Thinkful Data Science Capstone

A Treatise in Unsupervised Learning with
Anomaly Detection Methods
In CyberSecurity
By Kimberly Nowell-Berry

Agenda

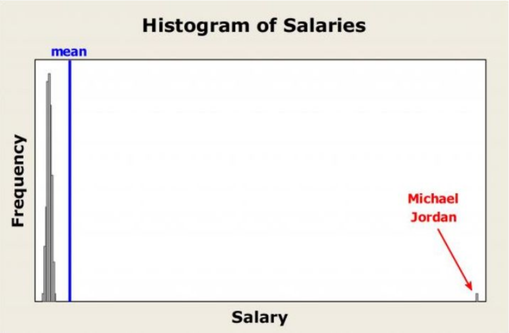
- Introduction
- Background
- Data
- Approach
  - Kalman Filters
  - Time Series
  - One Class SVM
- Conclusion
- Real World Applications
- Future Work

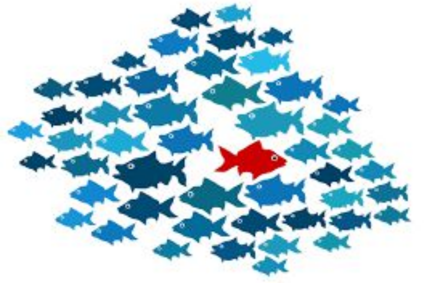
Problem Statement

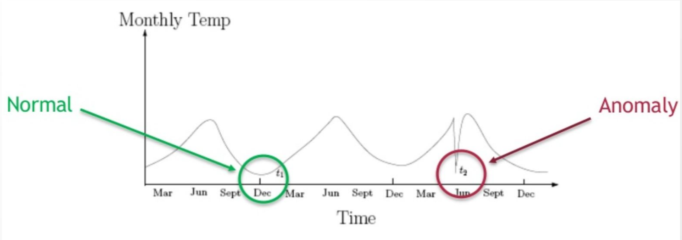
- Detecting Malicious Activity in corporations today is a real-world problem
- Determining where this malicious activity occurs often requires anomaly detection, where abnormal behavior can be evaluated by Analysts/Subject Matter Experts.
- Anomaly Detection models must strive to minimize False Positives because model output require Subject Matter Expert evaluation.

Background

- Anomaly Types
  - Point – events are anomalous with respect to all other events
  - Collective – events are anomalous with respect to adjacent events (time series consecutive events are different compared to other collections within the data set)
  - Contextual – Events that are anomalous with respect to context of additional information







Data

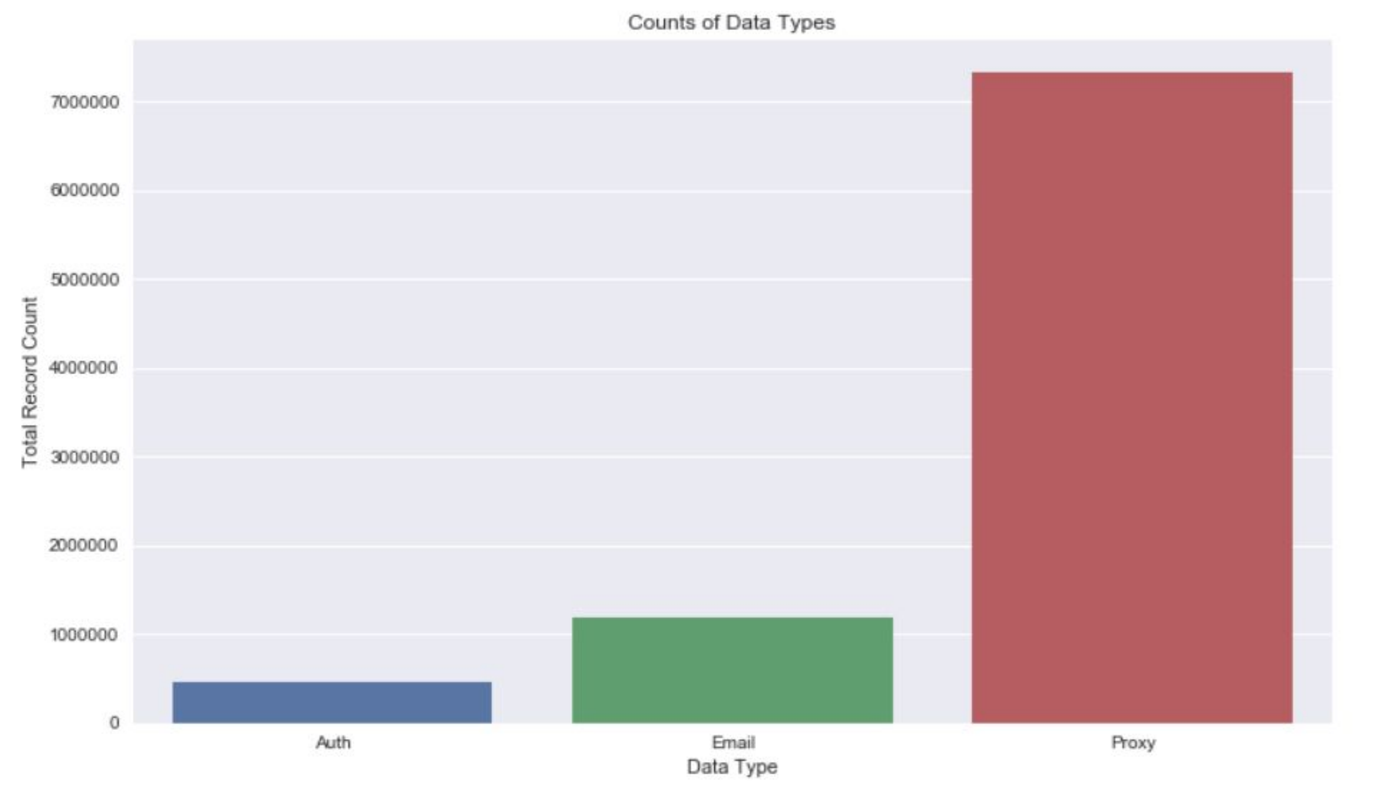
- Insider Threat Test Dataset
- Available from Carnegie Mellon University, Software Engineering Institute
- https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=508099
- Multi-modal data sources include:
  - Email Log data (similar to Exchange data)
  - HTTP Log data (similar to proxy log data)
  - Authentication Log data (similar to Windows Event Logs)

Data Summary

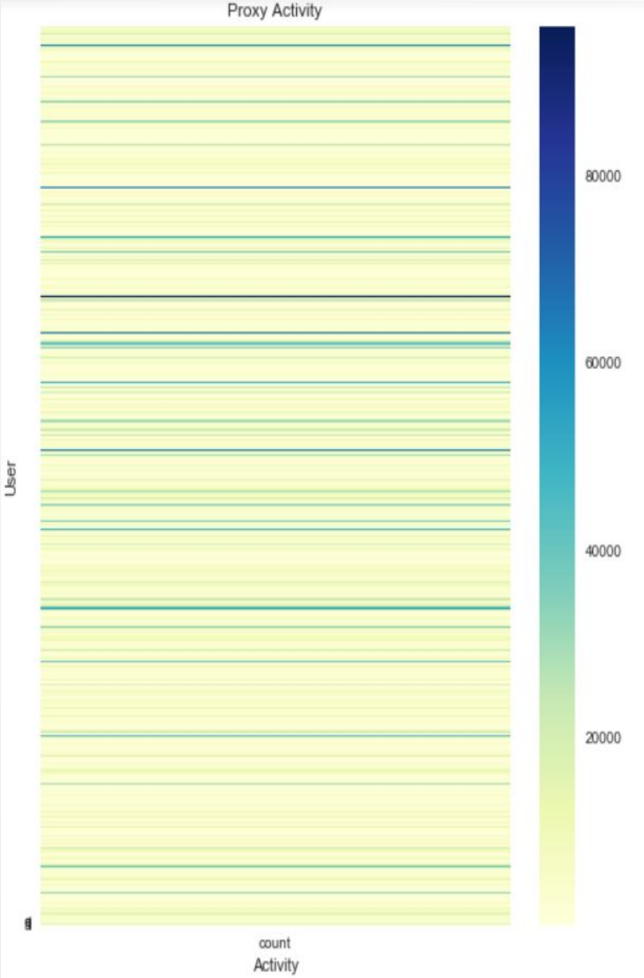
- For each data set, Email, Proxy, and Authentication, the total amount of users was 1000.

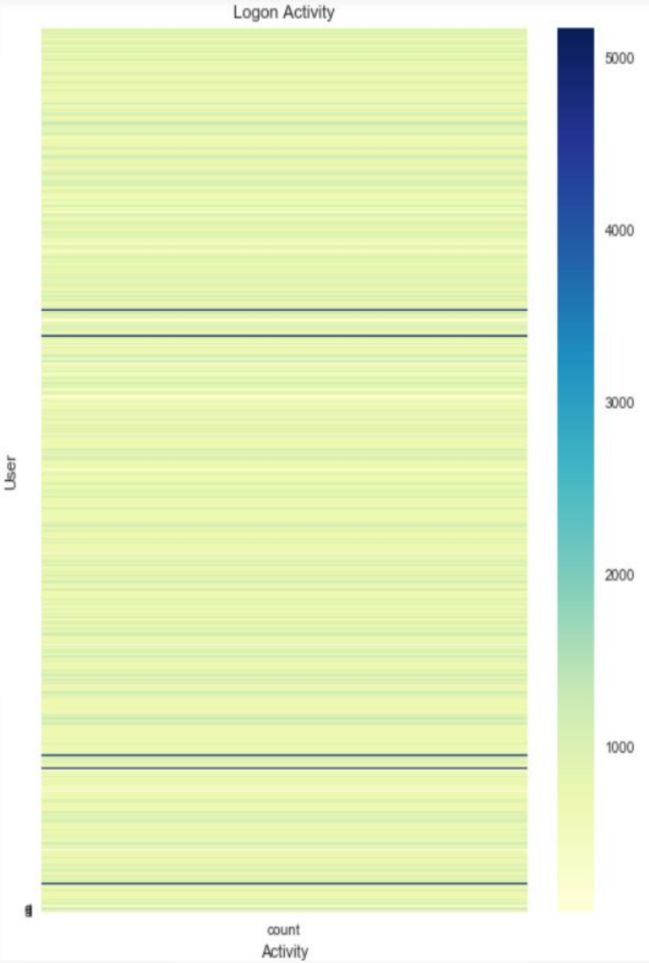
- All data types had user and computer information and unique Identifiers for each record.

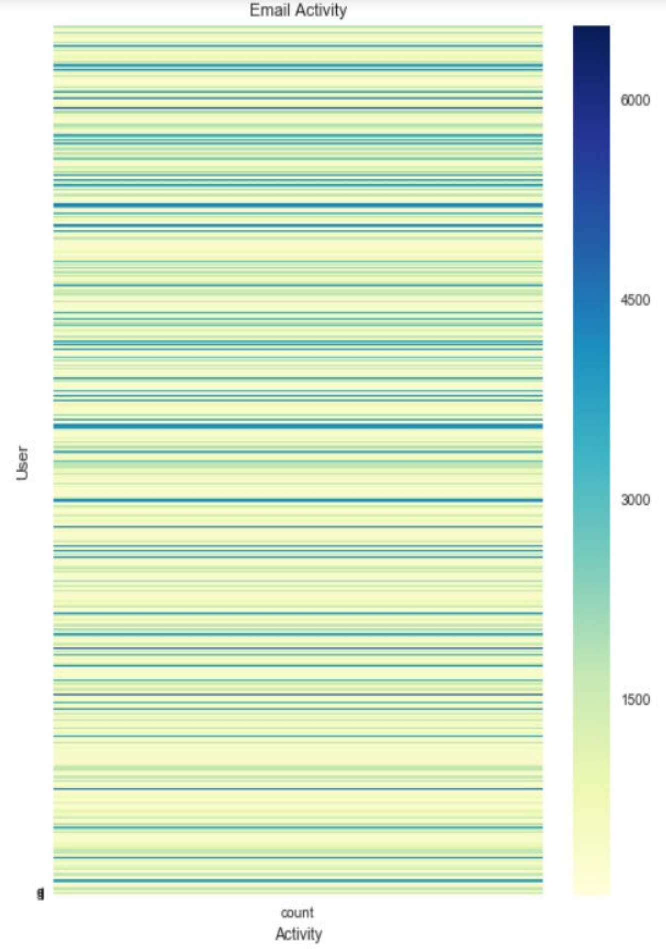
- Date range:4 month period



Data Summary





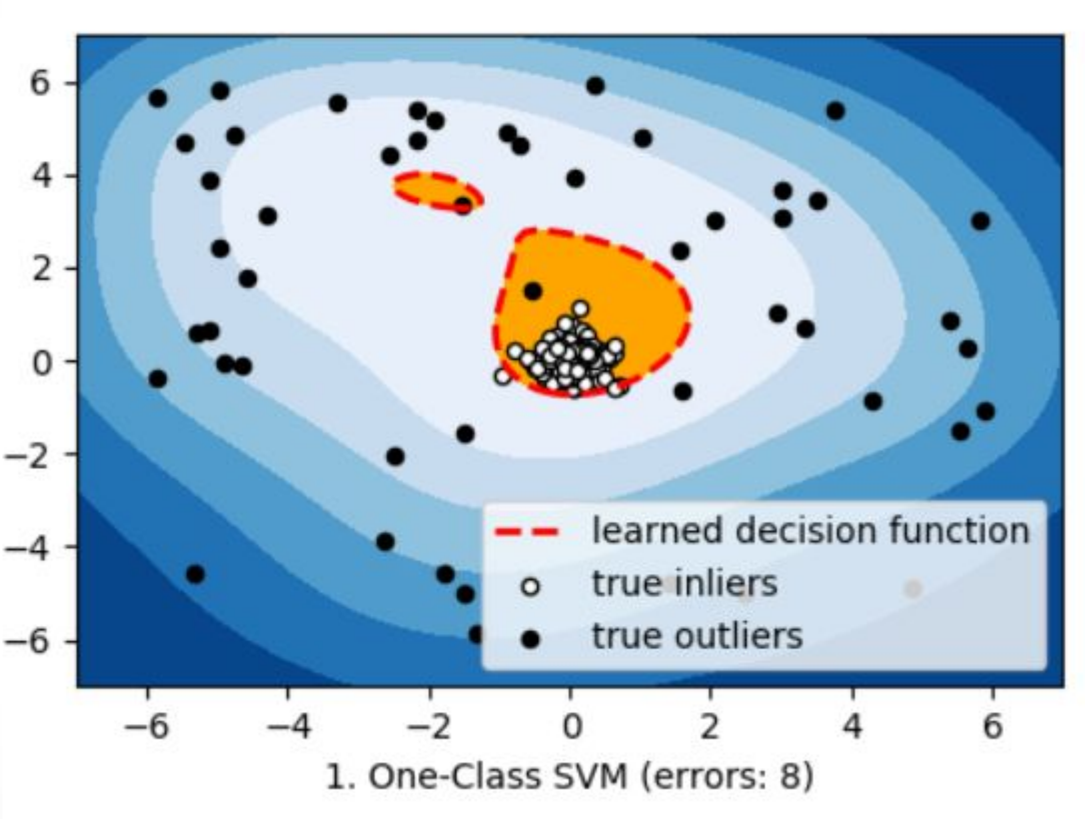


Approaches

- Multi-modal Anomaly Detection
  - Unsupervised Learning
  - Multi-source information
    - Email
    - Proxy
    - Authentication
- Techniques
  - Time Series Analysis
  - Kalman Filters
  - OneClass Support Vector Machine

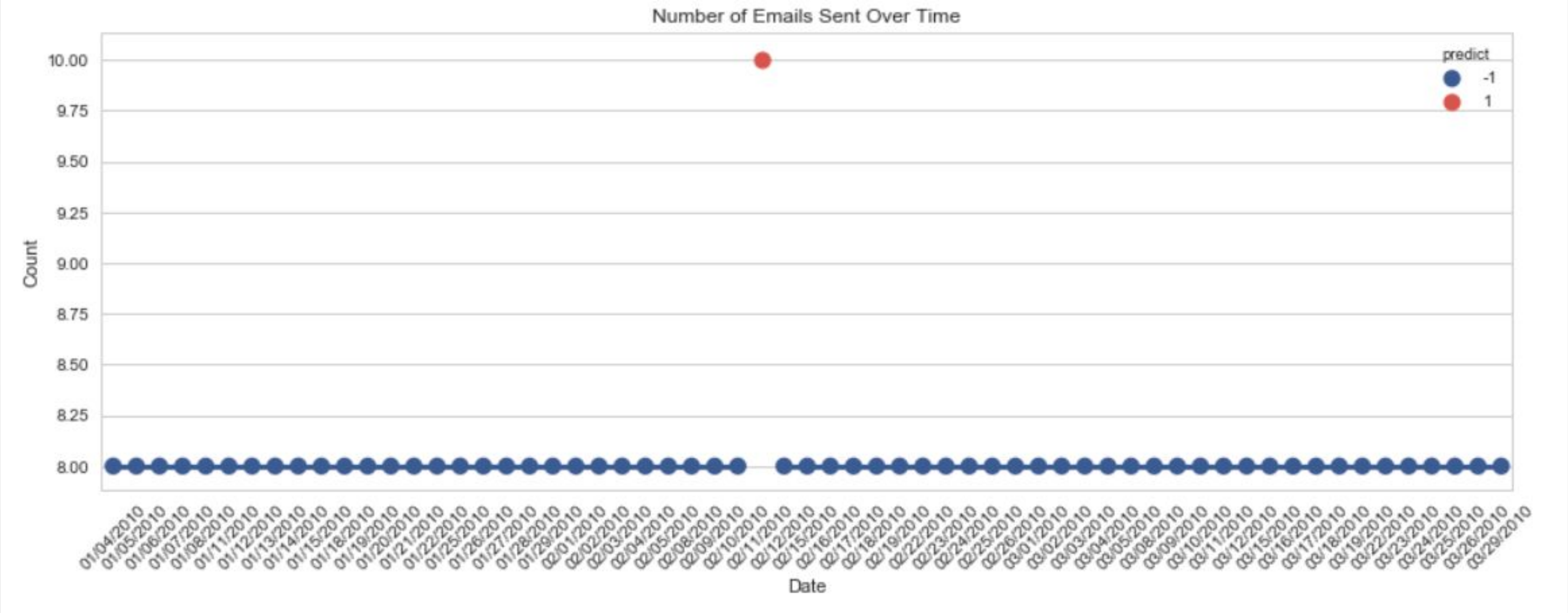
One Class SVM

- Unsupervised Learning, Special Case of SVM
- Trained on one class, unlabeled data, “the norm”
- Outliers are anomalies that are determined based on relative position to “the norm”
- Good for anomaly detection



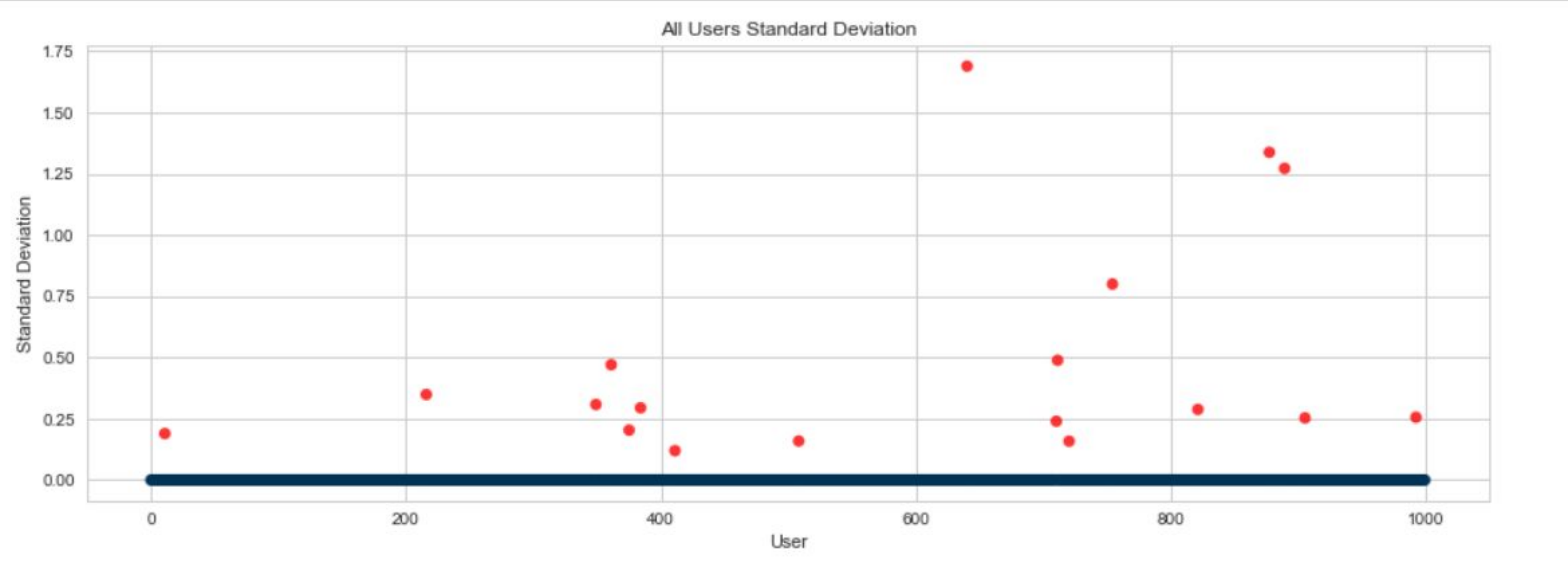
1. One-Class SVM (errors: 8)

Detecting Anomalies in Email Usage
OneClass SVM



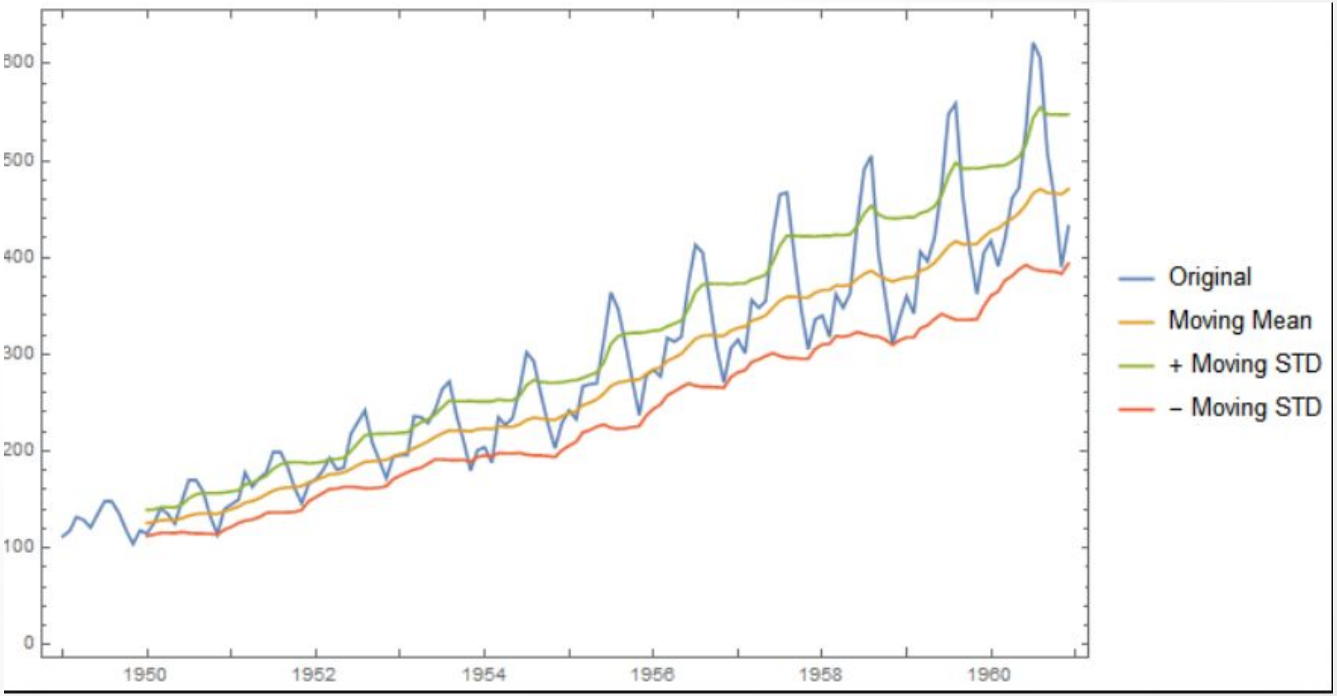
Impact Analysis OneClass SVM

- Total Number of Users: 1000
- Total Number of Anomalous Users: 18

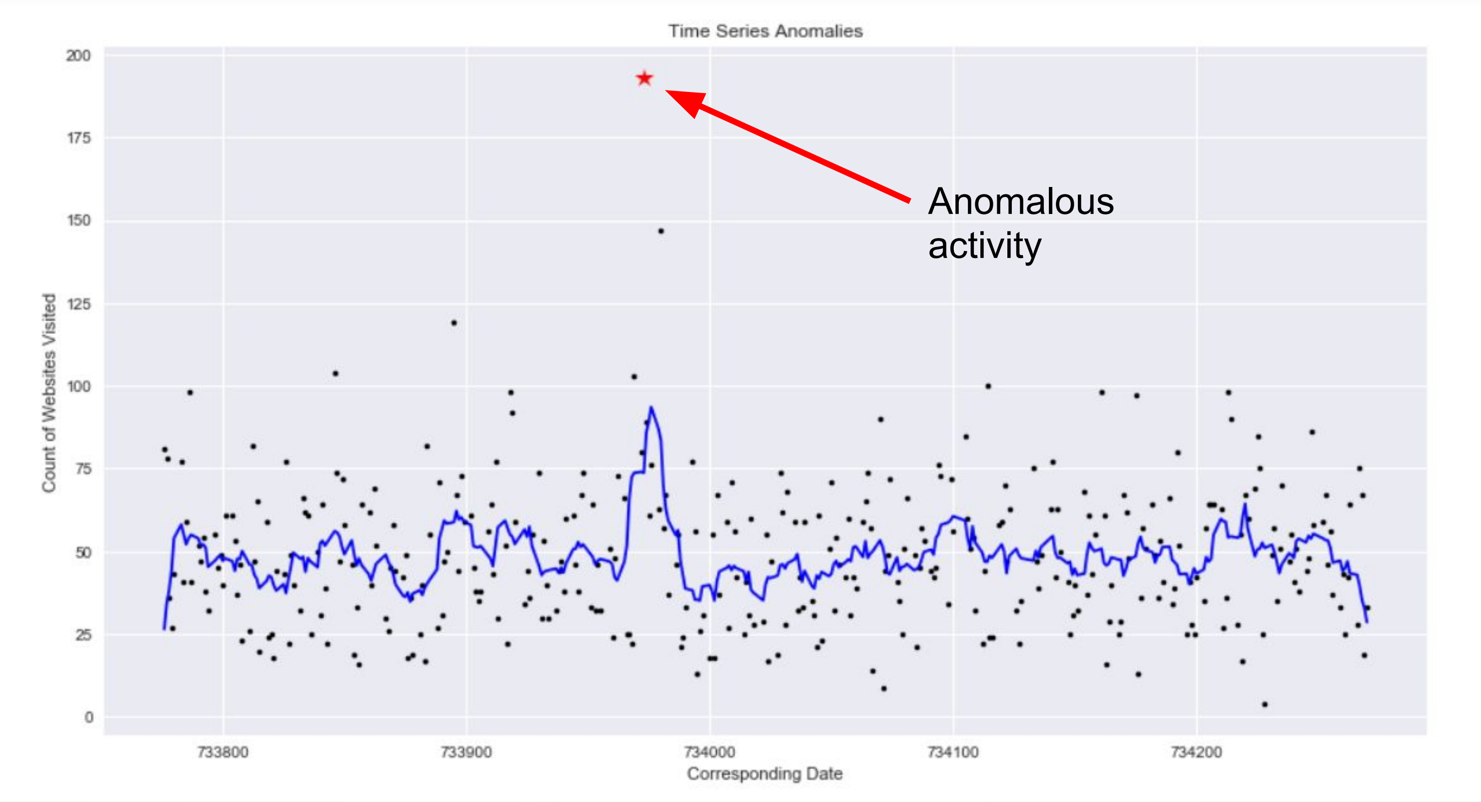


Time Series Anomaly Detection

- Time Series Data
- Moving Average
- Rolling Standard Deviation



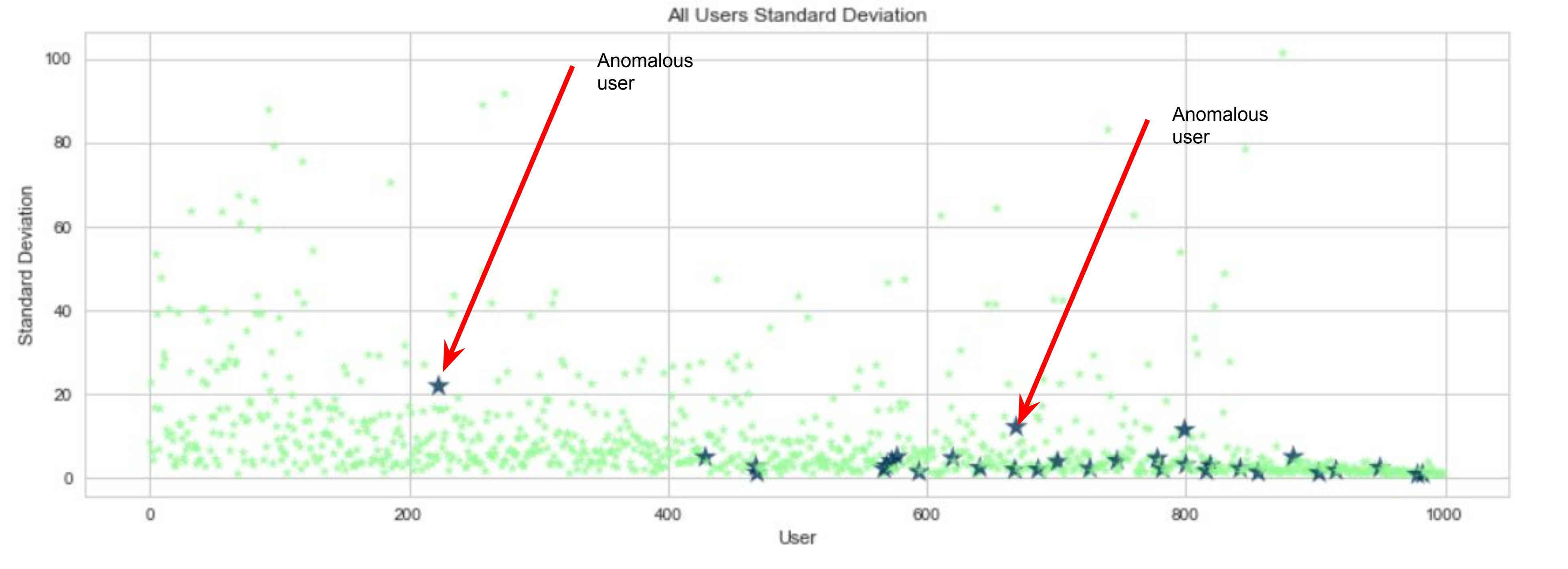
Time Series Anomalies Website Visits



Anomalous activity

Time Series Anomaly Impact Analysis

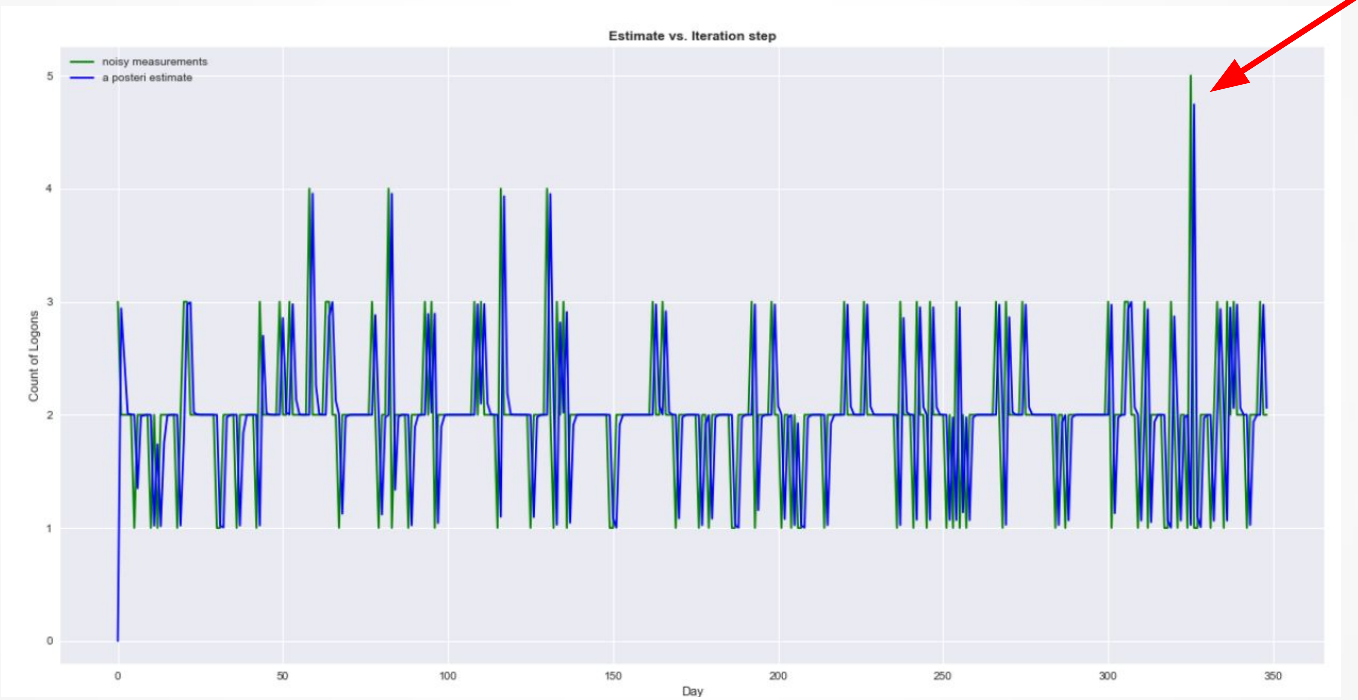
- Total Number of Users: 1000
- Total Number of Anomalous Users: 31



Kalman Filters

- Named for Rudolf Kalman
- Very Fast Computation
- Takes Measurements over time that have noise or inaccuracies and produces estimates of unknown variables
- Applied in Time Series Analysis
- Applied here, the anomalies are determined by values that surpass the threshold applied to the distance of the actual count versus the predicted amount

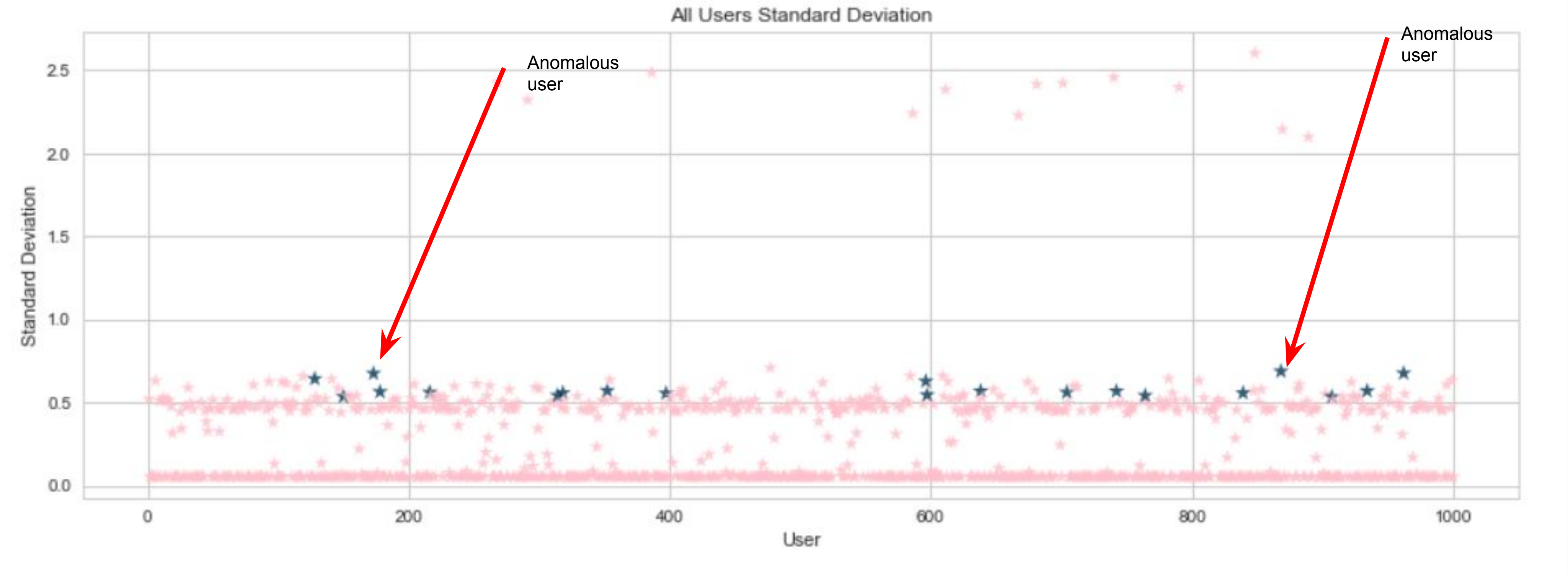
Kalman Filter Authentication Anomaly



Anomalous activity

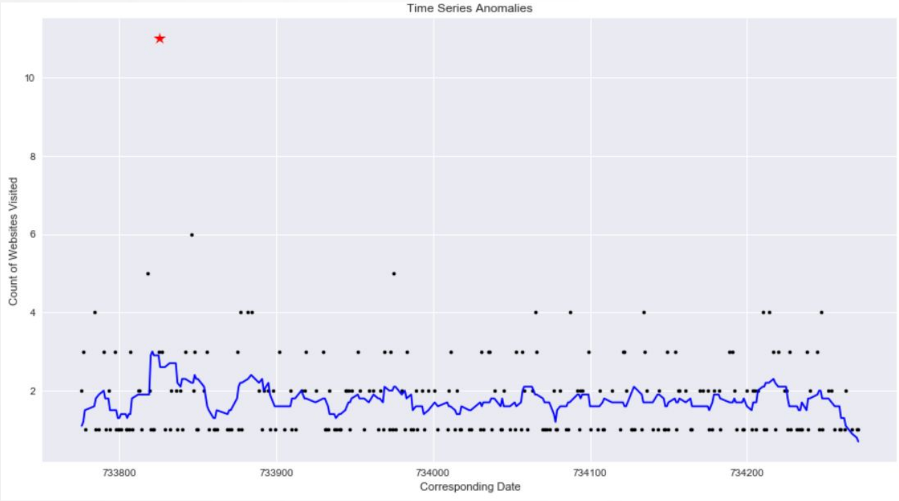
Kalman Filter Impact Analysis

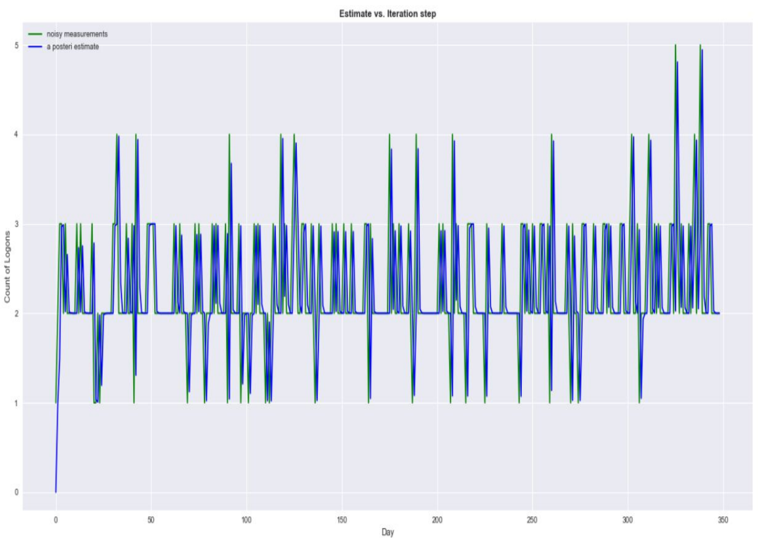
- 20/1000 User displayed anomalous Logon activity



Comparison

2 Users out of the final 1000 had anomalies in both Authentication and Proxy usage.





Email anomalies could not be correlated due to missing information relating email addresses to user id’s.

Discussion

Data Cleaning

- Date information did require reformatting into Python DateTIme objects. No further cleaning was required

Results

- All three methods of anomaly detection produced results that were validated via manual inspection of the individual user behavior.

Discussion Continued

- One Class SVM
  - 2nd fastest implementation
  - Easy to implement

- Time Series Anomalies
  - 3rd in performance, took the longest
  - More complicated to implement

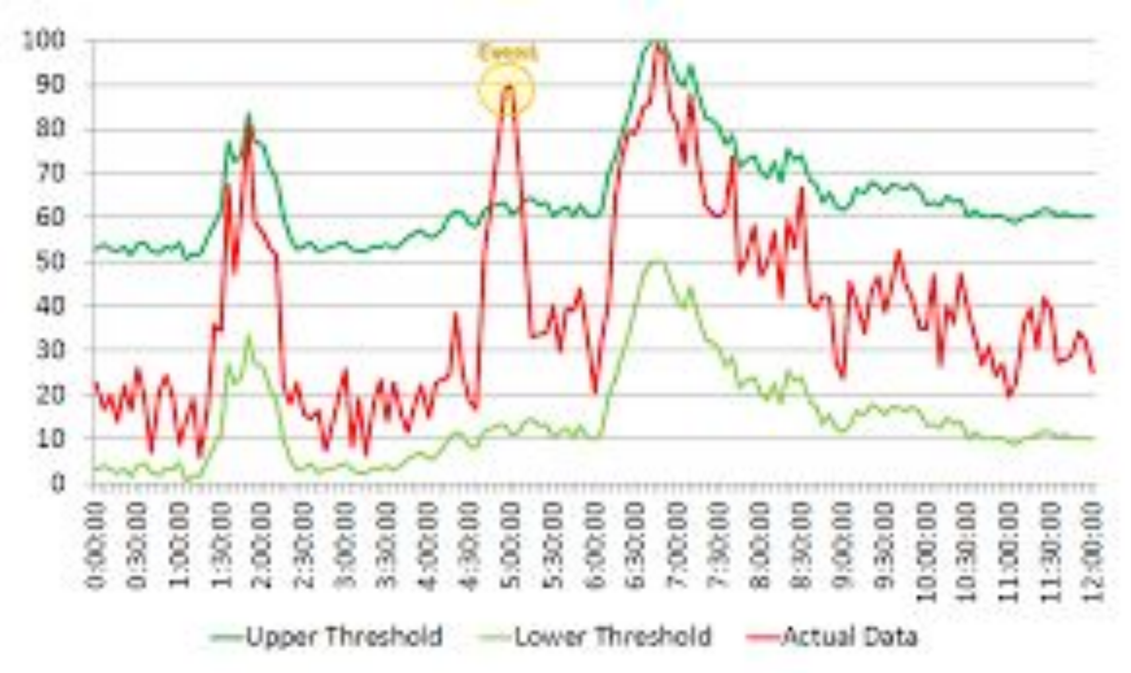
- Kalman Filters
  - Fastest Implementation
  - Most complicated to implement

Discussion Continued

- Thresholds
  - Static
    - Works when it is clearly defined/deterministic
    - Does not scale
    - A static threshold set too low can produce many false positives
    - A static threshold set too high can produce to many false negatives

Discussion Continued

- Statistical Thresholds
  - More advanced
  - Moving Averages (Time Series), Standard Deviation
  - Dynamic Thresholds



Conclusion

- No Free Lunch Theorem
  - There are multiple ways to perform anomaly detection, and all methods implemented during this capstone identified anomalous user behavior on various data.

- Model Optimization
  - Threshold identification would have to be tuned for the populations in which these models would be deployed. Continual evaluation of the effectiveness by Subject Matter Expert validation could be utilized to improve the models.

Real World Applications

- User Behavior
    - These models can be used in practice to identify users on a network who are behaving differently than normal. Users that suddenly change their behavior can sometimes be an indicator that something bad is happening.
- Implementation
    - In practice, proxy log data from proxy servers such as BlueCoat or Squid could be used for the Proxy model, Windows event logs, Event Code 4624 could be used for the Authentication model, and MS Exchange or Proofpoint logs could be used for the Email model.

Future Work

- MITRE ATT&CK Framework
  - Combining the models for each user and implementing a Markov chain that maps to specific Tactics, Techniques, and Procedures (TTPs) to produce a probability using Markov Chains that an actor has performed malicious chain of events will be completed.

- Model Ensembles
  - The Markov Chain should produce a probability and from this ensemble of models, another anomaly detection model can be used to map to specific event chains.

Thank You
Any Questions?