Anomaly Detection on Systems Data
Data Analytics Project Report
CS/STAT 5525

Team : Sun Gods

Ryan Kingery	rkingery@vt.edu
Navyaram Kondur	kvnav16@vt.edu
Soumya Vundekode	soumyav@vt.edu
Tim Arapov	tarapov@vt.edu
Jennifer McGuire	jmm8908@vt.edu

December 8, 2017

Introduction

The purpose of this project was to perform data analytics on a given systems monitoring dataset. The dataset consisted of various system records recorded by an unknown company on its employees over a period of about a year and a half. Such records include computer logons and logoffs, external device logs, files downloaded on external devices, emails, website visits, and employee information records. Note that the systems records only contain information for employees logged into a registered company computer.

Given the nature of this particular dataset, we decided that the primary form of data analysis performed in this project would be anomaly detection. The justification for this decision is that, presumably, the company's intent in collecting these records on its employees to begin with was to monitor the activities of its employees in the workplace to insure that they weren't violating any company policies. Anomaly detection is well-suited to identifying behaviors of employees who are acting abnormally relative to the majority of the company. Though anomaly detection doesn't guarantee that a given activity is violating company policy, identifying abnormal activity can allow system monitors to more efficiently determine which events are worthy of further investigation.

Our goal was thus to use various anomaly detection methods to figure out which employees were acting abnormally on company computers, and–if possible–to determine whether or not that abnormal behavior constituted anything serious. The anomaly detection techniques employed in this project include general exploratory data analysis, statistical based outlier detection techniques, email spam classification, network analysis, and sentiment analysis. The analyses and results of each such technique is described below.

Before moving on, we briefly describe the data management techniques and computational tools used to perform these analyses. Since supercomputing ability wasn't a huge need for our analyses, all data and scripts were stored in a remote git repository to ease collaboration and data access. All scripts were written using either Python or Bash due to their extensive data processing tools, and all data files were kept as CSV files for simplicity. Data visualization tools like Gephi and Tableau were also used.

Exploratory Data Analysis

Some interesting results were obtained just from doing EDA alone. One simple yet effective technique was looking at the website data. Analyzing the frequency of website visits, we observed that the usual popular websites, e.g. Facebook and Google, were visited most frequently, while quite a few websites were visited very rarely.

Analyzing the seldom visited websites yielded interesting results. We observed a few employees visiting suspicious websites like ActualKeyLogger and DailyKeyLogger, who turned out to work in the IT-security department, which makes sense given their roles. One suspicious employee turned out to be Robert Ali Webb from the Research department, who visited WikiLeaks frequently. Aside from Mr. Webb, we found that 44 employees who later left

the company early were known to have browsed suspicious websites at some point previously.

Another interesting result obtained from doing EDA was found by accident in the attempt to perform outlier detection on the data. Specifically, we observed that when binning various event counts per employee by date that some of the counts yielded something unusual: For a given event count, quite a few employees had exactly the same count value every single day. For example, looking at the number of websites visited per user per day, we found that an astounding 912 employees visited exactly the same number of websites every single day for 500 days, with that number varying only by employee. This strongly suggests that significant portions of the dataset were synthetically generated. An example of a clearly unrealistic employee is August Armando Evans, as he has the same event count each day across six completely different events, which would almost certainly not occur due to chance.

Outlier Detection

One of the most useful ways to detect anomalies is to use statistical outlier detection techniques. To do so, we first need to format the data in a way that makes these techniques useful and computationally tractible. To avoid having to one-hot encode every categorical variable and blow up the feature space, we instead bin the times into set periods and look at counts per time period of various log events for each user. This collapsed the feature space down to nine dimensions of counts. Various techniques were attempted for outlier detection, including Gaussian mixture models, seasonal-trend decompositions, local outlier factors (based on k-nearest neighbors), and isolation trees (based on random forests). Experimentally, we found isolation trees with by-hour bins to be the most useful outlier detection scheme for our purposes, as it is well suited at classifying far away points as outliers. The outlier contamination was chosen experimentally to be .5%.

Before classifying outliers, a PCA was performed to collapse the feature space down to two dimensions, partly for visual purposes to verify our choices of outliers looked reasonable, and partly because it was still able to explain about 85% of the variance in the data. Using exactly this technique we identified 120 outliers, which were generated by 57 different employees. One such employee turned out to be the president of the company, which seems reasonable.

Of the remaining 56 employees, 17 left the company early (possibly due to termination). Moreover, 21 of the 56 employees were identified previously as viewing suspicious websites, primarily WikiLeaks, with 11 of these leaving the company early. Note that we could probably identify even more outliers either by tuning the algorithm or by labeling these points as outliers and then using a supervised learning method to classify outliers, but due to time limitations we decided to stop at this point.

Network Association Analysis

Network analysis can be useful to for determining who communicates with whom and how frequently. To perform such an analysis, we used the email records to generate a directed

weighted graph with nodes representing email addresses, directed edges representing emails sent from one address to another, and weights representing the number of emails sent between addresses. Figure 1 shows a plot of the email association network, where node distances in the network are determined by out-degree. We can see from this a large clustering of nodes in the center, which likely represents primarily company email addresses. We can also see a few smaller clusters, as well as several outlying nodes likely representing either people outside the company or employees using non-company email addresses. Figure 2 shows an image plot of the adjacency matrix for the network, with email addresses sorted in descending order by frequency. We can see that there is frequent communication among about 1000 or so email addresses, but very sparse communication with others.

To understand the behavior of the network better we also calculated a PageRank for the network. We observed that Libby Rosalyn Richard is the highest-ranked employee, which makes some sense given she is a manager in the Finance department. However, since PageRanks alone didn't lead to any noteworthy anomaly detection mechanisms, we decided to pivot and instead analyze the sentiment of emails instead.

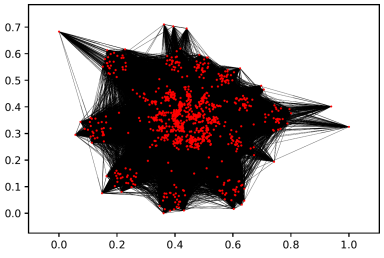

Figure 1: Network of email associations.

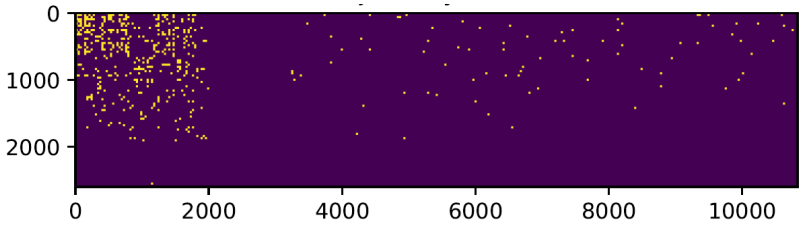

Figure 2: Image plot of the adjacency matrix.

Spam Detection and Sentiment Analysis

The employment records provided allowed us to identify 155 employees who left the company. Strangely, no new hires during this time period were recorded. Turning back to the emails, it was apparent that a significant number of the emails contained content with unusual, repetitive words. Because we believed that this would bias the sentiment analysis, we first ran the email content through a spam detection classifier. The spam classifier used was written by Peixuan Ding, and was pre-trained on a large corpus of emails using Naive Bayes. The classifier identified about 43% of all emails as spam. Additionally, we noticed that a couple employees had a significantly greater percentage of spam sent. Lunea Priscilla Hardy's and Declan Gareth Mclaughlin's emails were almost 50% spam, which was about twice the average over all email addresses. As it happens, both employees left the company early, possibly due to termination.

Due to its popularity and simplicity, the contents of the non-spam and spam data were separately analyzed using the sentiment analysis implementation provided in the Python TextBlob package. This implementation is also based off of Naive Bayes, and was pre-trained on a movie reviews corpus. Visual analysis of these sentiments yielded mixed results.

One useful result was a comparison of sentiment between employees who stayed in the company compared to employees who left early. A plot of this comparison is shown in figure 3. While one might expect employees who leave to have noticably more negative sentiment than employees who stay, this turns out not to be the case. Instead, employees who leave early appear to have significantly higher variability in their sentiments than employees who stay. Perhaps more level-headed employees are less likely to leave, or more emotionally unstable employees are more likely to be forced out or quit. Strangely enough, this same behavior shows up in the spam email as well.

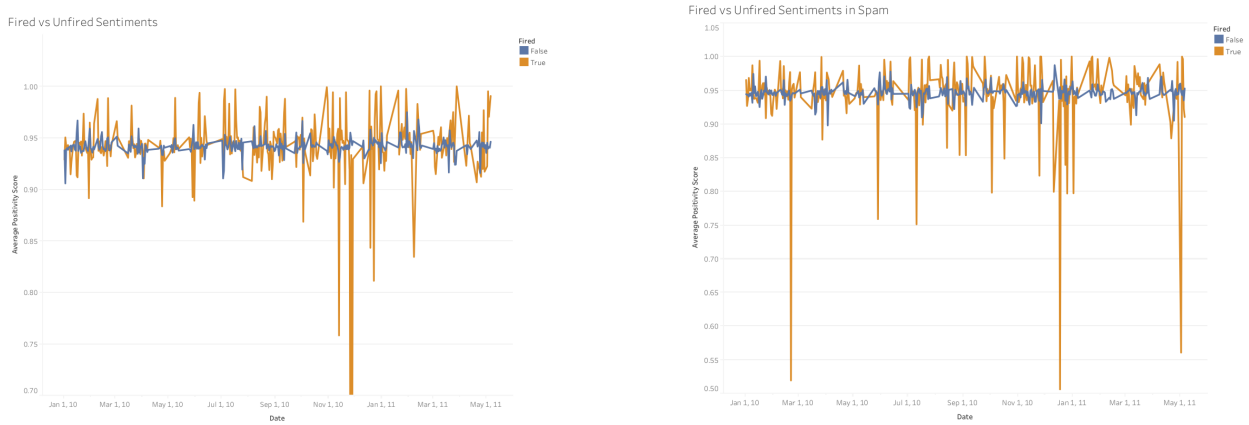

Figure 3: Mean sentiment of fired employees vs not fired employees in non-spam and spam emails.

Conclusion

In conclusion, we believe we have found an effective list of ways to efficiently determine which employees are acting abnormally using company systems data. EDA is always useful for identifying obviously abnormal events. Though possibly not used as extensively, outlier detection techniques appear to work very well. This is a benefit since outlier detection can largely be automated using machine learning methods, which would allow security personnel to spend more time focusing on other tasks. To support the results of an outlier detection scheme, network analysis, spam detection, and sentiment analysis are all useful ways to identify which outliers are resulting in truly suspicious activity. While there are certainly more advanced anomaly detection techniques out there, we have at least managed to identify a few simple, effective, and practical techniques for security professionals to try out.