Outlier detection on network flow analysis
Quang-Vinh Dang

To cite this version:
Quang-Vinh Dang. Outlier detection on network flow analysis. 2018. hal-01854006

HAL Id: hal-01854006
https://hal.science/hal-01854006
Preprint submitted on 6 Aug 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Outlier detection on network flow analysis

Quang-Vinh Dang

Received: date / Accepted: date

Abstract It is important to be able to detect and classify malicious network traffic flows such as DDoS attacks from benign flows. Normally the task is performed by using supervised classification algorithms. In this paper we analyze the usage of outlier detection algorithms for the network traffic classification problem.

Keywords outlier detection · classification · ddos detection

1 Introduction

A denial-of-service (DoS) attack is characterized “by an explicit attempt by attackers to prevent the legitimate use of a service” [Mirkovic and Reiher, 2004]. If the attackers coordinate the DDoS traffic from multiple sources to perform the attack, it will be Distributed Denial-of-Service (DDoS) [Koay et al, 2018].

Multiple studies have analyzed the detection and prevention strategies of DDoS attacks by using classification algorithms [Douligeris and Mitrokotsa, 2004; Fouladi et al, 2016; Koay et al, 2018; Alsirhani et al, 2018]. While these methods achieved a lot of success, they suffer from imbalanced dataset problem [Krawczyk, 2016] and lack of detecting unfamiliar flows. For instance, these techniques mail fail to detect a new DDoS attack technique that they did not see during the training period. Furthermore, supervised classification algorithms usually exhaust of data.

Outlier detection algorithms [Aggarwal, 2017] try to distinguish outlier points from normal traffic data. Hence, the techniques might be performed in unsupervised manner [Campos et al, 2016]. Furthermore, outlier detection
algorithms can deal well with extremely imbalanced dataset, such as 1:1000 ratio [Krawczyk, 2016].

In this paper we evaluate the performance of outlier detection algorithms on detecting DDoS traffic. Our work is related to other evaluation studies, such as of [Hodge and Austin, 2018]. However, the authors of [Hodge and Austin, 2018] do not analyze the performance in case of imbalanced datasets.

2 Evaluation

2.1 Algorithms

We evaluated the following algorithms:

- **CBLOF** (Clustering-Based Local Outlier Factor) [He et al, 2003].
- **HBOS** (Histogram-Based Outlier Score) [Goldstein and Dengel, 2012].
- **IForest** Isolation Forest [Liu et al, 2008].
- **k-NN** (k Nearest Neighbors)[Ramaswamy et al, 2000].
- **MCD** (Minimum Covariance Determinant) [Rousseeuw and van Driessen, 1999].
- **OCSVM** (One-Class SVM) [Ma and Perkins, 2003].
- **PCA** (Principal Component Analysis) [Shyu et al, 2003].

2.2 Dataset

We used the dataset provided by [Ghorbani and Lashkari, 2018] that contains 464,976 samples that are assigned the labels “Attack” or “Benign”. We consider these labels as ground truth. A set of 76 features is provided. Some of the features are:

- Flow.Duration
- Tot.Fwd.Pkts (Total Forward Packets)
- Tot.Bwd.Pkts (Total Backward Packets)
- Fwd.Pkt.Len.Max
- Fwd.Pkt.Len.Min

The “Attack” traffic contains 3.76% of the whole dataset (17,462 samples). In order to evaluate the performance of the outlier detection algorithms, we create different datasets with different “Attack” Ratio, range from 0.01 to 0.99.

To change the class ratio, we applied the following function to select a subset of the dataset:

```python
def make_dataset(benign_ratio = 0.9,
                 df_attack = df_attack,
                 df_benign = df_benign):
    N_attack = len(df_attack.index)
    N_benign = len(df_benign.index)```
n_attack = N_benign / (benign_ratio / (1-benign_ratio))
sample_df_attack = df_attack.
sample(int(n_attack), replace=False)
res = pd.concat([df_benign, sample_df_attack])
res = shuffle(res)
res = res[[col for col in res
if not len(set(df[col]))==1]]

#divide train/test
train = res.sample(frac = 0.7, random_state=200)
test = res.drop(train.index)
return (train, test)

3 Results

In this section we present the performance in term of AUC and Accuracy scores of each algorithms. Overall, while the benign_ratio increases the Accuracy scores increase for all algorithms, that can be explained by the imbalanced of the dataset. Another observation is that the performance of the algorithms are very similar between training and testing set due to the fact that these algorithms are all unsupervised. In term of AUC score we could see that IForest and PCA algorithms achieved the best scores while the benign_ratio increases.

4 Conclusions

In this paper we evaluate the performance of unsupervised outlier detection algorithms in detecting DDoS attacks. We showed that the outlier detection algorithms perform, particularly Isolation Forest or PCA-based algorithms, perform best if the proportion of outlier instances is small. This contrasts with popular classification algorithms. In the future we will focus on analyzing Isolation Forest and PCA algorithms in other scenarios.

References

Fig. 1: CBLOF

Fig. 2: HBOS

detection: measures, datasets, and an empirical study. Data Min Knowl Discov 30(4):891–927
Outlier detection on network flow analysis

Outlier detection on network flow analysis

Fig. 7: PCA