24 | Anomaly Detection for DDoS attacks by using Flow Data and Machine Learning

Author: Christiaan van den Bogaard

Abstract

Distributed denial-of-service (DDoS) attacks is a rising challenge which impact, among others, companies and websites. With an increasing number of DDoS services these attacks become more accessible and the extent of these attacks can become considerable. Studies have shown that for large enterprises the costs of DDoS attacks have grown from $1.6 million in 2016 to $2.3 million in 2017 on average per attack. Before a DDoS attack can be mitigated, it needs to be detected. The biggest challenges in DDoS detection are reducing the detection time and increasing the accuracy. This paper proposes an improvement on the current landscape of DDoS attack detection methods using machine learning and flow data. Using data aggregation on network flow data a new method is proposed to detect DDoS attacks. Different classifiers are used and compared on both time and classification performance. Results show that they equally perform well, but Random Forests and Logistic Regression have a significant advantage in terms of computation time. There is no significant performance increase in aggregation size of the flow data.

24.1 Introduction

Distributed denial-of-service (DDoS) is a rapidly growing problem, posing an immense threat to the Internet. An attack can be executed if a group of computers connect to a server to temporarily disrupt services. The financial costs of DDoS attacks are growing each year. Studies of Kaspersky have shown that for large enterprises the costs of DDoS attacks have grown from $1.6 million in 2016 to $2.3 million in 2017 on average per attack Labs (2017). DDoS attacks also not only case financial impact, as other impacts such as reputation are in jeopardy. The longest attack recorded by Kaspersky in 2017 was 277 hours, which is more than 11 days Kaspersky (2017).

To decrease these aforementioned impacts better mitigation techniques are required. It is important however, that these attacks are detected and mitigated before large impacts transpire. As one could conclude, time is of the essence therefore fast and precise methods should be made. Machine learning is used in this case to detect attacks but training but time and computation resources are limited, so not all data can be used.

Several papers propose methods for DDoS anomaly detection with different algorithms and data sets. Lazarevic, Ertoz, Kumar, Ozgur, and Srivastava (2003) use several machine learning methods which have promising results, the problem however is it that it uses packet based data which makes the analysis a slow process. Sekar, Duffield, Spatscheck, van der Merwe, and Zhang (2006) implemented a larger-scale DDoS detection system which consists of a lightweight and a heavy detection method. While the results are positive, the model is quite extensive which makes in in terms of time and memory efficiency a less favorable model. Karimazad and Faraahi (2011) uses Neural Networks to detect DDoS attacks, again detection accuracy reaches high levels but package based data is used as input which makes it slower. Using these aforementioned papers we will try to a method to detect DDoS attacks with a high accuracy, while keeping high performance using Netflow data. In this paper we propose to improve the state of the art on anomaly detection for DDoS attacks by using flow data and machine learning.

To pursue our goal, we have defined the following research questions (RQ) as the basis of our research:

- **RQ1:** What are the limitations of the current anomaly detection techniques?
- **RQ2:** What methods and metrics can be used to detect anomalies in DDoS attacks?

- **RQ3:** Which machine learning algorithm gives the best speed and accuracy for the given metrics?

The remainder of this paper is organized as follows. Section 2 will discuss the limitations of the current state-of-art. We then discuss the possible methods and metrics which can be used to detect anomalies. Finally, we conclude with results and a conclusion.

24.2 Background Theory

In this section we discuss the definitions used in this paper. First flow data is explained and how this compares to traditional package based data. Furthermore, a subset of different DDoS attacks are described.

24.2.1 Flow Data

Flow data is, in comparison with the traditional package based data, a lot smaller as it does not contain many features. All packets are aggregated into flows, which contain the following fields: *Start time, end time, Source IP, Source Port, Destination IP, Destination Port, Protocol, Number of Flows, Number of Packets, Number of Bytes.* While flow data is more lightweight, data analysis should be computationally less extensive. We expect that classifiers trained with normal package based data generally is slower in both training and prediction. In real time situationws these packets need to be inspected which have an overhead in network communication. This brings us to the trade-off between detection accuracy and speed.

24.2.2 Attacks

The landscape of different DDoS attacks is extremely wide, but we will focus for simplicity on 3 different types of attacks: ICMP, SYN-Flood and UDP-Flood. Each with its own characteristics. Having distinguishable characteristics makes it easier to detect attacks.

ICMP

This attack is based on the ICMP protocol, where the target is overwhelmed with ICMP echo requests. In normal network traffic there will only be a small amount of

ICMP based network communication, as it is mainly used for determining whether a host is online. If in some period the target observes that a high percentages of all observed flows uses the ICMP protocol it should be flagged as an anomaly. The following characteristics are compared against normal traffic:

- All packets through ICMP protocol
- Low amount of packets per flow
- Low average packet size per flow

SYN-flood

Syn-flood attack exploit the three-way handshaking process between the source and the target. First a SYN message with a spoofed source IP address is sent. The victim then responds with a SYN/ACK and waits for the ACK packet to arrive, which never happens. This leaves the target with an open connection which over time creates an overflow in finite reserved data structures. Characteristics that identify SYN-flood flows:

- Only SYN flag set in flow
- Low average of Packets per flow
- Low average of Packet size

UDP-Flood

UDP-flood attack is quite similar to ICMP flood. The attacker uses (forged) packets and sends these to the target, with the goal of overwhelming the bandwidth of the target. All available bandwidth is therefore consumed and other communications are too slow or d. These generally can be identified with the following characteristics

- High average of Packets per flow
- High average of Flow size

24.3 Limitations

There are multiple different approaches to detect DDoS attacks. We first have signature based detection. In this case we classify a flow as DDoS attack in case it

matches some pre-defined signature. In this case, we first will need to know what signature matches a DDoS attack. Furthermore, we also have rule-based detection. While this is similar to the former, this means that a flow must abide rules to be either classified as normal data or attack data. Both approaches share a common downside. They rely on the knowledge provided by an expert system, usually a human export, to do the job. These systems are well-known to detect attacks that must adhere to rules or match with signatures (Casas, Mazel, & Owezarski, 2012). In case of new, unknown attacks these will not be detected, as a system can not detect what he does not know. The false positive rate, e.g. the amount of normal data classified as an attack, however will be low.

Former research shows that machine learning techniques work for DDoS attack detection. There are several drawbacks however. Some papers use package based data for training and classifying (Chen, Ma, & Wu, 2013; Karimazad & Faraahi, 2011; Lazarevic et al., 2003). The main problem with pcap data is that every individual network packet is inspected. Also, these packets contain a lot more information in comparison with a network flow. If we were to use machine learning techniques on the network flow data we would expect better performance as there is less input data with fewer features. We assume that there is a performance decrease in classification accuracy.

24.4 Detection

To determine how detection of DDoS attacks using machine learning trained with flow data compares to detection methods using package based data, we will create several classifiers to assess the performance. The normal data used in this experiment is obtained from Appneta (n.d.). The data containing DDoS attacks are obtained through DDoSDB (n.d.). A summary of all used datasets can be found below:

- Normal data: 2 datasets consisting of 52731 flows
- ICMP data: 2 datasets consisting of 306121 flows
- SYN data: 2 datasets consisting of 5560 flows
- UDP data: 2 datasets consisting of 23878 flows

An complete overview of the setup can be found in Figure 24.1. In this section all of the following subsystems are described:

1. Preprocess: For a given integer N, N flows are grouped and averaged into one frame.

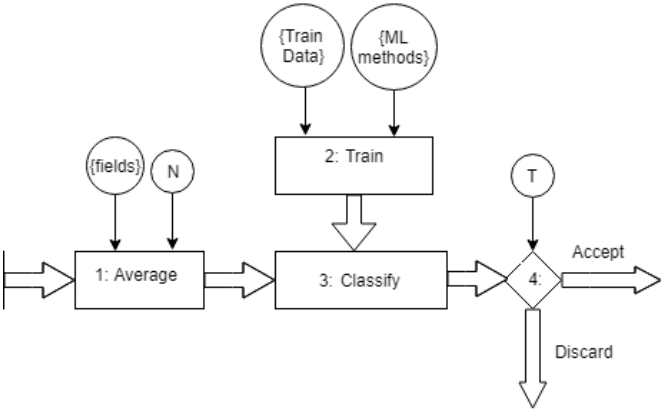

Figure 24.1: Detection Setup

2. Train: For given train data sets and a method, a classifier is trained.
3. Classify: A probability is calculated whether a frame is observed as DDoS or not.
4. Threshold: The frame is compared with a threshold and conclusively accepted or discarded.

24.4.1 Data Preprocessing

The data preproccesing stage is the first step as defined in Figure 24.1. The N incoming flows are first aggregated into one **frame**. Where we define a frame as: a frame of N is an aggregation of N flows into one single data row. In the experiment N is varied with different sizes. As data is aggregated, different methods are applied to aggregate the data. The metrics that are initially selected are based on the information of Section 24.2.2. These metrics are then aggregated with the following operations:

- average packet size: mean
- destination port: number of unique values
- number of incoming packets: mean
- source port: number of unique values

- source address: number of unique values
- flag: Split into individual flag fields and normalized

The flag is one hot encoded into seven different flags: ACK, FIN, SYN, PUSH, REST, URGENT and ALL. These are then normalized on the distribution of these values. For example, during a SYN Flood the large majority of incoming flows will only have the SYN flag set thus the frame will have a value close to 1 for the SYN flag. Using these flags there can be a strong distinction between attacks. Examples of a SYN flood attack frame and a normal traffic frame can be found in Table 24.1 and Table 24.2. We can clearly see the distinction between the normal traffic and the syn flood traffic just by observing the SYN flag value.

The reason why the flows are aggregated is based on the fact that an attack is continuous over time. If in some period a lot of flow deviate from normal behaviour, something might be wrong. A significant attack will last longer than a small amount of flows. Furthermore, the majority of all incoming flows will be the flows linked to the DDoS attacks. The assumption in this case is that the amount of false positives is decreased. In the situation where all flows are individually inspected there can be a case that one flow and one flow only is classified as positive. Maybe this flow has a large amount of packets and bytes, and might be linked to an UDP flood. It also can be a UDP stream. In the case where all other flows within the frame are classified as negative, we can more safely assume this UDP stream is classified as a false positive.

One main advantage of data preprocessing this data in the aforementioned way is that it does not need some form of network scaling to make it work in all networks. For example, if one would classify based on packet rate per second trained with data sets from a high speed network, it will never perform well on a regular home network. Also, if it is the other way around, the high speed network will only generate false positives.

The main disadvantage of preprocessing this data in the aforementioned way is that when the minority is actually negative, while the complete frame is classified as positive, the negative flows are dropped and therefore the classifier indirectly "denies the service" of clients. In the ideal situation the frames classified as positive should solely consist of actual DDoS attack flows.

24.4.2 Training & Classification

Using SKLearn and Python three different classifiers are built to detect DDoS attacks; Random Forest, Logistic Regression and Neural Networks. Mok, Sohn, and

Table 24.1: 5 Frames of Normal traffic, with N=100 aggregated rows

	A	F	P	R	S	U	X	avg_size	dp	ipkt	sp	sa	target	TCP	UDP	ICMP	IGMP
0	0.08	0.05	0.08	0.0	0.22	0	0	112.84	59	3.5	43	6	0	0.22	0.68	0.1	0.0
1	0.37	0.18	0.37	0.07	0.44	0	0	113.02	57	7.12	45	1	0	0.44	0.56	0.0	0.0
2	0.61	0.32	0.58	0.04	0.56	0	0	247.28	52	7.53	60	43	0	0.61	0.39	0.0	0.0
3	0.9	0.8	0.74	0.04	0.73	0	0	394.85	91	19.21	14	28	0	0.9	0.1	0.0	0.0
4	0.86	0.7	0.71	0.09	0.71	0	0	324.02	63	6.41	42	25	0	0.86	0.14	0.0	0.0

Table 24.2: 5 Frames of Syn flood traffic traffic, with N=100 aggregated rows

	A	F	P	R	S	U	X	avg_size	dp	ipkt	sp	sa	TCP	UDP	ICMP	IGMP
0	0	0	0	0	1.0	0	0	20.0	1	1.0	100	100	1.0	0.0	0.0	0.0
1	0	0	0	0	1.0	0	0	20.0	1	1.0	100	100	1.0	0.0	0.0	0.0
2	0	0	0	0	1.0	0	0	20.0	1	1.0	100	100	1.0	0.0	0.0	0.0
3	0	0	0	0	1.0	0	0	20.0	1	1.0	100	100	1.0	0.0	0.0	0.0
4	0	0	0	0	1.0	0	0	20.0	1	1.0	100	100	1.0	0.0	0.0	0.0

Ju (2010) uses Logistic Regression to classify attacks with promising results on package based data. Neural Networks are chosen as Karimazad and Faraahi (2011) found that Neural Networks can be used effectively in DDoS detection. As an addition, we will choose Random Forest as this often performs well in many different situations. As we are interesting in assessing the performance of these classifiers we use K-fold cross validation to determine the performance. In K-fold cross validation we divide the complete dataset into K different parts, and use K-1 for training. Then, we use 1 for testing. This is done for K times and the results are averaged. As the data is not balanced we find that this is the best way to assess the performance of the classifiers.

24.4.3 Detection Performance

We define a positive as a DDoS attack frame and a negative as a normal data frame. In terms of classification this can result in 4 different outcomes for each frame:

- True Positive (TP): A DDoS attack frame correctly classified as an DDoS attack frame.
- False Positive (FP): A normal frame wrongly classified as an DDoS attack frame.
- True Negative (TN): A normal frame correctly classified as a normal frame.
- False Negative (FN): A DDoS attack frame wrongly classified as a normal frame.

In practice, you want to maximize the True Positives, this means you correctly detected a DDoS attack. Also, we want to minimize the amount of False Positives. When a normal frame is incorrectly classified as a DDoS attack it will be dropped and therefore communication is useless. To get a view on the efficiency of the detection setup we evaluate it with recall and precision which is given in Equations 24.1 and 24.2.

$$recall = \frac{TP}{TP + FN} \tag{24.1}$$

$$precision = \frac{TP}{TP + FP} \tag{24.2}$$

As mentioned before, detection speed is of high importance, therefore the average detection speed will be defined as the total time needed to perform the complete K-fold cross validation. This gives a solid view on the distinction between the used classifiers and different values for N.

24.5 Results

We can find the results of the different classification for the different frame sizes in Figures 24.2, 24.3 and 24.4. Also, the computation times needed can be found in Figure 24.5. We see that there is no clear distinction between the different frame sizes, we observe some spikes or dips but these are not significant. We rather see that the frame size has no significant impact on the recall and precision for Random forest. There is a large dip for the Neural Network with a frame size of 600. There is no explanation for this. If the three classifiers are compared roughly equally high scores are observed. Random Forest has the highest scores in terms of recall and is similar to Neural Network in terms of precision.

If the computation times are compared, there is a significant difference in the computation time of the Neural Network. These are much higher than Logistic Regression and Random Forest. The latter are both equally small, with small, lower computation times for the Random forest. In all the graphs no significant difference is observable in frame sizes.

24.6 Discussion

The data used in this research was not balanced enough to simulate real-life situations. Normally one would expect a small amount of DDoS attack flows, and a large

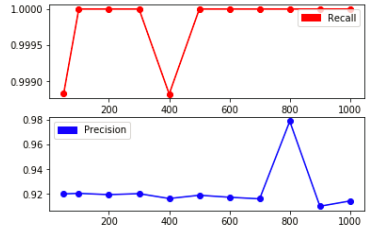

Figure 24.2: Classification results for Random Forest

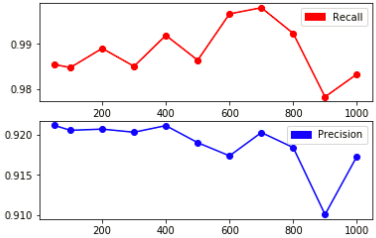

Figure 24.3: Classification results for Logistic Regression

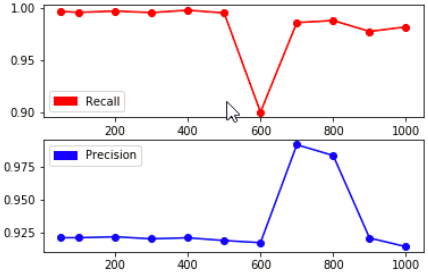

Figure 24.4: Classification results for Neural Network

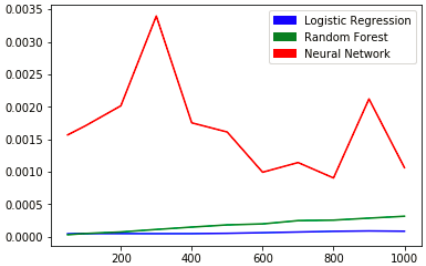

Figure 24.5: Computation times for K-Fold cross validation

amount of normal flows. This was not the case in this research. The classifier scores may therefore slightly deviate from the real life situation. Also, other different attacks such as DNS amplification or NTP amplification should be added to the dataset to enlarge the amount of detectable DDoS attacks. This will have impact on the results so this can also introduce a slight deviation on the real life classifier scores. Furthermore, the frame sizes used in this experiment should have been compared to lower frame sizes, but this was not manageable due to memory and time limits in the used system.

24.7 Conclusion

In this paper an improvement on DDoS anomaly detection using machine learning on flow data by aggregating data is described. Other DDoS attack detection meth-

ods are machine learning algorithms based on package based data, which generates more overhead in both the training and detection phase. An alternative is using flow data for machine learning. Flow data is more lightweight and therefore has a memory and time performance increase. Less information is contained within flow data compared to package based data however, so there is a possibility that classification performance decreases. Different attacks are described and their are clear difference between the given attacks which are used. Machine learning is used to detect DDoS attack in groups of flows. Using different machine learning algorithms and an aggregation on flow data for different values we find that Random Forest, Neural Networks and Logistic Regression all function equally well, while there is a significant computational difference for Neural Networks. The aggregation of network flows does not show a significant performance increase or a precision and accuracy increase.

References

Appneta. (n.d.). *Sample captures.*

Casas, P., Mazel, J., & Owezarski, P. (2012). Unsupervised network intrusion detection systems: Detecting the unknown without knowledge. *Computer Communications*, *35*(7), 772–783.

Chen, Y., Ma, X., & Wu, X. (2013). Ddos detection algorithm based on preprocessing network traffic predicted method and chaos theory. *IEEE Communications Letters*, *17*(5), 1052–1054.

DDoSDB. (n.d.). *Ddosdb.* (https://ddosdb.org/)

Karimazad, R., & Faraahi, A. (2011). An anomaly-based method for ddos attacks detection using rbf neural networks. In *Proceedings of the international conference on network and electronics engineering* (Vol. 11, pp. 44–48).

Kaspersky. (2017). Longer, expanding, demanding: Botnet ddos attacks highlighted in kaspersky lab quarterly report. (https://www.kaspersky.com/about/press-releases/2017_longer-expanding-demanding-botnet-ddos-attacks-highlighted-in-kaspersky-lab-quarterly-report)

Labs, K. (2017). It security risks survey 2017. (https://goo.gl/2QVDSZ)

Lazarevic, A., Ertoz, L., Kumar, V., Ozgur, A., & Srivastava, J. (2003). A comparative study of anomaly detection schemes in network intrusion detection. In *Proceedings of the 2003 siam international conference on data mining* (pp. 25–36).

Mok, M. S., Sohn, S. Y., & Ju, Y. H. (2010). Random effects logistic regression model for anomaly detection. *Expert Systems with Applications*, *37*(10), 7162–7166.

Sekar, V., Duffield, N. G., Spatscheck, O., van der Merwe, J. E., & Zhang, H. (2006). Lads: Large-scale automated ddos detection system. In *Usenix annual technical conference, general track* (pp. 171–184).