Feature Selection Evaluation towards a Lightweight Deep Learning DDoS Detector

Odnan Ref Sanchez*, Matteo Repetto†, Alessandro Carrega*, Raffaele Bolla*‡ and Jane Frances Pajo*‡
*CNIT - S2N National Laboratory, Genoa, Italy
Email: {odnan.sanchez, alessandro.carrega}@cnit.it
†IMATI - CNR, Genoa, Italy
Email: matteo.repetto@ge.imati.cnr.it
‡DITEN, University of Genoa, Italy
Email: raffaele.bolla@unige.it, jane.pajo@tnt-lab.unige.it

ICC 2021 - IEEE International Conference on Communications | 978-1-7281-7122-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICC42927.2021.9500458

***Abstract*—Today's networks and services undoubtedly require a high level of protection from cyber threats and attacks. State-of-the-art solutions that implement Machine Learning (ML) have shown to improve the accuracy and confidence in threat detection compared to previous approaches, making it suitable for detecting today's sophisticated attacks such as Distributed Denial of Service (DDoS). However, in real-world deployments, input data streams take large bandwidth and processing capacity, especially for Deep Learning (DL) solutions that require extensive input data. On the other hand, deployment environments usually have limited bandwidth and computing resources, such as in the Internet of Things (IoT). Thus, a lightweight detection solution that satisfies such constraints is needed. In this paper, we utilize a feature reduction approach for our DL-based DDoS detector based on the Analysis of Variance (ANOVA), which is used to identify important data features and reduce the data inputs needed for detection. Our result shows that we can reduce the data input needed by up to 84.21% while only reducing 0.1% detection accuracy. We also provide a detailed analysis of the characteristics of DDoS attacks using ANOVA and compared our work with recent DL-based DDoS detection systems to demonstrate that our results are comparable to existing approaches.**

***Index Terms*—ANOVA, DDoS Detection, Deep Learning, Feature Selection**

I. Introduction

With the advent of digitalization, the number of data-driven digital services is expected to rise significantly to meet the increasing customer demands [1]. These services feed on user data, then process and integrate them with other services to extract much more valuable information. In this respect, cyber threats and attacks that disrupt on-going services are also on the rise. Among others, Denial of Service (DoS) and Distributed DoS (DDoS) attacks have become prevalent with an increased projection from the previous years [2]. Valuable services, when stopped, could result in the loss of large sums of profit. Thus, early detection and prevention of such attacks are critical topics of interest.

With the recent advancement in Machine Learning (ML), numerous studies have shifted from rule-based detection (e.g., Ingress Filtering [3]) to ML-based detection strategies for the detection of DDoS attacks. More specifically, Deep Learning (DL) methods have been recently developed [4], which is suitable for more complex data input and more sophisticated pattern recognition.

However, DL approaches are cumbersome in real-world deployments [5]. One of the main limitations is the availability of resources in the deployment environments. For instance, in the Internet of Things (IoT) environments, the memory, bandwidth, and computational capacity are limited, especially on the ultra low-powered gateways. Among others, such resource constraints are one of the reasons why a centralized security architecture has been proposed in literature [1], which executes the heavy computation in a centralized security server for the security of digital services. In the centralized approach, a lightweight agent is deployed in the local environment, which has the responsibility of monitoring and streaming the traffic information needed for attack detection. Deployed local agents also have to respect the resource limitations while being able to provide large streams of network traffic data.

Thus, this work aims to identify and use only the needed input data for the detection of DDoS attacks by using the Analysis of Variance (ANOVA) statistical method [6]. ANOVA is widely used in feature selection (e.g., [7]) since it measures the significance of numerical features depending on the difference between the means of the groupings of these features according to the target vector (class label).

We developed a DL-based detector using the DDoS datasets provided by the Canadian Institute of Cybersecurity (CIC). Then, we used the results from ANOVA to significantly reduce the input data. The evaluation results show that we can achieve this target with a minuscule trade-off with detection accuracy, allowing for a major cutback in resource utilization and data input streams from traffic monitoring agents. Also, we compared our detector to current DL solutions, which achieved similar detection performance.

This paper is organized as follows. Section II presents the details of the DDoS detector and explains the flow of the experiments. Section III provides a detailed data analysis using the ANOVA statistical method, discussing the important data features for detection. Section IV presents the evaluation results of the DL detector and the trade-off between accuracy and data inputs, as well as the comparison to current DL solutions. Finally, our concluding remarks are discussed in Section V.

978-1-7281-7122-7/21/$31.00 ©2021 IEEE

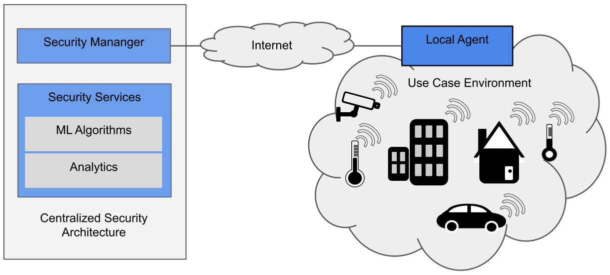


Fig. 1: Simplified overview of the DL Detector data pipeline

II. Deep Learning for DoS/DDoS

In this section, an overview of the DL architecture is briefly discussed, followed by the description of the DDoS datasets, the DL model training specifics, and the evaluation metrics.

A. Architecture

The overview of the architecture is depicted in Figure 1. It shows a simplified view of the main data pipeline for the detection of attacks. The DL detector is part of the centralized security framework that provides programmable security appliance solutions in the advent of heterogeneous digital services managed by different service providers, which are leveraged by both GUARD [8] and ASTRID [9] European Projects. The detailed discussion for this architecture can be found in [1].

We show the figure, including the centralized framework, to point out the locally deployed agent on the client-side. It is responsible for the extraction of features for attack detection. For this study, specifically, this agent extracts network traffic parameters and statistics (e.g., eBPF, Netflow, nProbe, CICflowmeter) to feed it to the DL attack detector. Depending on the agent's capabilities, the collected traffic information can range from simple traffic logs, flow statistics to more fine-grained information such as individual packet information. In theory, these agents can generate large streams of data, which could strain the dedicated resource, especially in limited power and bandwidth setting (e.g., IoT environments). Thus, this work aims to reduce the input data needed without sacrificing much of the detection accuracy.

B. Dataset Description and Features

Table I shows the denomination of the datasets used in this study which are taken from the CIC, University of New Brunswick (UNB). Only the datasets with DoS/DDoS attacks were extracted which include ISCXIDS 2012 [10], CICIDS 2017 [11], CSE-CIC-IDS 2018 [11], and CICDDoS 2019 [12].

The ISCXIDS (2012) [10] dataset contains real traffic traces that include Internet Relay Chat (IRC) Botnet DDoS. The dataset comes with flow statistics, generated by using the IBM QRadar appliance together with the attack labels, indicating whether the flow is benign or an attack. We used the CICflowmeter tool [13] to generate the flow features from the PCAP file and extracted the label information from the output of the IBM QRadar tool. We used data features from CICflowmeter to be consistent with the other datasets and obtained 69,520 total samples as shown in Table I.

TABLE I: Denomination of samples used from CIC Datasets

Dataset	Date	attack type	attacks	benign
ISCXIDS 2012 [10]	15/06/2010	IRC Botnet	34760	34760
CICIDS 2017 [11]	07/07/2017	LOIC (TCP)	128025	97686
	05/07/2017	Hulk	231073	440031
	05/07/2017	GoldenEye	10293	-
	05/07/2017	slowloris	5796	-
	05/07/2017	SlowHTTPtest	5499	-
CSE-CIC-IDS 2018 [11]	20/02/2018	LOIC (HTTP)	125130	124914
	21/02/2018	HOIC (TCP)	400	77876
	21/02/2018	LOIC (UDP)	360	-
	15/02/2018	GoldenEye	8851	2334
	15/02/2018	Slowloris	2417	-
	16/02/2018	Hulk	30287	97040
	16/02/2018	SlowHTTPTest	30391	-
CICDDoS 2019 [12]	12/01/2019	DNS	27065	2690
	12/01/2019	LDAP	12798	1280
	12/01/2019	MSSQL	15981	1573
	12/01/2019	NTP	115149	11454
	12/01/2019	NetBIOS	13530	1374
	12/01/2019	SNMP	12072	1198
	12/01/2019	SSDP	6016	615
	12/01/2019	UDP	17085	1702
	12/01/2019	Syn	3025	320
	12/01/2019	TFTP	202160	20250
	12/01/2019	UDP-lag	29635	2989
	12/01/2019	WebDDoS	39	-

The CICIDS (2017), CSE-CIC-IDS (2018), and CICDDoS (2019) are the latest datasets provided by UNB that include multiple attacks. These datasets have real data traces in PCAP format and also have flow features generated by the CICflowmeter. The attacks include DoS such as DoS Hulk, GoldenEye, slowloris, and slowHTTPtest. It also includes DDoS attacks generated by the DDoS Low Orbit Ion Cannon (LOIC) [10], a tool used for network stress testing that can also be used to deploy attacks. Finally, protocol-based DDoS attacks are also included, as reported in Table I.

The extracted network flow samples of all the datasets contain 76 features that consist of traffic flow statistics. The main features are the number of packets flows in both forward and backward direction (abbreviated as fwd and bwd in the figures), Inter-Arrival Times (IAT) of the packets, packet length information, header flag counts, and header information together with their simple statistical parameters such as minimum, maximum, average and standard deviation. For the complete description, the reader can refer to the CICflowmeter documentation [13].

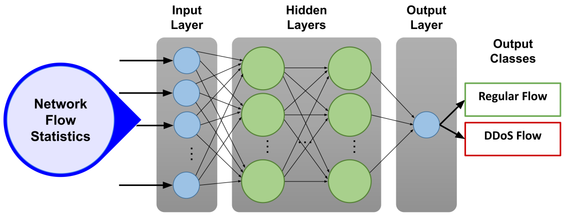
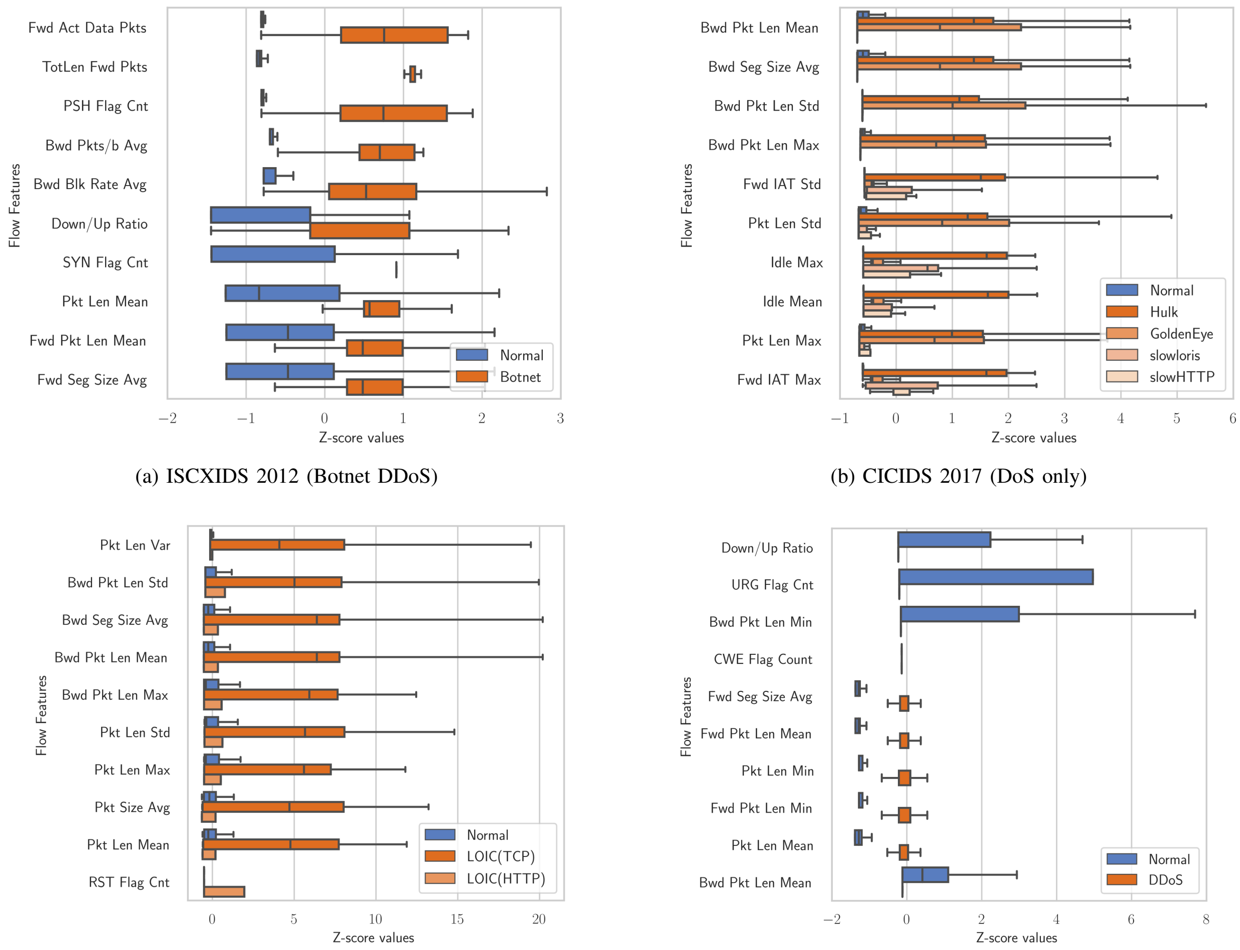


Fig. 2: The Deep Learning model configuration

C. Deep Learning Approach

We split each dataset and use 75% for the training and 25% for the final testing data. In the training phase, 10% of the



(a) ISCXIDS 2012 (Botnet DDoS)

(b) CICIDS 2017 (DoS only)

(c) DDoS LOIC TCP (2017) and HTTP (2018)

(d) CICDDoS 2019 (DDoS)

Fig. 3: Ranking features using ANOVA and showing the boxplot of the regular and attack flows

training data was used for validation. Then, we scaled the data using z-score normalization [14] to convert each feature into a zero-mean and unit-variance distribution using the scikit-learn's standard scaler function[1].

Neural Networks (NNs) are inspired by biological neurons, which consist of a multi-layer collection of nodes. A basic NN consist of the input layer, a single hidden layer, and an output layer. Recently, NNs having more than a single hidden layer became popular given their ability to capture more complex patterns [15]. In this respect, we present a lightweight 4-layer DL solution with each layer having the number of nodes equal to the number of features (i.e., 76) except for the output, as depicted in Figure 2. The DDoS detector uses the flow statistics as the input data and outputs a binary decision, whether a specific flow is associated with a DDoS attack or a regular flow.

We use the Tensorflow[2] module for developing the DL model. The model has a total of 11,781 trainable weights and biases. Upon training, we used binary cross-entropy loss function, Adam optimizer to update the network weights, a fixed learning rate of 0.001, and a training batch size of 64 samples.

The standard evaluation metrics for binary classification, which include Accuracy (Ac), Precision (Pr), and Recall (Rc), are used in this study. Additionally, F1-score (F1) is also used, which is a harmonic mean measure between Pr and Rc [16].

III. Feature Analysis

We used ANOVA statistical method [6] to identify and rank the most important features, which is widely used in feature selection [7]. It forms groupings according to the categorical label and measures the variance between the means of the groupings. Features with larger variance indicate better separability, which means they are more appropriate for detection.

Figure 3 shows the top 10 most important features according to ANOVA for the different sets of attacks. The features are arranged in decreasing order of importance starting from the top. The figures show the boxplot of the normalized values, where the mean of the original values is centered at zero.

[1] StandardScaler function: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

[2] Tensorflow: https://www.tensorflow.org/

Then, for each feature, we show the regular and DDoS traffic separately.

Figure 3a shows the IRC Botnet's ten most important features, having the active forward packets (packets with more than 1 byte of TCP data), forward packet length, and flag count for the PUSH flag as the most significant. It clearly shows a large difference between the normal and DDoS values, where the normal traffic values are very close to the mean (i.e., zero) and almost invariant while the attack has a large variance. During the attack, the IRC Botnet executes multiple downloads using the HTTP GET request and uses the PUSH flags to force the TCP to send the TCP segment immediately without waiting for the buffer to become full.

The most important features for the DoS attacks are shown in Figure 3b. The four most important features are the packet length (mean, standard deviation, and maximum) and average segment size of the backward packets. Thus, it is the size of the packets that differentiates DoS attacks from normal flows particularly for Hulk and GoldenEye. Moreover, the list indicates that IAT and idle times also provide significant importance in detecting DoS attacks.

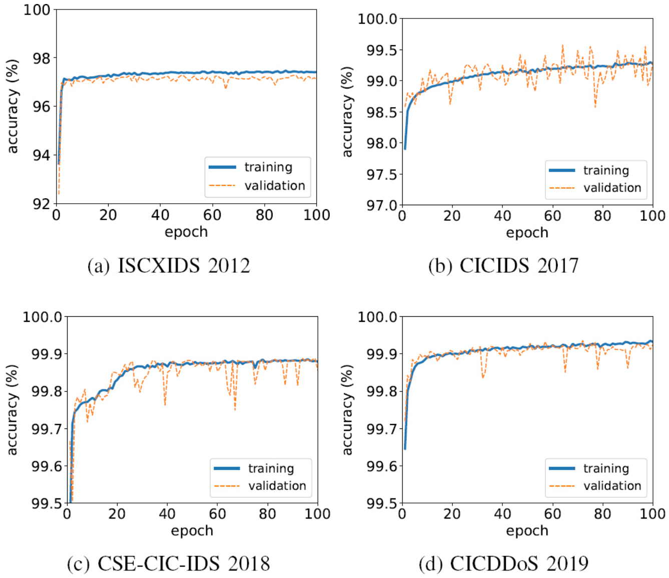
Regarding the attack differences, Hulk and GoldenEye tend to have similar properties. They mainly execute through a large volume of backward packet length. On the other hand, slowloris and slowHTTPtest also have similar properties and are opposite to GoldenEye and Hulk. These attacks operate through maintaining open connections with the server with minimal bandwidth [11] for a long duration and exhaust server resources resulting in DoS. IAT and active/idle times show the most important features for both slowloris and slowHTTPtest, as confirmed by ANOVA.

Figure 3c shows the most important features for the LOIC family. In the figure, we show the LOIC TCP and HTTP attacks and computed the most important features altogether. Packet features in the backward direction achieved the most significant features, which are shown clearly for LOIC TCP.

Figure 3d shows the most important features for all the DDoS attacks in the 2019 dataset. In this mixed data, the ratio of download and upload resulted as the most significant. The values are inverted from the rest, meaning the regular flows are more variant and have values higher than the mean value. On the other hand, attack flows are more invariant and are usually smaller than the mean value. This is true since attacks have a very large upload rate, making the download versus upload ratio very small. Also, the importance of forward packet features in detecting the 2019 DDoS attacks is confirmed by ANOVA as it includes the minimum, mean, and average segment size of the forward packets in the top features. Figure 3d shows that these features have value ranges that do not overlap, thus, we expect high detection rates for the DL models given the large separability.

IV. Experimental Results

In this section, we present the evaluation of the DL detectors during the training period and final testing phase, followed by the discussion of the feature reduction results. Then, we show their comparison with existing DL approaches. For this purpose, separate models are built for each dataset.



(a) ISCXIDS 2012

(b) CICIDS 2017

(c) CSE-CIC-IDS 2018

(d) CICDDoS 2019

Fig. 4: Model training vs validation accuracy

The experiments were conducted using a 32-core server node with 64 GB of RAM. We used Python's Tensorflow and scikit-learn[3] modules for the development of DL models.

TABLE II: Evaluation of the DL attack detector

Dataset	Ac (%)	F1 (%)	Pr (%)	Rc (%)
ISCXIDS (2012)	97.473	97.483	97.156	97.811
CICIDS (2017)	99.328	99.198	98.497	99.908
CSE-CIC-IDS (2018)	99.872	99.872	99.771	99.972
CICDDoS (2019)	99.932	99.962	99.964	99.960

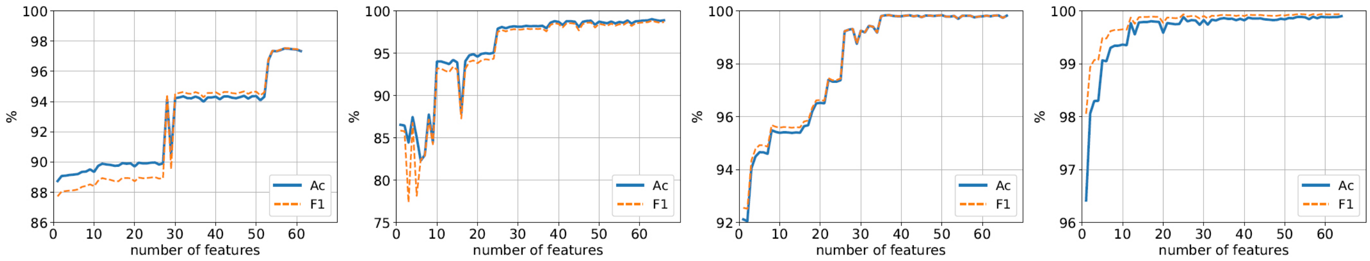
A. DL Performance Evaluation

Figure 4 shows the evaluation of the models during the training phase, showing both the training and validation accuracy for 100 epochs. All the training accuracies steadily increased while there were no large gaps observed between the training and validation accuracies that may constitute overfitting.

Figure 4a shows that the model built from the 2012 dataset performed the least, reaching 97.47% of training accuracy while the model built from the 2019 dataset achieved best as shown in Figure 4d, reaching 99.93% of training accuracy. The final training accuracy for 2017 and 2018 datasets also reached 99.32% and 99.87%, respectively.

For the final test evaluation, the Ac, F1, Pr, and Rc are shown in Table II. The models have achieved high Ac across all the test datasets. Similar to the training phase, detecting the Botnet achieved the lowest Ac. Nonetheless, the model still detected 97.81% (Rc) of all the attack flows associated with the 60-min Botnet attack, which is more than enough to

[3] https://scikit-learn.org/stable/index.html



(a) ISCXIDS 2012

(b) CICIDS 2017

(c) CSE-CIC-IDS 2018

(d) CICDDoS 2019

Fig. 5: Feature reduction evaluation using ANOVA

raise an alarm during an actual event. The rest of the models achieved high detection performance, reaching over 99% Ac.

B. Feature Reduction

The main aim of this work is to reduce the input features and, hence, the resource consumption without sacrificing much of the accuracy. The previous results were conducted using all the available 76 features. Here, we show the importance of feature analysis and reduction, selecting subsets of features with the results of ANOVA from Section III.

We start the subset selection by using the best input feature given by ANOVA and iteratively add the next best feature in the rank (i.e., forward selection approach). For each subset of data, a DL model is built and trained up to 10 epochs. Figure 5 shows the results of feature reduction for each of the CIC datasets, mapping the Ac and F1 for each subset of data.

The graphs show that utilizing all the features is not needed to reach optimum detection. For instance, in detecting IRC Botnet, it is enough to use the 54 most important features to achieve almost the same detection accuracy as shown in Figure 5a, reducing the input data needed by 29%. Furthermore, using only the 28 best features yields 94.15% Ac and 94.46% F1, which trades off 3.23% Ac for 63.15% of input reduction.

For the 2017 dataset, utilizing only the 25 best features already yields 97.89% Ac and 97.51% F1, as shown in Figure 5b. It means that a slight decrease of 1.43% in accuracy yields 67.11% of input data reduction with respect to using all features. The graph also shows that it can be further reduced to the 10 best features, reaching 94.02% Ac and 93.19% F1. The decrease of 5.3% in detection accuracy has lead to an 86% reduction of input data.

For the 2018 and 2019 datasets, the single best feature already yields high accuracy, as shown in Figures 5c and 5d, respectively. Detecting DDoS using only the average segment size of forwarding packets already achieved 92.12% Ac and 92.55% F1 for the 2018 dataset. It decreases the accuracy by 7.75% but heavily reduces the input by 98.6%. For the 2019 dataset, the download/upload ratio has been the best feature, as explained in Section III. Utilizing this single feature already achieved 96.41% Ac and 98.05% F1, which is only 3.5% less than the optimum accuracy but with a 98.6% reduction of input data. These findings are similar in the Border Gateway Protocol (BGP) scenario, where some features can individually detect anomalies and hijacks that result in DDoS [17], [18].

If high accuracy still needs to be respected and only a small margin is permitted, the best trade-off that maximizes accuracy can still be achieved. For the 2018 dataset, utilizing 26 and 35 features achieved 99.22% and 99.82% Ac, which reduces input data by 65.78% and 53.94%, respectively, and only sacrificed a minuscule fraction of 0.64% and 0.04% from the original accuracy. For the 2019 dataset, using the best 12 features achieved 99.83% Ac and 99.90% F1, which means that a reduction of only 0.1% accuracy from the best-case scenario results in a reduction of 84.21% of the input data. These results are similar to the feature reduction techniques in [19], where they achieved up to 68% data reduction while trading only 0.03% on accuracy.

TABLE III: Comparison using ISCXIDS 2012

Year	Detector	ML	Ac	F1	Pr	Rc
-	**DLDDoS Detector**	**DNN**	**97.47**	**97.48**	**97.15**	**97.81**
2020	LUCID [4]	CNN	98.88	98.89	98.27	99.52
2018	TR-IDS [20]	CNN+RF	98.09	-	-	95.93
2017	DeepDefense [21]	LSTM	98.41	98.40	98.34	98.47

C. Literature Comparison

Furthermore, we compare the results from our models, which we call DLDDoS, with the existing approaches in the literature that also utilized the CIC datasets. The same set of literature studies is also used in [4] for benchmark comparison.

Table III shows the detectors using the IRC Botnet as the test set. These recent studies utilize NN approaches such as Convolutional NNs (CNN) for LUCID [4] and Long-short Term Memory (LSTM) for DeepDefense [21]. TR-IDS [20] used CNN for the initial stage of processing unstructured data and used Random Forests for the final classification.

The table shows that our results are comparable to current approaches, having only around 1% lesser on Ac. As explained in Section II-B, the monitoring tool used for this dataset was IBM QRadar, but we used CICflowmeter features to be consistent with the new datasets. Thus, some flows were not labellable due to the conversion. Although we also performed tests with the original flow features from IBM QRadar, which

TABLE IV: Comparison using CICIDS 2017

Year	Detector	ML	Ac	F1	Pr	Rc
-	**DLDDoS Detector**	**DNN**	**99.33**	**99.20**	**98.50**	**99.91**
2020	LUCID [4]	CNN	99.67	99.66	99.39	99.94
2019	Deep Learning [22]	CNN+LSTM	97.16	-	97.41	99.10
2018	DeepGFL [23]	RF	-	94.05	92.62	95.52

obtained 99.02% Ac and 99.03% of F1 using the same methods, outperforming other approaches.

LUCID [4], DeepGFL [23], and DL approach from [22] utilized the attacks for the 2017 dataset, which is reported in Table IV. Similar to our result, LUCID achieved over 99% Ac, F1, Pr, and Rc. We took the best results from the DL models developed in [22], which used the CNN+LSTM model and reached 97.16% of detection accuracy. For the DeepGFL, we present the result for detecting DoS Hulk, which reached 94.05% of F1 using Random Forests.

TABLE V: Comparison using CSE-CIC-IDS 2018

Year	Detector	ML	Ac	F1	Pr	Rc
-	**DLDDoS Detector**	**DNN**	**99.87**	**99.87**	**99.77**	**99.97**
2020	LUCID [4]	CNN	99.87	99.87	99.84	99.89

For the 2018 dataset, the results of this study are compared with LUCID having identical performance, which is reported in Table V. Thus, we conclude that all our results are comparable with the existing literature. Finally, we have not yet found existing literature that utilizes the recent 2019 dataset for comparison as of the time of writing.

V. Conclusion

The goal of the paper is to reduce the input data to scale down the computation and bandwidth usage of monitoring agents without sacrificing much of the detection performance, which was obtained using the ANOVA statistical method. We developed DDoS detectors using DNNs and achieved high detection performance. We achieved a reduction of 63.15%, 67.11%, 65.78%, and 84.21% of the input data features for 2012, 2017, 2018, and 2019 datasets, respectively, reducing only 3.23%, 1.43%, 0.64%, and 0.1% accuracy. Our results are comparable to existing approaches in the literature that utilize DL methods with the same datasets. With these results, traffic monitoring agents can significantly reduce their live data streams in real-world deployments.

The choice of the DL approach can be improved in future work. For instance, reinforcement learning can be a suitable alternative given the difficulty of acquiring real-time labeled data. Also, federated learning can be explored for use cases in which edge devices have computing capability.

Acknowledgment

This work has received funding from the European Commission under Grant Agreement 786922 (H2020 ASTRID) and 833456 (H2020 GUARD).

References

[1] M. Repetto, A. Carrega, and A. Duzha, "A novel cyber-security framework leveraging programmable capabilities in digital services," in *Italian Conf. on CyberSecurity (ITASEC)*, 2020.

[2] F. Donovan, "DDoS attacks increase in size, frequency and duration," 2020. [Online]. Available: https://securityintelligence.com/articles/avoid-ddos-attacks/

[3] A. K. Soliman, C. Salama, and H. K. Mohamed, "Detecting DNS reflection amplification DDoS attack originating from the cloud," in *13th Int. Conf. Comput. Eng. Syst. (ICCES)*, 2018, pp. 145–150.

[4] R. Doriguzzi-Corin et al., "LUCID: A practical, lightweight deep learning solution for DDoS attack detection," *IEEE Trans. Netw. Service Manag.*, 2020.

[5] B. Sliwa, N. Piatkowski, and C. Wietfeld, "LIMITS: Lightweight machine learning for IoT systems with resource limitations," in *2020 IEEE Int. Conf. Commun. (ICC)*, June 2020.

[6] D. A. Freedman, *Statistical models: theory and practice*. Cambridge University Press, 2009.

[7] M. Sheikhan et al., "Modular neural-SVM scheme for speech emotion recognition using ANOVA feature selection method," *Neural Comput. Appl.*, vol. 23, no. 1, pp. 215–227, 2013.

[8] "GUARD cybersecurity framework." [Online]. Available: https://guard-project.eu/

[9] "ASTRID (addressing threats for virtualised services)." [Online]. Available: https://www.astrid-project.eu/

[10] A. Shiravi, H. Shiravi, M. Tavallaee, and A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Computers & Security*, vol. 31, pp. 357–374, 2012.

[11] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *4th Int. Conf. Inf. Syst. Security Privacy (ICISSP)*, 2018, pp. 108–116. [Online]. Available: https://registry.opendata.aws/cse-cic-ids2018/

[12] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy," in *2019 Int. Carnahan Conf. Security Technol. (ICCST)*, 2019, pp. 1–8.

[13] A. Lashkari, Y. Zang, G. Owhuo, M. Mamun, and G. Gil, "CICflowmeter (formerly iscxflowmeter)- a network traffic flow analyzer." [Online]. Available: https://github.com/CanadianInstituteForCybersecurity/CICFlowMeter

[14] A. Jain, K. Nandakumar, and A. Ross, "Score normalization in multimodal biometric systems," *Pattern Recognition*, vol. 38, no. 12, pp. 2270–2285, 2005.

[15] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT Press Cambridge, 2016, vol. 1.

[16] M. Sokolova, N. Japkowicz, and S. Szpakowicz, "Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation," in *Australasian Joint Conf. on AI*. Springer, 2006.

[17] R. Fontugne, A. Shah, and E. Aben, "AS hegemony: A robust metric for AS centrality," in *Proc. SIGCOMM Posters and Demos*, 2017, pp. 48–50.

[18] O. R. Sanchez et al., "Comparing machine learning algorithms for BGP anomaly detection using graph features," in *Proc. 3rd ACM CoNEXT Workshop on Big Data, ML and AI for Data Commun. Netw.*, 2019.

[19] M. Aamir and S. M. A. Zaidi, "DDoS attack detection with feature engineering and machine learning: the framework and performance evaluation," *Int. J. Inform. Security*, vol. 18, no. 6, pp. 761–785, 2019.

[20] E. Min, J. Long, Q. Liu, J. Cui, and W. Chen, "TR-IDS: Anomaly-based intrusion detection through text-convolutional neural network and random forest," *Security and Communication Networks*, vol. 2018, 2018.

[21] X. Yuan et al., "DeepDefense: identifying DDoS attack via deep learning," in *IEEE Int. Conf. SMARTCOMP*, 2017, pp. 1–8.

[22] M. Roopak, G. Y. Tian, and J. Chambers, "Deep learning models for cyber security in IoT networks," in *IEEE 9th Annual Comput. Commun. Workshop and Conf. (CCWC)*, 2019, pp. 0452–0457.

[23] Y. Yao, L. Su, and Z. Lu, "DeepGFL: Deep feature learning via graph for attack detection on flow-based network traffic," in *IEEE Military Commun. Conf. (MILCOM)*, 2018, pp. 579–584.