Experiments with Reduction of Network Datasets for DDoS Analysis

Veronika Krobotova
CESNET
Prague, Czech Republic
veronika.krobotova@cesnet.cz

Martin Zadnik
CESNET
Prague, Czech Republic
ORCID: 0000-0002-2099-2348

Ondrej Sedlacek
CESNET
Prague, Czech Republic
ORCID: 0009-0004-8670-7396

***Abstract*—DDoS analysis and precise mitigation are still challenges due to more sophisticated DDoS attacks, their growing volume, and the diversity of network traffic itself. The machine learning methods enable automated analysis and subsequent mitigation by learning the legitimate traffic to be able to infer the boundary between the current DDoS and legitimate traffic during an attack. Since processing large packet samples is costly, especially if the sample is used during the DDoS analysis online, this paper assembles and evaluates several pipelines to reduce a large legitimate capture into a compact but representative packet sample for the timely analysis. The quality of the reduction is evaluated statistically and based on the resulting effectiveness of the ML method. The results show that the reduction pipelines produce samples with higher variability and contribute to the creation of boundaries that include a smaller proportion of legitimate traffic during mitigation than when using an unreduced sample of the same size.**

***Index Terms*—reduction, dataset, classification**

I. Introduction

Recent reports show that Distributed Denial of Service (DDoS) attacks have become significantly shorter and more adaptive, posing new challenges for mitigation systems. According to Cloudflare's 2023 Q4 DDoS Threat Report [1], 91% of attacks last less than ten minutes, while Nokia Deepfield [2] highlights the growing prevalence of short, bursty "hit-and-run" attacks designed to avoid detection by traditional defenses. These attacks require fast analysis tools to keep pace with their evasion strategies.

CESNET [3], the national research and education network (NREN) of the Czech Republic, also observes this trend and addresses these challenges by developing and deploying its DDoS protection platform [4]. This platform integrates a variety of DDoS mitigation strategies. Some are deterministic [5], giving administrators complete control over their effect on the traffic, while others are based on machine learning (ML) [6] and deep learning [7]. These strategies raise administrators' fears of the negative impact on the traffic, i.e., inadvertent blocking of legitimate traffic.

The ML methods require representative samples of legitimate and malicious traffic to construct reliable models. However, network engineers often express concerns about the diversity of the legitimate samples. For instance, a five-minute packet sample may fail to capture the variability of normal network behavior, which can differ significantly by hour or weekday. The variability of legitimate traffic is crucial for machine learning–based mitigation, as a diverse set of positive examples helps models distinguish between benign and malicious traffic during the attack and thus reduce false positives that could block legitimate services.

A week-long sample provides greater diversity and a more comprehensive traffic view, but also results in large datasets that are challenging to process efficiently with ML tools. This paper addresses this challenge by designing and evaluating several dataset reduction pipelines to reduce a dataset into a representative but compact packet sample.

Our contributions are threefold: (a) we demonstrate the practical need for larger and more representative benign traffic samples, (b) we show that significant dataset reduction is possible without impact on model quality, and (c) we compare reduction techniques ranging from simple random sampling to more advanced, structured approaches.

We discuss the related work as well as the background motivating our work in Sec. II. Sec. III proposes five pipelines for network dataset reduction. The dataset and results are described in Sec. IV, and the conclusions are discussed in Sec. V.

II. Related work and Background

There are several techniques for data reduction [8]: dimensionality reduction, data compression, and numerosity reduction, either *parametric* methods which work only with the model parameters instead of the whole data, or *non-parametric* methods which work directly with the data.

The reduction of network traffic datasets has been explored in several studies. For example, [9], [10] focus on data compression techniques, while [11]–[14] investigate dimensionality reduction methods for network traffic analysis and intrusion detection.

The problem of reducing the number of instances in a network traffic dataset, while maintaining the statistical properties of the data, is addressed in [15], [16], who combine entropy, Kullback-Lieber distance and Marginal Utility for reduction.

Garg et al. [17] introduced a method to reduce the number of packets in a dataset by using DBSCAN clustering combined with dimensionality reduction, aiming to optimize the performance of Intrusion Detection System (IDS). They

978-3-903176-75-1

concluded their work by comparing the performance of IDS on the original and reduced datasets.

In the context of DDoS mitigation, we consider only non-parametric numerosity reduction applicable. This is because mitigation methods must retain all original dimensions of the traffic data, as attacks may exploit any of these dimensions to evade detection. Moreover, the ML methods are often applied in parallel in DDoS protection platforms. Each preprocesses its network traffic specifically or works directly with the raw packet capture, such as [6], which analyzes legitimate and DDoS samples at the time of the attack to infer specific rules for the attack.

Dimensionality reduction or removing specific packets that may not be relevant to DDoS attacks risks discarding features or packets that are critical for identifying adversarial patterns, especially if the attacker is aware of it.

Lastly, the data compression strategies do not lower the number of samples, thus do not lower the complexity of the automated analysis by ML techniques; on the contrary, they introduce additional complexity during analysis due to the need for decompression.

III. Pipelines of reduction techniques

A total of five reduction pipelines are proposed, the steps of which are depicted in Fig. 1. The first three pipelines start with deduplication. The purpose of deduplication is to reduce the volume of data by identifying and removing redundant data [18]. There are several causes of duplicate packets in network traffic, such as port mirroring, retransmissions in the network [19], but if the payload is removed during the capture, as in our case, two-thirds of duplicates are packets of data-heavy connections. The deduplication key consists of the classical 5-tuple extended with the packet length to preserve the same packets of various lengths. In the last two pipelines, we first group packets based on flows [20] and subsequently work with the flow representation of these packets in the rest of the pipeline. If a flow is selected by the method to be preserved, the first two, one random from the middle, and the last packet of the flow make it to the reduced dataset.

Each pipeline takes a different perspective on the input dataset, dividing the data into groups of varying sizes and levels of detail. Each pipeline is called by its unique step, distinguishing it from the others (Uniform, Services, Subnets, Active IPs, Clusters). Subsequent random sampling within each group uses dual sampling probabilities calculated from the relative representation of each group in the entire dataset, except for the first pipeline (Fig. 1 (a)), where all packets are considered equal and randomly sampled with equal probability. Random sampling, without any prior grouping, allows us to evaluate whether this straightforward pipeline alone can achieve the desired diversity or if it is better to partition the dataset into smaller groups based on specific characteristics before performing the reduction itself.

Dual Sampling

Due to the long tail effect of Internet traffic, we expect that splitting traffic into individual groups will result in a large

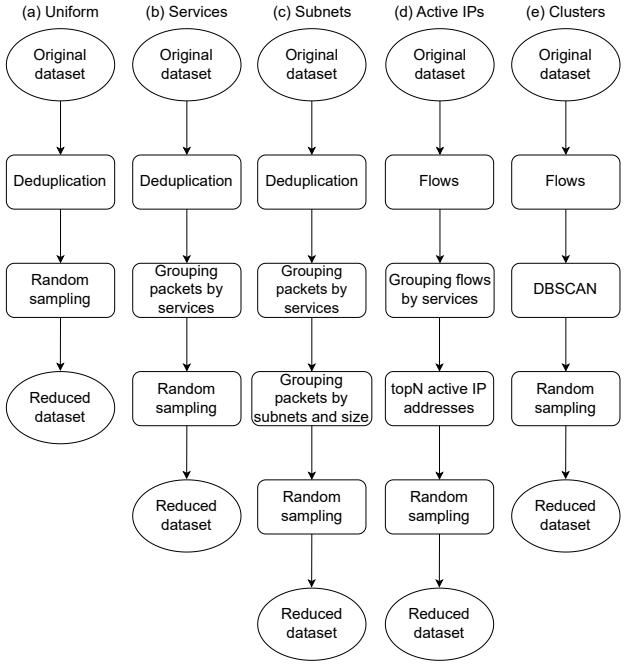

Fig. 1. Diagrams illustrating the structure and steps of the reduction pipelines.

number of small groups with low sampling probability. This leads to a higher probability that these groups will not be represented at all in the resulting dataset, and hence that a given reduction algorithm will not reach the target number of packets. This effect is more pronounced for reduction algorithms that split traffic into smaller units (like subnets-based reduction and cluster-based reduction methods described later in this section) than for other approaches.

For this reason, we decided to introduce the possibility of a dual sampling approach for pipelines that group or cluster traffic. The adaptive sampling probability is calculated on the basis of the relative representation of the group in the data. A fixed sampling probability is used for groups whose representation does not exceed a specified minimum representation threshold within the dataset.

The minimum representation threshold (min_repr) and the value of the fixed sampling probability are configurable to meet the dataset characteristics and the administrators' requirements. This will ensure that small groups are represented in the sample, while also bringing the methods closer to the specified target number of packets. We also expect the optimal value of the minimum group representation parameter to vary between pipelines, depending on how the pipeline divides traffic into groups of different sizes.

The sampling probability for group i, which is above the min_repr threshold, is determined by the following relations:

$$T_i = \text{round}\left(\left(\frac{n_i}{N}\right) \cdot N_t\right), p_i = \frac{T_i}{\tilde{n}_i}$$

where T_i is the target number of sampled objects from

the group i, p_i is the resulting sampling, probability for the group i, n_i is the number of objects in group i in the original dataset, N is the total number of objects in the original dataset, N_t is the target number of objects in the reduced dataset, $\tilde{n}_i$ is the number of available objects in the group i after preprocessing (e.g. deduplication). The aggregated probability does not sum up to one; therefore, we set these parameters based on experiments to reach the target number of packets.

Grouping by services

The representation of individual services is one of the most important requirements for us in the resulting dataset, as we want to avoid misclassifying critical services. Network traffic is classified according to its destination ports. Services running on privileged ports (range 0–1023) are each placed into separate groups, as they represent key services that should be preserved appropriately in the reduced dataset. We also separately place selected popular services on registered ports (1024-49151), drawing from the list of commonly used ports [21]. The remaining TCP/UDP traffic that does not fall into the above categories is placed into one common group. This group includes, for example, traffic from P2P applications using randomly selected dynamic ports (49152-65535) or applications communicating with clients on these ports. We place packets without a transport layer into groups according to their protocols (e.g., ICMP).

Grouping by subnets and sizes

The pipeline (Fig. 1 (c) Subnets) represents a view of the dataset in terms of the packet sizes within each subnet that uses a specific service. The source IP addresses are used to further divide packets into smaller groups based on their subnets. The subnet mask for IPv4 addresses is set to /24, allowing up to 254 hosts per subnet, which suits many common network scenarios. For IPv6 addresses, the prefix length /64 is used, as it is the standard prefix length defined in the IETF document [22]. Within each subnet, packets are divided into three groups based on size. The packet size distribution in network traffic is bimodal, with a large number of small packets under 100 bytes in size and a large number of packets in the 1400–1500 byte range [23]. Based on this distribution, the packets will be divided into the following three size groups (in bytes):

$$S_1 = [0, 99], \quad S_2 = [100, 1399], \quad S_3 = [1400, \infty)$$

TOP-N active IP addresses

This step aims to partially break our requirement and reduce the diversity of traffic in the resulting dataset by filtering the most frequently occurring communications. In this way, we want to examine the impact of reducing towards typical traffic and suppressing edge cases.

Within each group, we identify the *N* most active IP addresses, i.e., those that use the service most frequently. The activity of IP addresses is determined by the number of established connections and the number of flows in which the IP address is the source IP.

We then perform random sampling with sampling probability proportional to the representation of the service in the original dataset, focusing exclusively on flows associated with the *N* most active IP addresses.

Clustering - DBSCAN

The flows are clustered into groups using the DBSCAN algorithm. In selecting an appropriate clustering method, we choose density-based methods because they do not require a predetermined number of resulting clusters. This is important because we need to form clusters according to current traffic characteristics. Moreover, these methods can identify clusters of arbitrary shape, which is important given the unstructured and variable nature of network traffic. Another advantage is the ability to detect outliers, which can be interpreted as suspicious or illegitimate traffic. Such points can then be excluded from the resulting dataset, thereby increasing its quality and reducing the risk of including potentially malicious traffic.

DBSCAN algorithm was subsequently chosen from the density-based methods, primarily because of the availability of a very efficient implementation that allows processing even large data volumes in an acceptable time. Details about the algorithm will be given in Sec. IV.

DBSCAN performs clustering based on statistical flow information, including the total number of packets, the average packet size, and the number of bytes transmitted. These characteristics were selected based on a study by Erman et al. [24], who compared different clustering methods, including DBSCAN, to classify network traffic flows.

We use the Euclidean metric to calculate the distance between flows. Before calculating the distance, we apply a logarithmic transformation to the selected flow characteristics. According to a study by Erman et al. [24], this transformation improves the clustering results when the Euclidean distance is used. The reason is that network characteristics typically have a heavy-tail distribution (a few extreme values are much larger than the rest). The application of the logarithm compresses the range of values and reduces the influence of these outliers, to which the Euclidean distance is sensitive because the calculation involves squaring the difference between the values.

IV. Evaluation

The experiments assess the ability of the reduction pipelines to generate diverse datasets in terms of different attributes and evaluate the performance of an ML method to infer DDoS mitigation rules.

Pipeline Settings

The main parameter of the pipelines is $target\_packets$, which controls local parameters (in the steps) of the pipelines to reach the number of packets in the reduced output dataset. We experiment with 10 k, 100 k, 500 k values.

The parameter min_repr serves to select groups with fewer than the minimum required number of packets. The

corresponding $high\_sampling\_prob$ defines the sampling probability of groups that do not reach the min_repr. We set $high\_sampling\_prob$ at 0.9 and min_repr at 0.01% to ensure sufficient representation of even minority groups based on experiments with multiple values and observing how well the reduced dataset captures original characteristics and how it reaches the target number of packets.

The flow-based traffic reduction pipeline Active IPs contains top_n parameter, which we empirically set to 10 based on the principle of "heavy hitters", where a small number of IP addresses generate the majority of traffic. The value represents a compromise between preserving the most important part of the network traffic and significantly reducing the data volume. As part of the clustering flow-based reduction method, we used the DBSCAN clustering algorithm, for which it was necessary to set appropriate parameter values defining the maximum distance between two points (ϵ), and defining the minimum number of points needed to form a cluster ($MinPts$). The final parameter values, ϵ = 0.02 and $MinPts$ = 3, were set using a k-distance plot and the Silhouette score tracking.

Input dataset

The dataset for our experiments consists of 129 PCAP files[1] (five minutes per hour) affiliated with a single large organization (/16 subnet). Its traffic was collected from March 26th to April 4th, 2025, on the links connecting the Czech National Research and Educational Network to its peers. The capture infrastructure is described in detail in [25]. Although it was not possible to obtain a complete, continuous full packet capture of the link due to capture failures, we consider the available data sufficiently representative for the use case of inferring the DDoS mitigation rules. The overall input dataset (called Original) contains almost 58 million packets. To anonymize the dataset, all /16 IP address prefixes are consistently replaced with randomly generated values, i.e., two IP addresses sharing the /16 subnet in raw dataset share the new /16 prefix in the anonymized dataset.

For each of the $target\_packets$ values (10 k, 100 k, 500 k), we created a comparison dataset containing the corresponding number of consecutive packets from the captured network traffic, without any reduction. These datasets serve as reference samples for comparison with the outputs of the reduction methods to show what would happen if only a short consecutive packet capture is used. We will refer to these datasets as Compare Data with respect to their size.

A. Diversity Preservation

This section focuses on evaluating the ability of each reduction pipeline to preserve key characteristics of the original dataset. At the same time, a comparison is made with reference datasets (Compare Data) of the respective sizes. The pipelines were run ten times per the $target\_packets$, and the medians of each metric were used.

[1]The dataset is available at Zenodo: https://doi.org/10.5281/zenodo.16795107 and the source code is available at Github: https://github.com/Korunka1/packet_redcution_pipelines

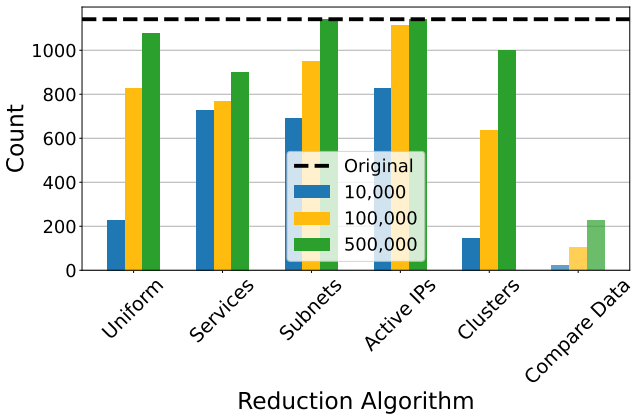

Fig. 2. Unique Service Counts preserved by each method.

Fig. 2 shows a comparison of the median counts of unique services for the different dataset sizes of each pipeline and Compare Data, and also shows the unique count of services in the original dataset (dashed line). The reduced datasets contained more unique services than the Compare Data datasets. The best results in the number of unique services were expected for the Services, Subnets, and Active IPs pipelines, due to the formation of groups specifically based on services.

While Subnets and Active IPs met the expectations, the Services method proved to be less effective with larger datasets. The reason is caused by the dual sampling, where the groups of services were further split into even smaller groups by the additional step (grouping by subnets and sizes, topN active IP addresses, respectively) in the Subnets and Active IPs pipelines. These smaller groups were sampled with $high\_sampling\_prob$, thus providing more diversity. The Compare Data provides only a small portion of the original diversity.

The graph showing the percentage of flows preserved (Figure 3) shows that the Clusters method performed the best, achieving the highest TCP flow preservation rate across all dataset sizes. For example, when reducing to a dataset size of 500k packets, approximately 7% of the original flows were preserved, which is predictable since the method focuses on flows. On the other hand, it is a significant result given the reduction rate of more than 99% with respect to the original dataset.

B. Impact on mitigation

The following experiments demonstrate the impact of reduction on the DDoS mitigation method driven by machine learning [6]. The ML method (decision tree) needs data from benign periods to serve as counterexamples to the traffic during the DDoS period, as described in Sec. II. The reduction pipelines are used to create five legitimate samples (called Uniform, Services, Subnets, Active IPs, Clusters) of approximately 500 k packets each. Half a million packets approximately corresponds to a continuous unreduced reference packet capture lasting 5 minutes (Comp. Data 500).

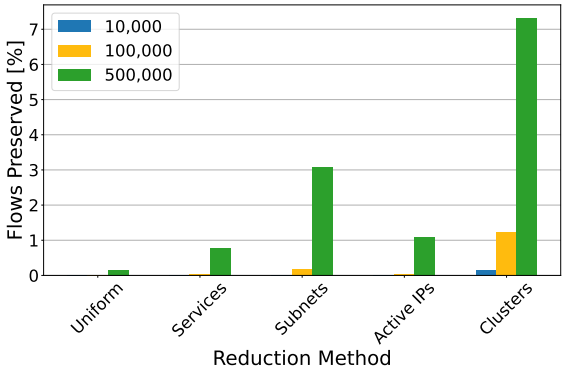

Fig. 3. Percentage of preserved flows for each reduction pipeline and different sizes of the resulting datasets.

TABLE I
MITIGATION RESULTS OVER REDUCED LEGITIMATE DATASETS DURING ICMP ATTACK.

Legit. data	TP [%]	FP [%]	F1	Precision
Uniform	100.00	**19.44**	0.91	0.84
Services	100.00	80.21	0.71	0.55
Subnets	100.00	81.43	0.71	0.55
Active IPs	100.00	32.37	0.86	0.76
Clusters	100.00	**21.68**	0.90	0.82
Comp. Data 500	100.00	**93.26**	0.67	0.52

We prepared SYN, ICMP, and NTP flood attacks as the attack samples. Each attack sample consists of 4,000 packets with randomly spoofed IP addresses so that the mitigation method cannot easily block the attack by its specific source IP address. Additionally, these attack samples are extended with legitimate traffic (captured separately from the original dataset) but marked deliberately as an attack (in the ratio of 40% legitimate traffic, 60% attack traffic). This simulates the real-life situation when there is still legitimate traffic during the attack, and the mitigation algorithm receives a notification to start mitigation from scratch.

The Tables I, II, III show the results of measuring the quality of mitigation rules created by the ML method for mitigating the selected type of DDoS attacks. The tables show the averaged metrics for three runs per each reduction pipeline and the reference sample Comp. Data 500. We mark selected false positive rates in bold to draw attention to the interesting results. Moreover, the presented true and false positive rates are calculated only from the portion of the testing sample that contains the same protocol as the attack traffic (e.g., NTP attack over UDP traffic). This perspective reflects our focus on evaluating how well the reduced dataset helps preserve the portion of legitimate traffic that is most likely to be accidentally blocked together with an attack using the same protocol (the mitigation method always reaches 100% TP and 0% FP for protocols that are not part of the attack).

Based on the design, the best coverage of ICMP traffic was expected to be provided by Services, Subnets and Active IPs methods, which create separate groups for ICMP packets. However, most ICMP packets were contained in the

TABLE II
MITIGATION RESULTS OVER REDUCED LEGITIMATE DATASETS DURING NTP ATTACK.

Legit. data	TP [%]	FP [%]	F1	Precision
Uniform	100.00	7.86	0.96	0.93
Services	100.00	**2.89**	0.99	0.97
Subnets	100.00	3.21	0.98	0.97
Active IPs	100.00	8.58	0.96	0.92
Clusters	99.67	3.82	0.98	0.96
Comp. Data 500	100.00	**26.22**	0.88	0.79

TABLE III
MITIGATION RESULTS OVER REDUCED LEGITIMATE DATASETS DURING SYN ATTACK.

Legit. data	TP [%]	FP [%]	F1	Precision
Uniform	100.00	20.61	0.91	0.83
Services	100.00	18.00	0.92	0.85
Subnets	87.25	27.05	0.81	0.76
Active IPs	80.63	22.43	0.79	0.78
Clusters	100.00	**12.19**	0.94	0.89
Comp. Data 500	97.59	**43.21**	0.81	0.69

datasets of the Clusters, Uniform, and partially Active IPs methods. The reason why these datasets contained a higher number of ICMP packets is due to the 0.02% proportion of ICMP packets in the original dataset, which caused the proportion of ICMP packets in the Service and even Subnet methods to exceed the min_repr threshold. Thus, they were not explicitly set to the high sampling probability value of the $high\_sampling\_probability$ and were reduced. We can observe that the best-performing pipelines are the Uniform and Clusters. The low percentage of false positives is the major indicator for the mitigation method since this means blocking legitimate traffic. The reference Comp. Data 500 achieved the worst results (93% FP), failing to provide enough legitimate ICMP samples to steer the mitigation method from the benign ICMP traffic during the attack. The significant difference between the best reduced datasets and the Comp. Data 500 clearly demonstrates the need for representative datasets of legitimate traffic.

The NTP amplification attack tests the ability of the techniques to capture less frequent traffic, specifically packets with source port 123 in Table II. The Services and Subnets pipeline performed the best, while the Uniform and Active IPs missed the low number of NTP traffic, which resulted in a higher number of false positives. The Services pipeline benefited from preserving most of the NTP traffic as a service group, which was sufficiently small to be sampled with $high\_sampling\_probability$. The Subnets pipeline, as the Services pipeline extension, achieved low FP, closely followed by the Clusters.

In the case of SYN flood in Table III, significantly better results were expected due to the higher number of SYN packets in the reduced datasets compared to the other experiments. The comparison of legitimate and attack packets showed that the SYN attack packets were very similar to those of legitimate ones, and the DDoS inference method had to base its mitigation on very specific fields, such as specific TTL,

checksums, ack numbers, etc. The Clusters preserved most of these characteristics and allowed the mitigation method to build the best mitigation rule.

The experiments for the 100k dataset brought similar results with only slightly worsened false positive rate while the true positive rate remained at 100% for the best-performing reduction pipelines, indicating the 100k dataset is still a feasible option. In the case of the 10k dataset, the false positive rate increased above 50% even for the best reduced dataset; therefore, we consider the 10k dataset as not sufficient to provide enough representative legitimate samples for the mitigation method.

Using the whole original dataset leads to a highly imbalanced number of legitimate examples, causing the ML method to preserve legitimate traffic and ignore the small DDoS sample, which yields near-zero TPs.

While mitigating each attack individually may not be complex enough, we also conducted an experiment in which six attack vectors (DNS, LOIC, NTP, UDP, TORSHAMMER, SYN, and HULK) were combined. Although the results are not shown, they are consistent with the findings from experiments involving individual attacks.

V. Conclusion

The main goal of this work was to design a pipeline of reduction techniques that can significantly reduce the number of packets in the large network traffic dataset while maintaining its quality in terms of representative and diverse samples. Our work was motivated by the practical experience of network administrators who argued that the short traffic sample cannot contain enough diversity to provide sufficient information for automated traffic analysis. The results show that the concern regarding the short traffic sample was well-founded, and indeed, a short traffic sample achieved the worst results when compared to the reduced datasets. Among the evaluated pipelines, the Clusters pipeline consistently achieved superior results across all tested attacks, although it is resource-intensive. When computational resources are limited, the Uniform pipeline offers strong overall performance. In scenarios where attacks focus on rare traffic, service-focused pipelines yield the best outcomes; however, their performance is inconsistent and poor in other cases.

For future work, we aim to develop pipelines that achieve results comparable to the Clusters while requiring significantly fewer computational resources. We will also research extensions regarding timing traffic dependencies, supporting the analysis of sequences in network communication.

Acknowledgement

We thank the Liberouter team for their support with the Traffic Capture Interface. Special thanks go to Matúš Mihaljevič for his insight into deduplication and experiments with smaller datasets.

References

[1] J. P. Omer Yoachimik, "4.2 tbps of bad packets and a whole lot more: Cloudflare's q3 ddos report." online, 2024.
[2] J. Meyer, "Five-minute chaos: Why short-lived ddos attacks pack a bigger punch than you think." Nokia Deepfield blog, Dec. 2024.
[3] CESNET, "The cesnet3 network." online, 2025.
[4] CESNET, "Adaptivní ochrana proti ddos útokům." online, 2022.
[5] P. Goldschmidt and J. Kučera, "Defense against syn flood dos attacks using network-based mitigation techniques," in *IM 2021 - 2021 IFIP/IEEE International Symposium on Integrated Network Management*, 06 2021.
[6] M. Žádník and E. Carasec, "Ai infers dos mitigation rules," *Journal of Intelligent Information Systems*, vol. 60, no. 1, pp. 305–324, 2022.
[7] P. Goldschmidt and J. Kučera, "Windower: Feature extraction for real-time ddos detection using machine learning," in *NOMS 2024-2024 IEEE Network Operations and Management Symposium*, pp. 1–10, 2024.
[8] J. Han, M. Kamber, and J. Pei, "3 - data preprocessing," in *Data Mining: Concepts and Techniques* (J. Han, M. Kamber, and J. Pei, eds.), The Morgan Kaufmann Series in Data Management Systems, pp. 83–124, Boston: Morgan Kaufmann, third edition ed., 2012.
[9] P. Almasan, K. Rusek, S. Xiao, X. Shi, X. Cheng, A. Cabellos-Aparicio, and P. Barlet-Ros, "Leveraging spatial and temporal correlations for network traffic compression," 2023.
[10] Y. Liu, D. Towsley, T. Ye, and J. C. Bolot, "An information-theoretic approach to network monitoring and measurement," in *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement*, IMC '05, (USA), p. 14, USENIX Association, 2005.
[11] T. Huang, H. Sethu, and N. Kandasamy, "A new approach to dimensionality reduction for anomaly detection in data traffic," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 651–665, 2016.
[12] K. Keerthi Vasan and B. Surendiran, "Dimensionality reduction using principal component analysis for network intrusion detection," *Perspectives in Science*, vol. 8, pp. 510–512, 2016. Recent Trends in Engineering and Material Sciences.
[13] Y. Geng, S. Cai, S. Qin, H. Chen, and S. Yin, "An efficient network traffic classification method based on combined feature dimensionality reduction," in *2021 IEEE 21st International QRS-C*, pp. 407–414, 2021.
[14] A. Shiravani, M. H. Sadreddini, and H. N. Nahook, "Network intrusion detection using data dimensions reduction techniques," *Journal of Big Data*, vol. 10, no. 27, 2023.
[15] A. Botta, A. Dainotti, A. Pescapè, and G. Ventre, "Reducing network traffic data sets," in *Proceedings of IEEE International Conference on Communications*, pp. 350–356, 06 2007.
[16] A. Pescape, "Entropy-based reduction of traffic data," *IEEE Communications Letters*, vol. 11, no. 2, pp. 191–193, 2007.
[17] S. Garg, R. Singh, M. S. Obaidat, V. K. Bhalla, and B. Sharma, "Statistical vertical reduction-based data abridging technique for big network traffic dataset," *International Journal of Communication Systems*, vol. 33, no. 4, p. e4249, 2020. e4249 IJCS-19-0891.R1.
[18] X. Zhang and M. Deng, "An overview on data deduplication techniques," in *Information Technology and Intelligent Transportation Systems* (V. E. Balas, L. C. Jain, and X. Zhao, eds.), (Cham), pp. 359–369, Springer International Publishing, 2017.
[19] M. Jonker, R. Hofstede, A. Sperotto, and A. Pras, "Unveiling flat traffic on the internet: An ssh attack case study," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pp. 270–278, 2015.
[20] C. Systems, "Cisco ios flexible netflow command reference." online, 2008.
[21] N. House, "Common ports cheat sheet: The ultimate list." online, 2024.
[22] B. E. Carpenter, T. Chown, F. Gont, S. Jiang, A. Petrescu, and A. Yourtchenko, "Analysis of the 64-bit Boundary in IPv6 Addressing." RFC 7421, Jan. 2015.
[23] D. Murray and T. Koziniec, "The state of enterprise network traffic in 2012," in *2012 18th Asia-Pacific Conference on Communications (APCC)*, pp. 179–184, 2012.
[24] J. Erman, M. Arlitt, and A. Mahanti, "Traffic classification using clustering algorithms," in *Proceedings of the 2006 SIGCOMM Workshop on Mining Network Data*, MineNet '06, (New York, NY, USA), pp. 281––286, Association for Computing Machinery, 2006.
[25] D. Soukup, J. Pešek, L. Hejcman, D. Beneš, and T. Čejka, "Tci: A system for distributed network monitoring, troubleshooting and dataset creation," in *NOMS 2024-2024*, pp. 1–6, 2024.