Andrew Bajumpaa
CIS 602 Project

Real Time Anomaly Detection with PCA

Abstract
This project implements a PCA to transform selected fields from live network traffic to three dimensional space, and estimates anomalous traffic by its distances from the mean expected variance of baseline traffic, as detailed in *Shyu et al. (2003)*. This approach is particularly useful for real time analysis because it is very easy to compute, and does not require full knowledge of the training set to begin making assumptions of anomalous behavior. As an example of the utility of this, the data in the proof of concept was generated while training the model. Each time stamp represents the capture of only 100 packets, meaning results can be generated quickly and with extremely little data.

Data Preprocessing and Data Limitations

For this implementation, I only capture TCP and UDP packets. This isn't a limitation of the method, but rather of my patience. Certain key fields are extracted from the packet, including the name of the protocol being used, ttl, IP flags, source and destination ports, urgptr, TCP flags, and whether certain commonly used layers under the transport layer exist within the packet. Notably, any field containing text data, or highly variable data like IP addresses is ignored. This is a limitation of PCA as a method of distilling useful information. The high variability of IP or hashed string data will severely overwhelm the representation of any other useful data once it has been transformed, negating the effects of other potentially more useful information in the final representation, and producing extremely sparse, linear clusters as shown in *Figure 1*. The metric used to determine if a packet is anomalous relies on roughly spherical distributions of variance in the final transformation, so linear artifacts are undesirable. When discarding any extremely variable data such as IP or text information, more desirable clusters are generated, as shown in *Figure 2*. While discarding highly variable data can still produce some linear clustering, the effect is significantly less pronounced than when using the highly variable data.

Figure 1

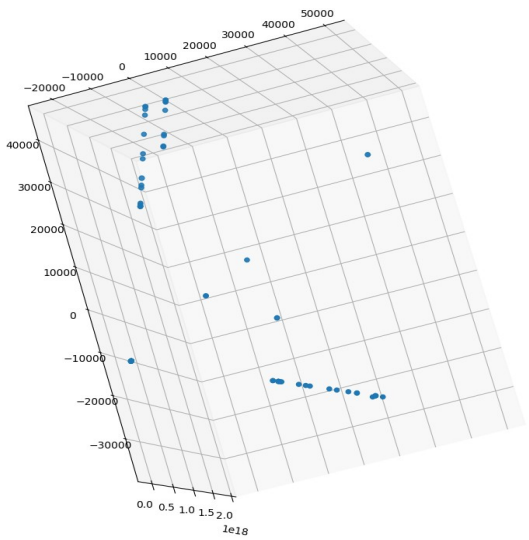
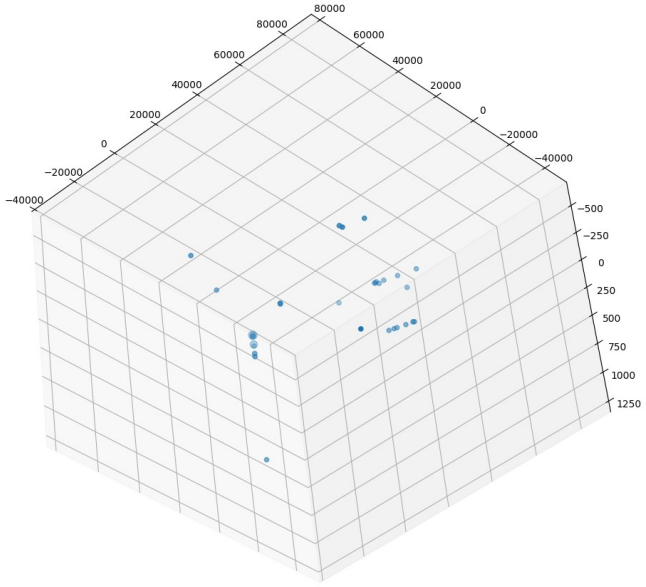


Figure 2



Anomaly Metric

As described by Shyu et al. a metric equivalent to the Mahalanobis distance is used to detect anomalies. A threshold value, *c* is used, where if the distance of a transformation exceeds *c*, that packet is classed as anomalous. Shyu et al. continue to describe what threshold value to use given some desired rate of false positives/false negatives, however, this is not implemented in this project.

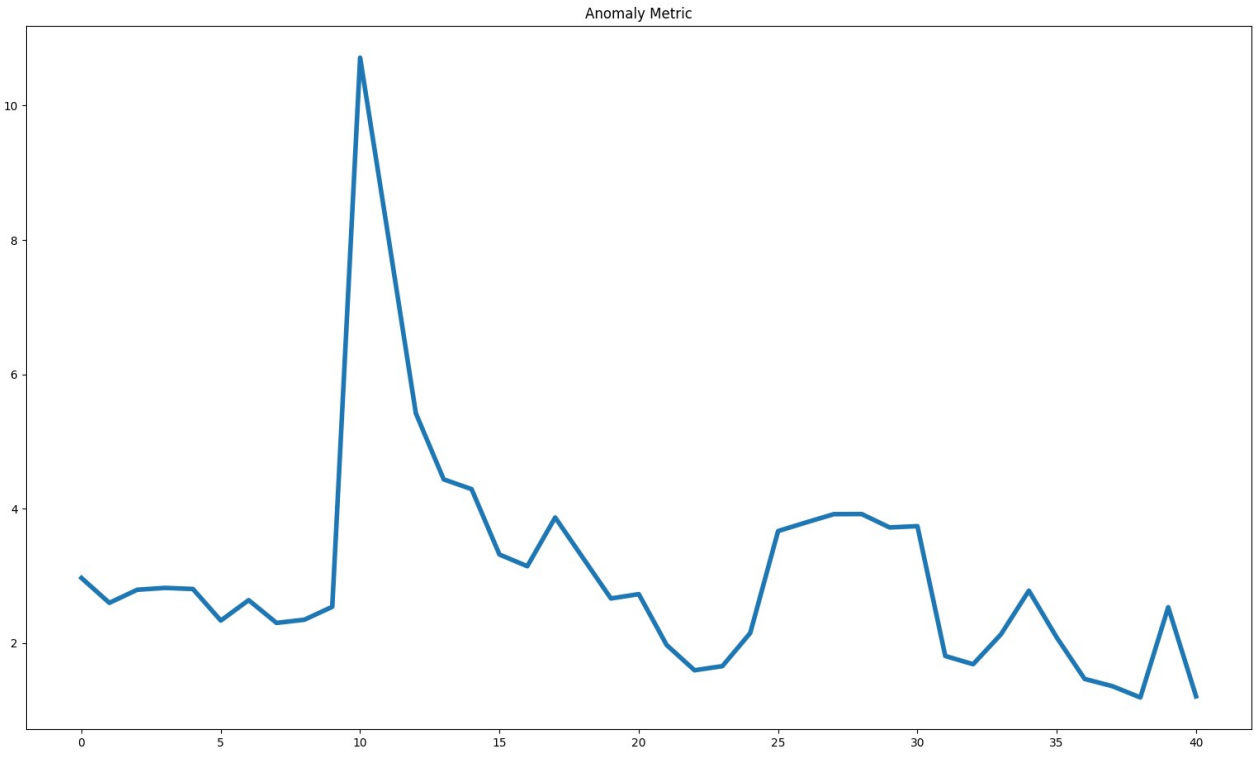
Equation 1: Anomaly if true

$$\sum_{i=1}^{p} \frac{y_i^2}{\lambda_i} > c$$

Where y_i is an observed prediction of the i^{th} field, and λ_i is the explained variance of that field within the model

Proof of concept

Figure 3



The proof of concept methodology is as follows, and the results are plotted in *Figure 3*. The procedure starts by initializing a new PCA at time stamp 0. The first 10 time stamps are collecting baseline network activity to train the new model. This baseline activity starts with very low scores on the anomaly metric. At the 10^{th} time stamp, a new type of network activity is initiated resulting in a large spike in the anomaly metric, which quickly lowers down in the following time steps as the model learns to expect this kind of traffic. At time step 20, the new network traffic is halted. At time stamp 25, the same traffic as before is initiated, this time with a significantly lower score than previous, before returning to baseline traffic. This indicates that the model is capable of learning and reacting to changes in network traffic and that the metric used can detect sudden changes in network activity. An appropriate threshold on the metric should be selected to determine when to report anomalous traffic. Both the training and anomaly detection occurred in real time.

In actual use, the training phases and anomaly detection phases are separated. Combining of the two is only used to demonstrate viability of the method.

Results
After training the model for a full 2 days on baseline network traffic, and setting the outlier threshold to some reasonably high value – 10 – the model successfully detects new kinds of network traffic that was not experienced during training and flags it as anomalous. However, it has a tendency to over-report by some amount. A more selective method for detecting outliers could track trends in the outlier metric over time, rather than flagging every individual packet that exceeds the threshold. This method would likely have a significantly lower false positive rate, but may reduce the true positive rate of anomalous behavior if surrounded by sufficiently large amounts of non-anomalous behavior.

Project Repository
https://github.com/racoltdev/PCA-Anomoly-Detection

User manual
https://github.com/racoltdev/PCA-Anomoly-Detection/blob/master/README.md

References
Shyu M-L, Chen S-C, Sarinnapakorn K, Chang L. A novel anomaly detection scheme based on principal component classifier. In: IEEE foundations and new directions of data mining workshop, in conjunction with ICDM'03, 2003. p. 171–9.

as referred to by Ahmed, M., Naser Mahmood, A., & Hu, J. (2016). A survey of Network Anomaly Detection Techniques. *Journal of Network and Computer Applications*, *60*, 19–31. https://doi.org/10.1016/j.jnca.2015.11.016