Blake Conrad

CSCI 43200 – Security in Computing – Final Project

Network Security Vulnerability Detection by Cluster Analysis

Abstract:

In this work, a historical approach is taken better understand network security vulnerability detection. This is done by looking at how both unsupervised and supervised learning methods have been implored in diverse situations throughout their existence in the area of network traffic. This idea is scaled by looking at many different techniques and how each of those may be useful in detecting network vulnerabilities. After surveying a cadre of techniques scientists and researchers have used in the area of network traffic, Blake takes a closer look at how vulnerability detection has been studied in recent years. With both data mining algorithms and techniques, alongside industry practices for network vulnerability detection studied, Blake undergoes a small reproducible simulation in hopes of seeing if data mining is a successful tool for detecting vulnerabilities in a network setting. The algorithm of choice for this simulation after much consideration was *K-means*, while the language of choice was R. The data set, the R code for the simulation, the HTML file used to house the images, and other example plots and code for the reproducible simulation can be obtained in the following link: https://github.com/conradbm/security/tree/master/final_project.

Introduction

The term cluster analysis has been coined by many researchers and is used in conversation extensively. The idea behind cluster analysis is to take a set of data and find the best set of partitions in it that represent it the best; we call these clusters. In the truest abstraction of the word *cluster* there exists no real definition, but when a domain is associated to it, it makes more sense. Some simple examples could be:

- If we have a data set containing three types of flowers, and we want to separate them the best based on their characteristics (Flower domain expertise would know this):
    - Desired number of clusters: three
- If we have a data set involving rocks and minerals and we knew that there existed 5 rocks and 3 minerals (Geology domain expertise would know this):
    - Desired number of clusters: eight
- If we have a data set involving crimes over time, and we knew that there were about 10 different patterns we saw in cities growing and shrinking in size (FBI or another crime organization would know this):
    - Desired number of clusters: 10

There are countless examples of how cluster analysis can be used to add feature variables to data set to allow for additional exploratory analysis, feature engineering, and even feature discovery. Determining the number of clusters in a data set is often a non-trivial task without domain expertise or the data set explicitly giving clues away about it (Example: 50% of the data is clearly one type of object type while the other is not). Top researchers Dr. Pham, Dimov, and Nguyen state that in their work *Selection of K in K-means clustering* that finding the appropriate number of clusters for a given data set is generally a trial-and-error process (Pham, Dimon, and Nguyen, 2004). As time has progressed, cluster analysis has shown itself useful in many domains, and network security has become one of many. Many researchers have endeavored to show how cluster analysis can be utilized to represent network traffic, detect network vulnerability, and even classify types of users based on port and packet flow. With network security and vulnerability becoming a hotter issue in these recent years, the hope is to now explore some of the ways cluster analysis can be implemented on different types of network data, what researchers have already done in this area, and how one could simulate some real data to find out whether the results make sense from the information security and cluster analysis scientific communities.

Historical Research: Network Traffic Analysis via Unsupervised Learning

Researchers Erman, Arlitt, and Mahanti from the University of Calgary have a research publishing by the name of *Traffic Classification Using Clustering Algorithms*. In the article, the authors make statements regarding the nature of network traffic and its involvement in how one can classify the data using clustering algorithms. They are noted saying:

> "Classification of network traffic using port-based or payload-based analysis is becoming increasingly difficult with many peer-to-peer (P2P) applications using dynamic port numbers, masquerading techniques, and encryption to avoid detection. An alternative approach is to classify traffic by exploiting the distinctive characteristics of applications when they communicate on a network. We pursue this latter approach and demonstrate how cluster analysis can be used to effectively identify groups of traffic that are similar using only transport layer statistics. **Our work considers two unsupervised clustering algorithms, namely K-Means and DBSCAN, that have previously not been used for network traffic classification**. We evaluate these two algorithms and compare them to the previously used AutoClass algorithm, using empirical Internet traces. The experimental results show that both K-Means and DBSCAN work very well and much more quickly then AutoClass. Our results indicate that although DBSCAN has lower accuracy compared to K-Means and AutoClass, DBSCAN produces better clusters."
>
> - Jeffrey Erman, Martin Arlitt, and Anirban Mahanti

As emphasized by the colored text, the researchers above have used *K-means* and *DBSCAN* to classify network traffic. Both algorithms are unsupervised techniques they used to take on the task of classifying traffic by exploiting distinctive characteristics. The findings from the article conclude that *DBSCAN* outperformed *K-means* in terms of clustering. This is a good piece of literature because we can learn from this specific type of network based clustering problem, the results obtained, and how this may affect our specific problem for network vulnerability detection. Another great group of literature on the subject of unsupervised learning for network detection has been written by researchers by the name of Yinggiu Liu, Wei Li, and Yun-Chun Li both together and independently. They each research at Beihang University, Beijing and produced a piece of work I want to note by the name of *Network Traffic Classification Using K-means Clustering*. The research findings on the same exact problem as Erman, Arlitt, and Mahanti are in the following passage from their work:

> "Considering the priority of the machine learning-based method, we experiment with unsupervised K-means to evaluate the efficiency and performance. We adopt feature selection to find an optimal feature set and log transformation to improve the accuracy. **The experimental results on different datasets convey that the method can obtain up to 80% overall accuracy, and, after a log transformation, the accuracy is improved to 90% or more."**
>
> - Yingqiu Liu, Wei Li, and Yun-Chun Li

The research shows that *K-means* is still a valid technique for particular types of problems in the domain of clustering problems, particularly it did a very good job to classifying network traffic for the researchers at Beihang University if it is obtaining

80% or more with log transformations. In the area of data mining and machine learning, if a model is build that classified unseen data at an 80% or higher rate the community tends to call this a *good model*. Considering the initial researcher at the University of Calgary, their position on DBSCAN over the K-means algorithm is an important one because now we want differing results as the problem changes. This type of conflict is nothing new in the data mining world; as problems evolve so do solutions. For the sake of our research to find the best unsupervised algorithm, which seems to be a tie between *DBSCAN* and *K-means*.

Historical Research: Network Traffic Analysis via Supervised Learning

By looking further still into the area of classification algorithms for network traffic, authors R. Deebalakshmi and V. L. Jyothi from Sathyabama University, Chennai published an article entitled *A survey of classification algorithms for network traffic*. With the title sounding similar to previous work on network traffic, these researchers actually explore both supervised and unsupervised learning techniques on network traffic. This research survey shows different classification algorithms contrasted with one another, namely: K-means algorithm, Classification based on Fuzzy Kernel K-means Clustering, Support Vector Machine algorithm, and Self-Learning Classifier Bayesian Classification. (Deebalakshmi, Jyothi 2016). Another group of researchers, Modiuddin Ahmed and Abdun Naser Mahmood have a paper entitled *Novel Approach for Network Traffic Pattern Analysis using Clustering-based Collective Anomaly Detection*. This work goes into depth about a very popular attack vector in the security world; *DDOS*. The authors state introductory thoughts about their analysis in the following:

> "In this paper, we formulate the problem of detecting DoS attacks as a collective anomaly which is a pattern in the data when a group of similar data instances behave anomalously with respect to the entire dataset. We propose a framework for collective anomaly detection using a partitioned clustering technique to detect anomalies based on an empirical analysis of an attack's characteristics. We validate our approach by comparing its results with those from existing techniques using benchmark datasets."
>
> - Mohiuddin Ahmed and Abdun Naser Mahmood

Ahmed and Mahmood highlight some key information in section 4.2 of their publishing, by stating their algorithms were density based, involvement near cluster centroid distances, and that they used thresholds to consider anomalies:

> "Since the goal of clustering is to group similar data, it can be used to detect anomalous patterns in a dataset. The following three key assumptions are made when using clustering to detect anomalies. Premise 1: if we create clusters of only normal data, any subsequent new data that do not fit well with the existing clusters are considered anomalies; for example, as **density-based clustering algorithms** ...

> Premise 2:it has been found that, when a cluster contains both normal and anomalous data, the normal data lie close to the nearest clusters centroid but the anomalies are far away from the centroids …
> Premise 3: in a clustering which has clusters of various sizes, the smaller and sparser ones can be considered anomalous and the thicker ones normal. Instances belonging to clusters the sizes and/or densities of **which are below a threshold are considered anomalous.**"
>
> - Mohiuddin Ahmed and Abdun Naser Mahmood

Noting that the algorithms main attributes are density based clustering, minimizing near cluster centroid error, and determining if a point is outside of a threshold, lets me know that there is evidence to believe that *K*-means may be more affective than the *DBSCAN* algorithm. By considering each researchers work, we can weight out which algorithm will aid our research goals the best in our independent study of network vulnerability detection.

Research in Network Vulnerability Detection

There has been several professors and published works that have attempted something a little more close to what I am trying to accomplish. One of several examples is a few inventors from *CISCO Technology, Inc.* that bound together an innovative solution to using packet analysis for an adaptive style of network intrusion detection. As noted in their work,

> "The method comprises monitoring network data traffic. The network data traffic is analyzed to assess network information. A plurality of analysis tasks are prioritized based upon the network information. The analysis tasks are to be performed on the monitored network data traffic in order to identify attacks upon the network."
>
> - Robert E. Gleichauf, Daniel M. Teal, Kevin L. Wiley
> *CISCO Technology, Inc.*

As one can see, this is not a new desire for a dynamic type of system that can detect network vulnerabilities. *CISCO Technology, Inc.* amongst many other companies have sought to fund such research for their own systems. The chief principle pertaining to how this architecture is set up I got from the technology invented was the following diagram from their publishing:

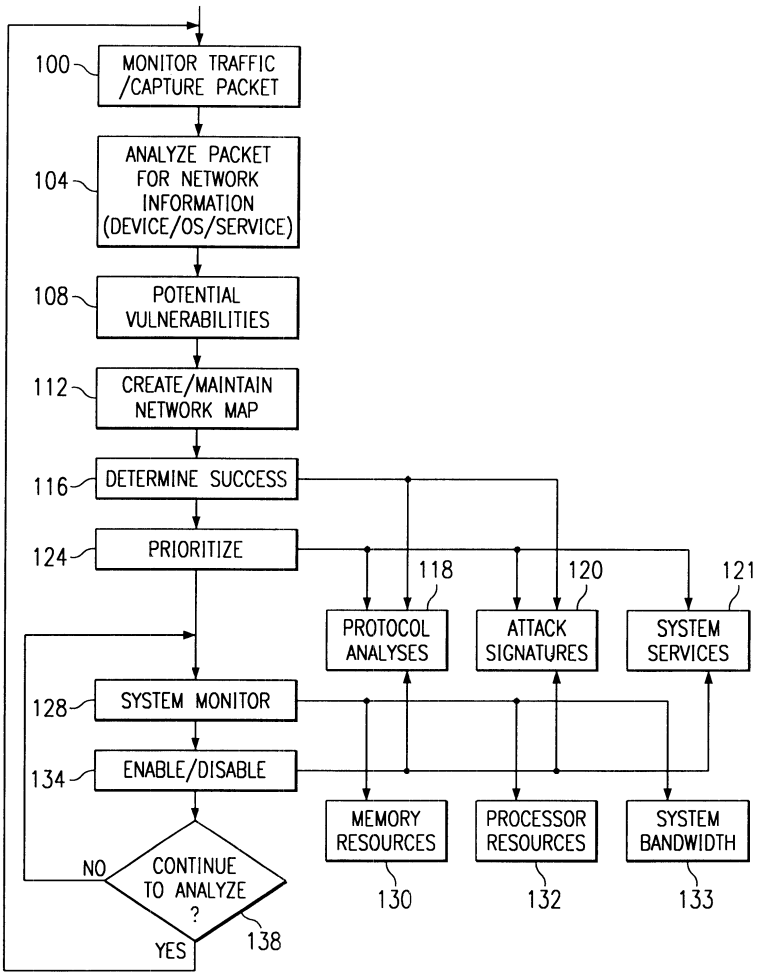


Figure from https://www-google-com.proxy.ulib.uits.iu.edu/patents/US6499107

An image from *CISCO Technology, Inc.* to illustrate my main thoughts captured on the work.

One can easily see the inventors attempt to make this explicitly clear through the visualization above. In the phase labels 100 and 104, we see the data coming in and being analyzed; the actual packets. The interesting part is in phase 108, which states where we check for vulnerabilities. I found that this is the area in which my personal simulations I will perform later in this work would come in most benefit. By understanding phase 108 better and exploring the use of unsupervised or supervised learning approaches, I can try to see what might be most effective toward an aftermath situation. Though, my research intent is not *dynamic detection* as the publishing is, it is detection nonetheless due to the ability to learn from the past which is a valuable lesson in all aspects of life; network security no less. Another work that does a pretty good job at aiming to learn from the past is entitled *Large-Scale Network Monitoring for Visual Analysis of Attacks*, which suggests that by visualizing our network traffic one can actually detect these types of vulnerabilities we have been discussing (Fischer, Mansmann, Keim, Pietzko, Waldvogel, 2008).

Simulating Network Security using R

By taking a historical look at unsupervised learning, cluster analysis, and classifying network traffic data, I found that the K-means clustering algorithm is a both relevant and useful tool at classifying traffic data successfully. Knowing that traffic data had success with it, alongside density based research, I am lead to conclude that it will also be a good metric for network vulnerability detection. I acquired an open source data set which included some security labels on each data point. This allowed me to have a metric of how well the K-means will do on the data set. With a level of security given, the main interest is:

- How well does K-means do on the data set at predicting?
- What is the network security's best label?
  - Binary: Secure or not (e.g., 0 or 1, with 1 as secure)
  - Ranked: Levels of security (e.g., 0-9, with 9 being the best)

The data retrieved was in the *.csv* format, so I brought in the data into my programming language of choice: *R*. After analyzing the data, I found that the data set had about 37 different features (1 of which I added on later after doing the cluster analysis). *Figure 1* shows us what features are included in the data set:

Figure 1:
An overview at all of the variables provided by the open source data set.

Analysis Goals

The main technique that I will explore to determine how well the *K-means* is going to predict will be the *Root Mean Squared Error (RSME)* metric. This will be done by taking a look at how well the *K-means* did at the following:

- Security as a binary classifier, even though the data set did not give it as such:
  - K=2
- Security as a categorical classifier, as provided:
  - K=10

Exploring the data

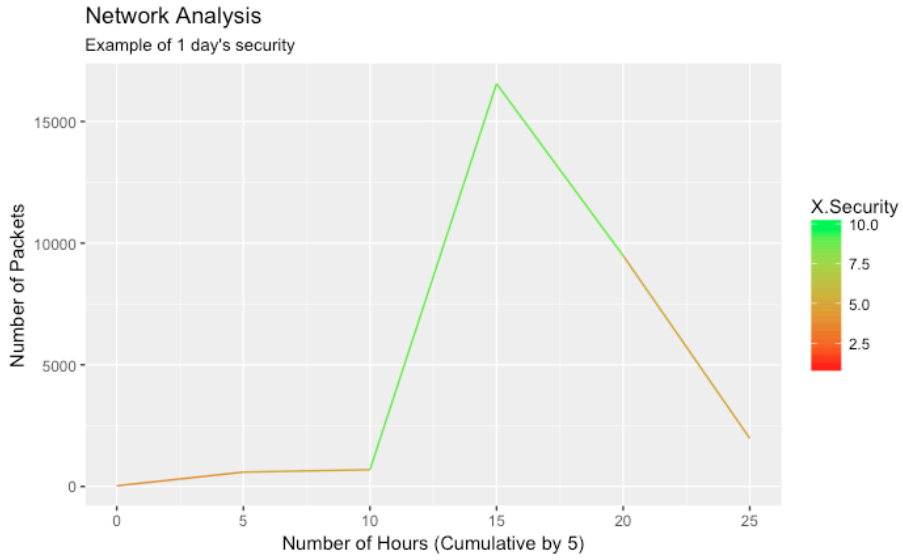
Next, *Figure* 2 shows a snapshot of just a small sample of our data set. The sample of our data is read in and plotted. This shows some network flow for 1 day and the security as given by the data set.

Figure 2
A sample of our dataset to show one days network traffic and associated network security.

One day's analysis as an example plot

Hide

```
plt1 <- ggplot() + geom_line(data=table[1:6,c("Starttime","X.Packets","X.Security")], aes(x=Starttime, y=X.Packet
s,color=X.Security))
plt1 <- plt1 + ggtitle("Network Analysis", subtitle="Example of 1 day's security")
plt1 <- plt1 + xlab("Number of Hours (Cumulative by 5)") + ylab("Number of Packets")
plt1 <- plt1 + scale_colour_gradient(limits=c(1, 10), low="red", high="green")
plt1
```



Next, we can actually plot our data set entirely through just a few short lines of code in R. The plot to illustrate our entire data set, and its associated network security is given in *Figure 3*. By examining *Figure 3*, we can see that in peak traffic we get higher ranked clusters, which tells us security gets worse.

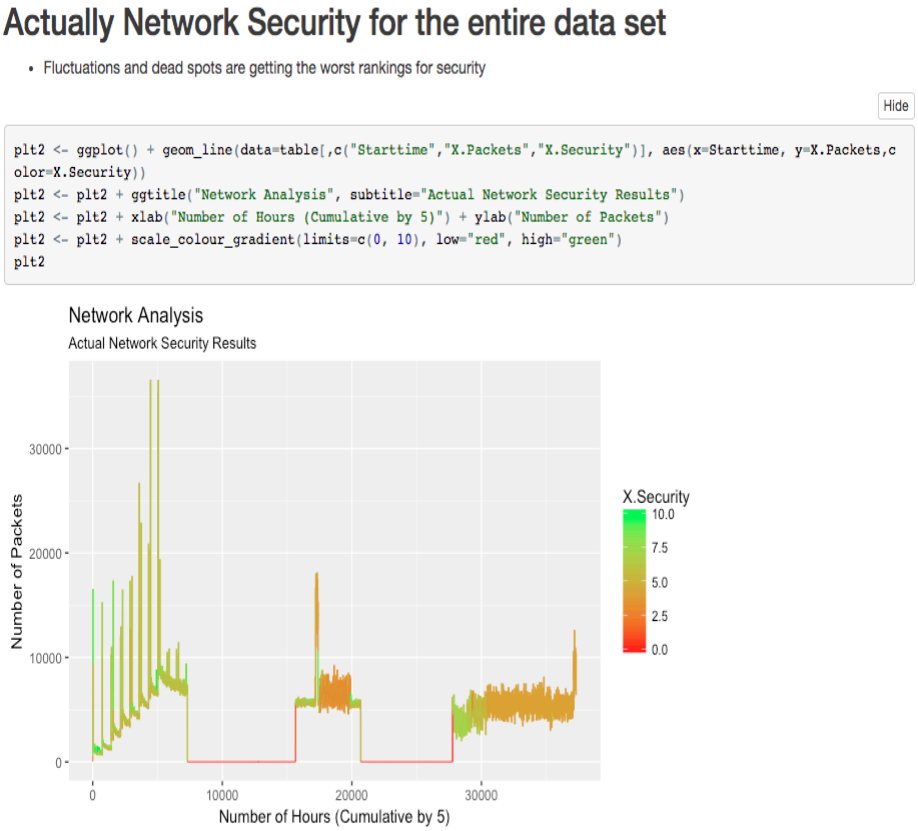


Figure 3

A high level overview of the entire data set and network security associated with it.

Exploring the K-means Results

As is not uncommon in network traffic data sets, we get flows. These are clearly indicated in *Figure 3* by seeing the hard jitters that are close by each other over time. There are times where the traffic is low and when the traffic is very high. By seeing the high jitter areas, we can see that the worst security is worse (which intuitively makes sense, as attackers like to blend in with normal traffic). By applying K-means to the data at K=10, we can attempt to try to simulate the type of labels that were given and determine how accurate our algorithm was. *Figure 4* shows the simulation results of K-means vs. the data given.

Figure 4

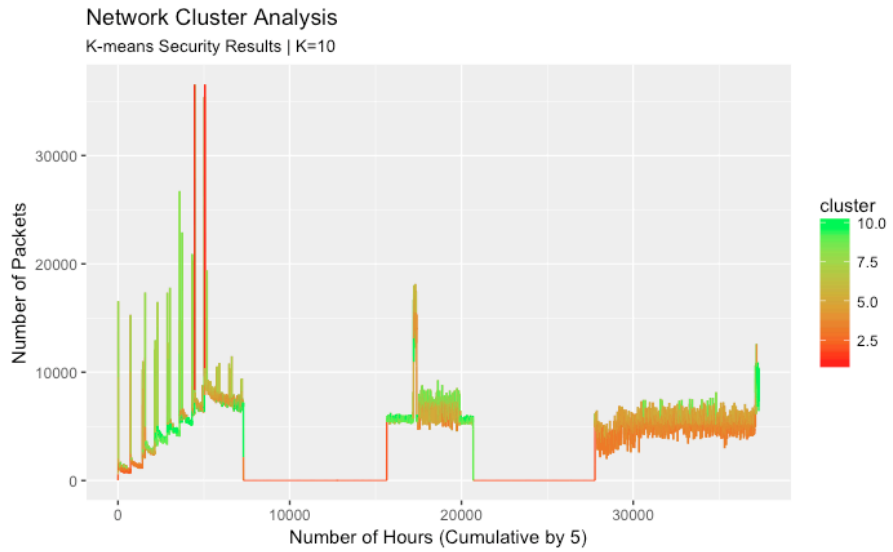
Simulation of K-means at K=10 to see how close it was to our original data set labels

K-means Results

- At K=10
- The initial security vulnerabilites were captured
- The dead spot vulnerabilities were undermined as not as bad as they should be considered
- The latter security vulnerabilities were for the most part also captured consistently

Hide

```
plt4 <- ggplot() + geom_line(data=table[,c("Starttime","X.Packets","cluster")], aes(x=Starttime, y=X.Packets,colo
r=cluster))
plt4 <- plt4 + ggtitle("Network Cluster Analysis", subtitle="K-means Security Results | K=10")
plt4 <- plt4 + xlab("Number of Hours (Cumulative by 5)") + ylab("Number of Packets")
plt4 <- plt4 + scale_colour_gradient(limits=c(1, 10), low="red", high="green")
plt4
```



By looking at *Figure 4* and *Figure 3* side by side, we really don't see too much of a difference just visually outside of a couple clear distinctions. It seems that the K-means has done a pretty good job at predicting the label at K=10. With some minor exceptions in the center area around 20,000 hours and the final area around 30,000 hours we can see some changes that our model predicted incorrectly. The real question up to this point is, how well did it do? My hope is to answer this question by looking at how well the *K-means* at K=2 did. By looking at *Figure 5* we can see just how well K=2 did. This definitely looks different than our initial plot, but just how different? At first glance it seems like K=2 actually did worse than K=10 because we don't see the *heat map* ranking visually. However, after computing our error metric (Root Mean Squared Error) in *Figure 6*, we can see that at K=10 our RSME was 4.39 and at K=2 our RSME was 3.17.

Figure 5

Simulation of K-means at K=2 to see how close it was to our original data set labels

K-means Results

- At K=2
- The initial security vulnerabilities weren't captured
- The dead spots and latter more severse secure vulnerabilities were captured

Hide

```
plt3 <- ggplot() + geom_line(data=table[,c("Starttime","X.Packets","cluster")], aes(x=Starttime, y=X.Packets,colo
r=cluster))
plt3 <- plt3 + ggtitle("Network Cluster Analysis", subtitle="K-means Security Results | K=2")
plt3 <- plt3 + xlab("Number of Hours (Cumulative by 5)") + ylab("Number of Packets")
plt3 <- plt3 + scale_colour_gradient(limits=c(1, 2), low="red", high="green")
plt3
```

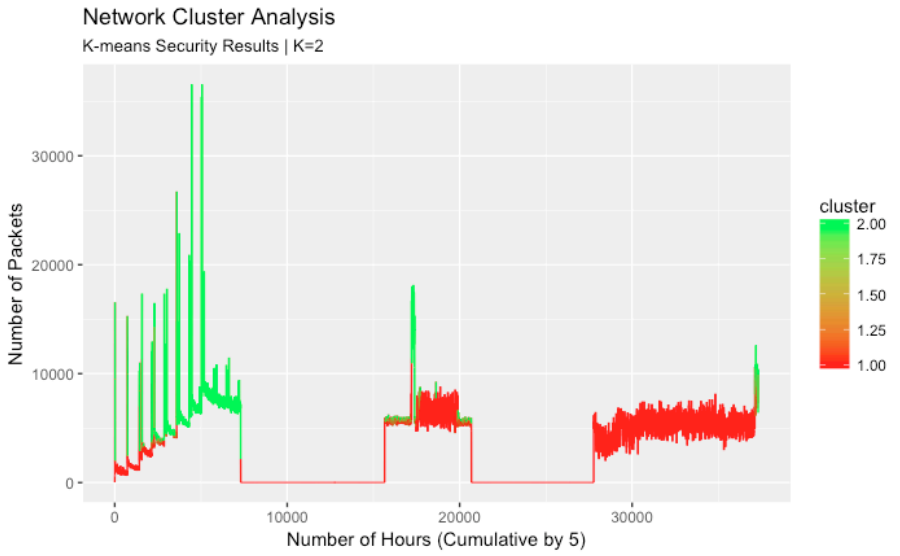


Figure 6

A code snippet to show how well K=2 and K=10 did on the data set at predicting.

How wrong was the K-means algorithm at predicting network security?

- At K=2 –> 3.171672
- At K=10 –> 4.392572

Hide

```
MSE <- sum( (table$X.Security - table$cluster)^2 )/length(table$X.Security)
RMSE <- sqrt(MSE)
RMSE
```

Conclusion

After taking an extensive look at historical research on network traffic, exploring a multitude of supervised and unsupervised algorithms, taking note of what algorithms perform best and worst under different situations, and taking on some particular simulations and problems in network vulnerability detecting, we can see how the results vary. More research measures were discussed by more particular solutions to the problem on exactly how vulnerability is detectable from a systems standpoint. In final remarks we undertook a simulation to attempt to see how well the *K-means* algorithm actually does in detecting network vulnerability. We did this by taking a look at the dataset which had original labels from 0 to 9. We then trained our *K-means* predictive model to try to detect the vulnerability label with the hyper-parameter *K* set to ten and two. We then took a look at several visualizations which showed our error metric (*RSME*) was actually better for K=2 then K=10 even though the data set provided was in the form of 10 labels. This analysis could be used to leverage many different types of problems associated with network security, for example if a new observation of a network over time were cast into our model, we could have the ability to predict with high probability a similar result to those simulated. By expanding on this work we could potentially touch many different areas of network security via supervised or unsupervised learning with the *K-means* clustering algorithm.

Works Cited

Ahmed, M., Mahmood, A., N. Novel Approach for Network Traffic Pattern Analysis using Clustering-based Collective Anomaly Detection. 14 May 2015. **DOI:** 10.1007/s40745-015-0035-y

Dimov, S., S. Retrieved from: https://www.ee.columbia.edu/~dpwe/papers/PhamDN05-kmeans.pdf. **DOI:** 10.1243/095440605X8298

Erman, J., Arlitt, M., Mahanti, A. Retrieved from: https://pdfs.semanticscholar.org/c9d6/2335443a48a18cbb6f532c4362fd04f6010a.pdf

Fischer F., Mansmann F., Keim D.A., Pietzko S., Waldvogel M. (2008) Large-Scale Network Monitoring for Visual Analysis of Attacks. In: Goodall J.R., Conti G., Ma KL. (eds) Visualization for Computer Security. Lecture Notes in Computer Science, vol 5210. Springer, Berlin, Heidelberg

Gleichauf E., Robert, Teal M., Daniel, Wiley, L., Kevin. Method and system for adaptive network security using intelligent packet analysis. Dec 24, 2002. CISCO Technology, Inc. US6499107 B1. *Google Scholar*. Retrived from: https://www-google-com.proxy.ulib.uits.iu.edu/patents/US6499107

Liu, Y., Li, W., Li, Y. Network Traffic Classification Using K-means Clustering. Computer and Computational Sciences, 2007. IMSCCS 2007. Second International Multi-Symposium. 13-15 Aug. 2007. IEEE. **DOI:** 10.1109/IMSCCS.2007.52

R. Deebalakshmi, V. L. Jyothi. A survey of classification algorithms for network traffic. Science Technology Engineering and Management (ICONSTEM), Second International Conference. 30-31 March 2016. IEEE. **DOI:** 10.1109/ICONSTEM.2016.7560941