Computer and Network Security Course Project Report Anomaly Detection

BY:
Youssef Ayman 34-1920 T-11

Merna Michel 34-0302 T-07

Kareem Nagui 34-0285 T-07

Youssef Ashraf 34-1410 T-08

Andrea Medhat 34-16887 T-13

Summary and Motivation:

Our program uses the KDD CUP 99 dataset to analyze and categorize different labels of DOS attacks, and has the ability to predict different attack scenarios and determine to which label of DOS attacks does this specific attack belong. A training sample is selected from the dataset. The machine is trained using unsupervised machine learning.

Design Choices:

Our program takes a random sample from the dataset belonging to the different DOS labels. 200 samples are selected randomly from each label. The dataset contains six different DOS labels including: neptune, back, teardrop, smurf, pod, and land. Then a specific subset of relevant attributes is selected to minimize the number of columns used, and principal component analysis (PCA) was applied to these attributes. The result of the PCA function is 3 components. We tried to cluster different attacks using K-means clustering into 4 clusters with maximum of 2 iterations. Finally the testing algorithm is applied on all of the four labels we have (Normal, TearDrop, Back, PoD).

Different Methods and Scenarios:

Due to the unbalanced nature of the dataset and attacks several preprocessing and analytic steps were needed prior to actually implementing the clustering algorithm. The Normal label had a huge count of data points, so much that the DoS labels were insignificant in comparison. Another issue was the high variance of data points in the Normal label which lead to the incorrect clustering of Normal points inside DoS clusters. Finally the 6 DoS types present in the dataset had a significant amount of overlap in features so 3 best fits needed to be chosen.

The first step that had to be done was filter the anomalies/noise in the Normal label. The idea was to cluster the Normal into a few clusters and use the largest of them as the main representative of the Normal label while the anomalies would be flushed in the lower count clusters. This approach worked well with 10 clusters with larger numbers having marginal additional benefit so 10 was the final cluster number.

The next step was to select the most appropriate 3 types of DoS to continue the analysis with. This step was done by running the clustering algorithm over the 6 DoS labels without including the Normal to observe the relations between the DoS labels. We discovered some interesting associations such that the Smurf and Neptune labels were almost always paired together in the same cluster as well as the fact that the PoD and Back labels were almost always distinctly clustered making them perfect fits for our application. Those two along with the Teardrop label were selected for further analysis.

The unbalanced dataset also posed a problem since the largest label would naturally engulf all other insignifiant labels, to overcome this sampling was used to select a random collection of data points with equal counts form each label for a better attempt at clustering. The least label was that of PoD with only 206 data points so a sample size of 200 was chosen for the clustering algorithm. And so each label would randomly provide 200 data points, given that the Normal label sample was taken after the anomaly/noise reduction.

Advantages and Disadvantages:

- Unsupervised learning: This method is quick and easy to run. Moreover it does not require prior knowledge of the data. On the other hand, sometimes not all generated classes match with the correct informational classes.

- PCA: The compaction of the numerous dimensions of the original dataset to 3 principal components resulted in two major advantages. The first being that the clustering process took less time while running, the later being that clustering produced more accurate results with better defined clusters as well as more consistent results across runs of the clustering algorithm.

- K-means clustering: For big data with lots of variables, K-means is computationally fast. Also K-means produces tighter clusters than other clustering methods. On the other hand, it is difficult to predict the K value, and also applying the method to different initial partitions may produce different final clusters.

Libraries and Frameworks:

We implemented our project using python language. Most of features implemented are from Sklearn libraries. For preprocessing the data we used PCA and Standard Scalar. For clustering we used K-means cluster. Moreover we used some python libraries as numpy, pandas, multiprocessing, and seaborn.

References:

- **DataSet Link:**
  https://drive.google.com/file/d/1O4SZa82AR41gI37U2NSk3KfDp0BBNOSZ/view?fbclid=IwAR0RDuFOg7vVAFlkzUFqGCYpy28UP4A4tqbiVR1y3NDZVSZp48toN1EIOdg
- **Libraries:**
  https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html
  https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

- **Papers:**
  We referred to the following papers to understand the KDD CUP 99 dataset and find the most relevant attributes to use in out classification method.
  https://ijaiem.org/pabstract_Share.php?pid=IJAIEM-2015-03-03-1
  https://pdfs.semanticscholar.org/97f2/79238718c32fa4c04ba888cd440b36c2ca8c.pdf

- **Code repository:**
  https://github.com/253Youssef/Security