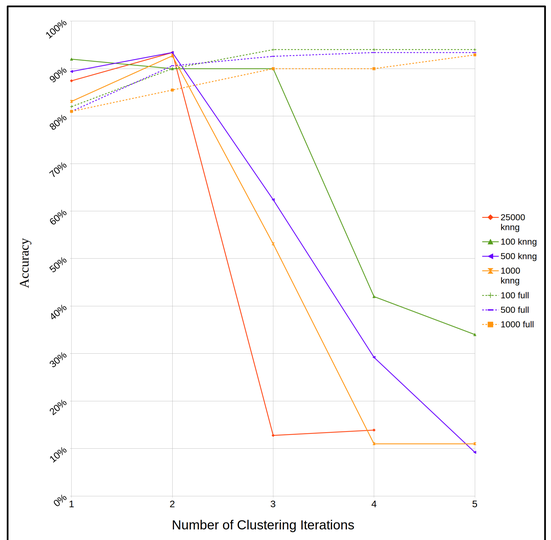


Figure 2 depicts 4 subsequent iterations of the barycentric clustering and intrusion classification process. The data set contains 9 normal records (green), 2 mailbomb intrusions (orange), and 1 rootkit intrusion (red).



Data Set Size	100	500	1,000	25,000
kNNG Accuracy @ 2nd Iteration	90%	93.4%	92.7%	93.4%
Full Graph Accuracy @ 5th Iteration	94%	93.4%	92.9%	N/A

Figure 3 shows detection accuracy at various iterations across data sets. k=50

Intrusion Classification

We classify every network record vertex within the largest connected component to be non-intrusive, and classifying all other network record vertices as intrusive.

Detection Accuracy

Detection accuracy is calculated as:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- TP is the number of true positives,
- FN is the number false negatives, etc.

K-Nearest Neighbor Graphs

- K-Nearest Neighbor Graphs (kNNGs) can be used to optimize the running time of the clustering algorithm instead of using complete graphs.
- KNNG's keep the edges that affect the clustering results the most while eliminating the edges of smallest weight.
- Throughout all experiments a k value of 50 was used.

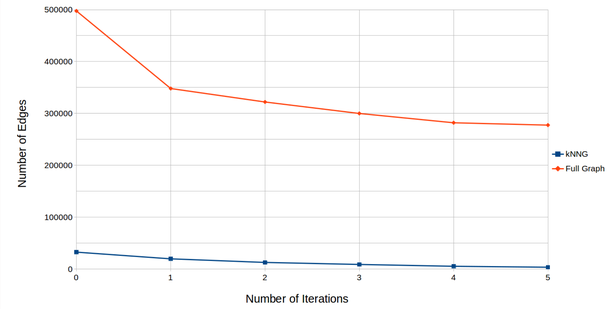


Figure 4 shows the number of edges after various iterations of barycentric clustering for both full graphs and kNNGs. The data used contained 1,000 network records. k = 50

Conclusion & Future Work

We find that graph-based clustering algorithms serve as an effective method for unsupervised network intrusion detection with detection accuracy consistently above 92%. Additionally graph creation and graph clustering can be distributed via Hadoop MapReduce and optimized with kNNGs in order to quickly detect network intrusions. KNNG optimizations are very effective for large data sets.

Future work includes determining how running time scales with Hadoop cluster size.

References

[1] J.D. Cohen, "Barycentric Graph Clustering," 2008; http://www2.computer.org/cms/Computer.org/dl/mags/cs/2009/04/extras/msp2009040029s2.pdf

[2] KDD data set, 1999; http://kdd.ics.uci.edu/databases/-kddcup99/kddcup99.html

Acknowledgement: This work is funded by the National Science Foundation under the Research Experience for Undergraduates Program CCF-1460900 grant.