Towards Explicable and Adaptive DDoS Traffic Classification

Yebo Feng
University of Oregon
yebof@cs.uoregon.edu

Jun Li
University of Oregon
lijun@cs.uoregon.edu

ABSTRACT

The traffic classification of the Distributed Denial of Service (DDoS) attack is a well-studied but not completed field. In this paper, we propose a learning-based DDoS traffic detection and classification method, which utilizes a modified k-nearest neighbors algorithm to detect DDoS threats, then conducts fine-grained traffic classification using risk degree sorting with grids. To improve efficiency, we use a k-dimensional tree to partition the searching space, shortening the time for queries significantly. Compared with the previous learning-based approaches, this method is highly explicable. Additionally, it can adapt to new environments with very little measurement information as prior knowledge.

1. INTRODUCTION

Decades of research and industry efforts have led to a myriad of DDoS detection and classification approaches. As machine learning algorithms are becoming more and more sophisticated, many researchers begin to harness such techniques on big data for in classifying DDoS attacks. For instance, Suresh et al. [1] evaluated a variety of machine learning algorithms in detecting DDoS, including SVM, Navies Bayes, K-means, etc.; Yuan et al. [2] trained a recurrent deep neural network to discover DDoS activities. The results of such methods demonstrate the strong ability of machine learning algorithms in digging useful knowledge from massive training data.

However, the negative aspects of learning-based approaches are also apparent. Most of the learning-based approaches are inexplicable when making predictions. This black-box feature is troublesome because the network administrator somehow needs to conduct access control on the prediction result, and an unexplainable result may lead to unexpected collateral damage. Moreover, learning-based methods are not adaptive. Their performance highly depends on the reliability of the training data, while DDoS attacks are very diverse. An attack in one environment may be considered as legitimate in another.

To fill this missing gap, we designed this explicable and adaptive DDoS traffic classification method based on machine learning. In the detection phase, it employees the k-nearest neighbors (KNN) algorithm and a k-dimensional tree (KD tree) to detect the DDoS attacks with network profiles. Here, the KD tree can dramatically accelerate the query process of KNN, increasing efficiency significantly. If a threat is detected, our approach enters the classification phase. It will sort the traffic sources based on risk degree using grids, then iteratively identify the malicious IP addresses until the traffic profile returns to a benign area.

The rest of this paper is organized as follows. After describing the method design in Section 2, we describe the future work in Section 3. We then conclude this poster paper in Section 4.

2. DESIGN

Our approach has two phases, which are DDoS detection and traffic classification. It monitors the traffic in batches. Each batch t is a uniform time bin, which is also the most basic detection unit. In our implementation, we set each batch as 5 seconds.

During each batch t, our approach will extract features to form a traffic profile S ($S = \{feature_1, feature_2, ..., feature_n\}$) and input it into the detection module.

2.1 Detection Phase

The goal of the detection phase is to judge whether the traffic profile S is under DDoS. We use the KNN algorithm to achieve this. KNN algorithm is a non-parametric method used for classification, which will find the k nearest neighbors of the traffic profile S and use the neighbors' identity to vote for the label of S. This algorithm is simple and straightforward. In our implementation, we chose six feature to construct the traffic profile, which are the number of bytes, the number of packets, the ratio of inbound packet number to outbound packet number, the number of SYN flags, the number of UDP packets, and the number of ICMP packets.

KNN algorithm has one weakness. Although it takes no time to train the model, the prediction requires a time complexity of $O(nlogn)$ to complete because it needs to enumerate the data points in the searching space to find the k nearest neighbors. Hence, we leverage the KD tree to partition the searching space, reducing the number of data points to enumerate. With the KD tree, whenever there is an incoming profile, we only need to search a subarea to predict the result. Figure 1 shows a simple but vivid example, where only two features are included in the training and prediction.

Furthermore, according to our experiment results, most of the DDoS profiles have relatively big differences compared with legitimate traffic profiles. This leads to an interesting fact that, most of the searching areas partitioned by the KD tree only have either benign traffic or malicious traffic profiles. Just as the red and green areas in Figure 1, we define the searching areas as a confirmed area if one type of the traffic profiles dominate the area, and the number of another type of traffic profiles is less than $k/2$. If the current traffic profile falls within the confirmed area, we

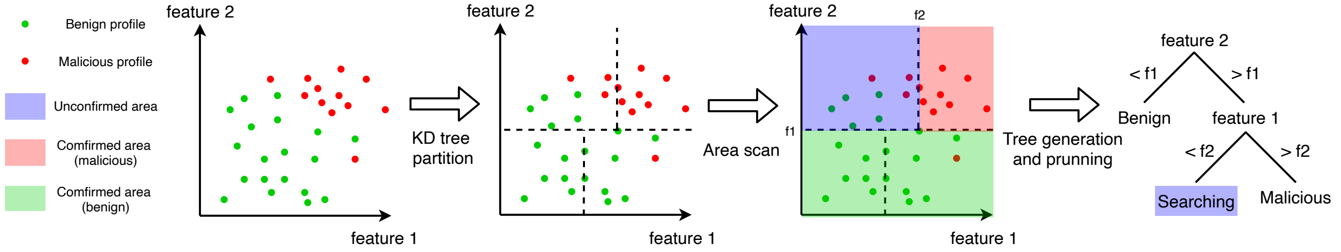

Figure 1: DDoS detection with modified KNN and KD tree

can directly label the profile S without running KNN. Thus, we convert the KNN searching space into a tree-like data structure. If DDoS attacks do not happen frequently, this tree-like data structure can reduce the time complexity for traffic monitoring to $O(1)$.

2.2 Classification Phase

Once an anomaly is detected, the method enters the classification phase. The design philosophy of the traffic classification is that the traffic profile is currently in a malicious position, and we need to conduct access control on some of the sources, so that the traffic profile can return back to a benign area. To avoid collateral damages, we need to figure out the shortest path to move the traffic profile.

The first step of the traffic classification is to calculate the shortest path. In the implementation, we used the breadth-first search to scan the surrounding areas until we reached a benign position. Then, our approach represents the shortest path as a vector p. For instance, if there are only two features and $p = (0, -x)$, we need to modify the traffic profile to reduce at least x in feature 2 but bring the minimum changes to feature 1.

We conduct the classification for malicious sources by building traffic profiles for each IP address. Afterward, we sort the IP addresses in the order of decreasing feature 2, and then in the order of increasing feature 1. Finally, we conduct access control on the IP addresses in such order until the overall traffic profile returns to a benign area.

The sorting of IP addresses is expensive, especially when the network we are protecting is ISP-level. Hence, we can divide the profile space of IP addresses into a grid configuration and only sort the IP addresses in the grids with the largest values in feature 2. This improvement will significantly accelerate the classification process.

2.3 Adaptability

Users do not need to retrain the proposed model to fit it into a different network environment because the design philosophy of this method focuses on the adaptability. We can use a variety of prior knowledge to evolve the model, making it even robust to the unacquainted environment.

Here, we assume the user will have some types of limited knowledge to start with. We discuss three cases below.

2.3.1 Traffic measurement

Assume that we have the measurement information about the new environment, we can normalize the KNN searching space from the trained environment to the new environment according to the two networks' traffic distributions.

2.3.2 Online learning

If the traffic monitoring system can obtain labeled traffic with the system running, we can efficiently conduct online learning on the proposed model. The KNN algorithm does not require training. However, the KD-tree, along with the confirmed areas, need to refresh to reflect new knowledge. Nevertheless, the time complexity of refreshing the model is only O(n).

2.3.3 Incomplete threshold

In some circumstances, the user of this method may know some incomplete threshold values or rules in the new network environment. They can then build a decision tree based on the preliminary knowledge and merge it with the trained classifier, a tree-like data structure. If the prior knowledge of the new environment contradicts with the trained model, the user can manually indicate the decision priority.

3. FUTURE WORKS

The design and implementation of this approach is completed. In the future, we seek to deploy this approach in diffent environments and evaluate its efficacy, accuracy, and system overheads. Additionally, we will use both simulated and captured traffic to train the model, and evaluate it in a totally different environment to test its adaptability.

4. CONCLUSIONS

This poster paper proposes an approach to detect and classify DDoS traffic, which is explicable and adaptive. With the KD tree and the modified KNN, this approach generates a tree-like classifier, which not only makes predictions promptly but also gives the network administrator a clear perspective of the network conditions. Furthermore, people can easily adapt the trained model to a different environment without retraining the model from scratch.

5. REFERENCES

[1] Manjula Suresh and R. Anitha. Evaluating machine learning algorithms for detecting ddos attacks. In David C. Wyld, Michal Wozniak, Nabendu Chaki, Natarajan Meghanathan, and Dhinaharan Nagamalai, editors, *Advances in Network Security and Applications*, pages 441–452, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[2] Xiaoyong Yuan, Chuanhuang Li, and Xiaolin Li. Deepdefense: identifying ddos attack via deep learning. In *2017 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 1–8. IEEE, 2017.