2019 International Conference on Computing, Networking and Communications (ICNC): Network Algorithms and Performance Evaluation

An empirical study of intelligent approaches to DDoS detection in large scale networks

Xiaoyu Liang
Department of Computer Science
University of Pittsburgh
Pittsburgh, USA
veronica@cs.pitt.edu

Taieb Znati
Department of Computer Science
University of Pittsburgh
Pittsburgh, USA
znati@cs.pitt.edu

***Abstract*—Distributed Denial of Services (DDoS) attacks continue to be one of the most challenging threats to the Internet. The intensity and frequency of these attacks are increasing at an alarming rate. Numerous schemes have been proposed to mitigate the impact of DDoS attacks. This paper presents a comprehensive empirical evaluation of Machine Learning (ML)-based DDoS detection techniques, to gain better understanding of their performance in different types of environments. To this end, a framework is developed, focusing on different attack scenarios, to investigate the performance of a class of ML-based techniques. The evaluation uses different performance metrics, including the impact of the "Class Imbalance Problem" on ML-based DDoS detection. The results of the comparative analysis show that no one technique outperforms all others in all test cases. Furthermore, the results underscore the need for a method oriented feature selection model to enhance the capabilities of ML-based detection techniques. Finally, the results show that the class imbalance problem significantly impacts performance, underscoring the need to address this problem in order to enhance ML-based DDoS detection capabilities.**

***Index Terms*—Empirical Evaluation, DDoS Detection, Machine Learning Techniques, Class Imbalance Problem**

I. Introduction

With the advent of new computing paradigms, such as Cloud computing, and the emergence of pervasive technology, such as the Internet of Things, Distributed Denial of Services (DDoS) attacks have been growing dramatically in frequency, sophistication and impact, making it one of the most challenging threats in the Internet [1]. Recent attack events demonstrate the severe impact of DDoS attacks and deepens security concerns [2]–[4].

DDoS attacks are typically launched from a very large number of distributed, remotely controlled devices, organized into botnets and aimed at attacking the targeted computing resources, including computer systems, network devices, servers and web applications [5]. Numerous schemes have been proposed to mitigate the impact of DDoS attacks. The early schemes utilize stochastic analysis to monitor network traffic flows' behavior and exploit the entropy of network traffic to identify normal behavior and detect anamalous intrusion events. Recently, the trend to mitigate the impact of DDoS attacks is to incoroporate "intelligence" into the defense strategy, leveraging Machine Learning (ML) techniques to classify and detect malicious traffic. A number of research works have studied DDoS attack detection with ML-based solutions [6]–[8]. These works often focus on specific schemes, with the aim to carry out a comparative analysis of their performance. Although informative of the state-of-the-art in intelligent techniques to detect DDoS attacks, the intricacies of the studied schemes add significant complexity to the analysis, and often affect the outcome. Furthermore, the focus on specific schemes constrain the applicability of the results to other schemes. Lastly, very little research work addressed "holistically" the performance evaluation of intelligent solutions to DDoS defense. As such, the outcome of these studies remains inconclusive.

A closer look at intelligent DDoS defense schemes reveals that at the core of the proposed schemes is a set of commonly used ML-based "building blocks" for DDoS detection and prevention. In this paper, we take a different approach than existing DDoS performing assessment schemes, focusing the impact of incorporating ML techniques to the DDoS defense. To this end, a representative class of intelligent techniques, which capture main trends in DDoS detection, is carefully selected. An empirical comparative analysis of these selected techniques is then carried out. The objective is to provide insights into the performance of the selected schemes and answer the question of which technique, if any, outperforms the remaining ones. The main contributions of this paper can be summarized as follows:

- The development of a framework for a comprehensive evaluation of the selected techniques, including a specific set of evaluation metrics, to assess the accuracy, sensitivity and specificity of these techniques.
- The development of a rich set of attack scenarios, using a combination of real-world attack and legitimate traffic data from different sources. The scenarios are engineered to exercise a wide spectrum of DDoS attack aspects and characteristics.
- The development of a series of experiments, using different attack scenarios, to explore the impact of packet- and flow-level features, and observable traffic proportions on the performance of the selected techniques. A specific focus of these experiments is on the impact of the common "Class Imbalance Problem", in terms of the ratio

978-1-5386-9223-3/19/$31.00 ©2019 IEEE

between legitimate and attack traffic, on the performance of these techniques. In practice, this problem is common in various disciplines, including anomaly detection. To the best of our knowledge, this is the first empirical study that addresses the impact of Class Imbalance Problem on DDoS detection performance.

The rest of the paper is organized as follows. Section II briefly reviews related works. Section III illustrates the experimental framework, including the selected ML-based techniques, evaluation metrics, and datasets. Section IV reports the experiments and results. Section V summarizes the proposed work and findings.

II. Related Works

A number of surveys provide an extensive classifications of both DDoS attacks and defense mechanisms. Mirkovic and Reiher proposed one of the earliest complete taxonomies of DDoS attacks and defense mechanisms [9]. More recent surveys, built on these taxonomies, focusing on different perspective of DDoS attacks and defense mechanisms are provided in [6]–[8], [10], [11]. Following the taxonomy proposed in [9], a review of various defense mechanisms in each category is provided in [6]. They also survey a collection of DDoS detection methods, summarize their challenges, and propose future directions in dealing with DDoS. In [7], [8], focus is on DDoS attacks in cloud environment. Osanaiye *et al.* provide a taxonomy of DDoS attacks in cloud computing, and review a number of defense schemes based on their deployment locations and detection techniques. Somani *et al.* organize DDoS defense into three stages: prevention, detection and mitigation. They review proposed solutions in each stage, and highlight their challenges [8]. In [10], the authors provide an extensive classification for DDoS flooding attacks, and review a number of defense mechanisms, accordingly. In [11], the authors review several DDoS defense schemes that leverage the architecture of Software-Defined Network (SDN), and also discuss the vulnerabilities of SDN itself to DDoS attacks. Overall, they provide a comprehensive understanding of DDoS attacks and depict the current solution space. There are also surveys that focus on network intrusion and anomaly detection, without specifically addressing DDoS attacks [12]–[15].

The aforementioned works mostly focus on specific DDoS defense methods. As such, they do not provide a comprehensive analysis of ML-based DDoS detection schemes' capabilities. Such an endeavor is not realistic, given the large number of these schemes. Rather than focusing on the capabilities of specific schemes, we take a different approach in this work to analyze the capability of a selected set of building blocks that are commonly shared among ML-based detection schemes.

III. Experimental Framework

In this section, we describe how the evaluated techniques are selected, discuss the evaluation metrics that are used to assess the performance of selected techniques, and introduce the datasets used in the experiments.

TABLE I
Detection Techniques Taxonomy

Categories	Techniques
Classification	Support Vector Machine (SVM)
	Artificial Neural Network (ANN)
	Decision Tree (DT)
	Naive Bayes (NB)
Clustering	K-Means
Nearest-Neighbor Based	K-Nearest Neighbor (KNN)
Statistical	D-WARD

A. Selected Machine Learning Techniques

To select a representative set of ML-based techniques, a thorough search of the academic publications using Google Scholar, IEEE Xplore and Science Direct, is carried out. The search space was limited to the publishing period ranging from 2008 to 2018. Additionaly, only publications with high citation numbers were considered. Using this process, 49 academic publications are included in this work. These publications are used to extract the building blocks underlying their DDoS detection schemes. Furthermore, to gain better understanding of the capabilities of ML-based techniques compare to classical techniques, we include D-WARD [16], a statistics based DDoS defense scheme, into this study. Table I lists the evaluated techniques, and categorizes them into four groups.

B. Evaluation Metrics

A successful DDoS detection requires correctly identifying attacks, while minimizing the number of false alarms. In this study, we adopt the evaluation metrics widely used to assess the performance of DDoS detection schemes. These metrics are: Accuracy, Sensitivity, and Specificity.

$$\begin{aligned} \textit{Accuracy} &= \frac{TN+TP}{TN+TP+FN+FP} \\ \textit{Sensitivity} &= \frac{TP}{TP+FN} \\ \textit{Specificity} &= \frac{TN}{TN+FP} \end{aligned}$$

Accuracy measures the ratio between the correctly identified traffic and the total traffic. Sensitivity measures the ratio between the attack traffic that is correctly identified and the total attack traffic. Specificity measures the ratio between the legitimate traffic that is correctly identified and the total legitimate traffic. In short, sensitivity and specificity measure how well the technique performs for one traffic category, whereas accuracy measures how well the technique performs for both traffic categories. The four outcomes of a detection scheme, True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN), are described in the confusion matrix, shown in Table II.

C. Datasets

In order to carry a meaningful and fair comparative analysis, an ideal benchmark that closely reflects the real-life network traffic, with clearly labeled legitimate and attack traffic, is

TABLE II
CONFUSION MATRIX

		Prediction	
		Attack	Legitimate
Known	Attack	TP	FN
	Legitimate	FP	TN

necessary. In this study, we combine two widely accepted and extensively used benchmarks. The first benchmark is CAIDA DDoS dataset, collected from an actual DDoS attack event [17]. The most significant advantage of this dataset is that the traffic consists of a real-world DDoS attack scenario. Moreover, DDoS is exclusively recorded in this dataset, thus making it appropriate for evaluating intrusion detection schemes solely for DDoS. However, the non-attack traffic was removed from the dataset. To augment the CAIDA benchmark with legitimate traffic, a second benchmark, DARPA dataset [18], is used. The DARPA benchmark contains total of 5 weeks traffic. The first and third weeks do not contain any attacks.

The attack traffic, obtained from the CAIDA benchmark, is split into 14 sets, each containing roughly five minutes traffic traces. The legitimate traffic is selected from DARPA first week attack-free dataset. We use tcpreplay [19] to replay the attack and legitimate traffic, and recapture the combined traffic for analysis. We use $D = \{d_k(I), 1 \leq k \leq 14\}$ to denote the traffic datasets, where I is the size of time interval, which is set to be 5 minutes, and $d_k(I)$ represents the kth time interval traffic.

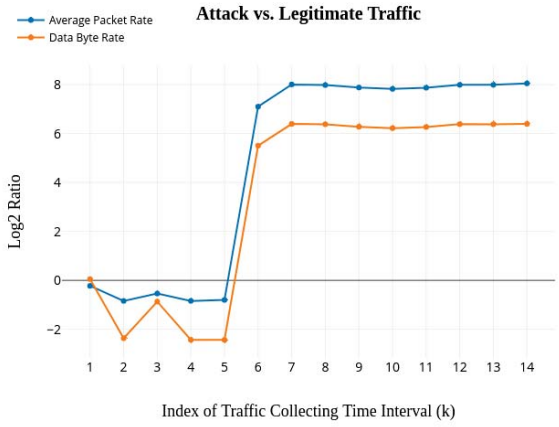

Fig. 1. Ratio of Attack traffic to Legitimate Traffic

Fig. 1 displays the ratio of attack traffic to legitimate traffic for each dataset $d_k(I)$, in terms of average packets rate (packets/second) and Byte rate (Bytes/second). For $k < 6$ (the first 25 minutes of the data captured), the average legitimate packet and Byte rates are larger than those of attack traffic, which translates into negative values in a log_2 scale. Starting from $k = 6$ (the 6th 5-minute period), the attack traffic volume increases dramatically. The changing traffic patterns depict different DDoS attack phases.

TABLE III
FEATURE SETS DESCRIPTION: DETAILED CONTENTS OF PACKET-HEADER BASED AND FLOW-LEVEL BASED FEATURES FOR TCP AND ICMP.

Protocols / Feature Types	TCP	ICMP
Packet Headers	srcIP, srcPort, dstIP, dstPort, Len, TTL, Seq, FIN, SYN, RST, PSH, ACK, URG, ECE, CWR, checksum	srcIP, dstIP, Len, TTL, Type, Code, icmpID, checksum
Network Flows	duration, rtt, protocol, srcIP, srcPort, dstIP, dstPort, iflags, uflags, isn, tag, pkt, oct, end-reason	

In all experiments, the focus is on TCP and ICMP protocols, where significant attack traffic is observed. UDP traffic is discarded due to the limited number of UDP attack traffic observed in the CAIDA dataset.

D. Feature Sets

DDoS detection schemes usually collect network traffic through passive network monitoring. The collected traffic is then analyzed to identify attack traffic. There are two general approaches for passive network monitoring. One approach is packet capture, which intercepts and logs network data packets, various tools can be used to gather data packets, including tcpdump, and wireshark. The other approach is network flow monitoring, which provides aggregated traffic statistics for a flow between two end points. As such, two feature sets, packet- and flow-level based, are considered for analyzing the performance of DDoS detection techniques.

Table III summarizes the packet- and flow-level based features. In this study, a flow is identified as a unidirectional sequence of packets, which share the same 5-tuple values, namely source IP address, source Port number, destination IP address, destination Port number, and Protocol ID.

In this work, the detection performance of ML-based techniques on both feature sets are studied. Since D-WARD requires specific features for the detection, we do not apply different feature sets to D-WARD.

IV. EXPERIMENTS AND RESULTS

In this section, we present three experiments used to assess the capabilities of the different ML-based techniques for DDoS detection. The results of these experiments are then discussed.

A. Experiment 1 – Comparative Analysis

The goal of experiment 1 is to conduct a comparative analysis of the overall performance of the selected techniques. In this experiment, we assume that the entire network traffic is available for each DDoS detection technique. Although the assumption is usually impractical, it provides a strong base to evaluate and compare the detection capabilities of these techniques. The benchmark datasets, D, is divided into T_r and T_s, for training and testing purposes, respectively.

$$T_r = \{d_1(I)\};$$
$$T_s = \{d_2(I), d_3(I), d_4(I), ..., d_{14}(I)\}$$

Classification models are usually trained offline with historical data. The trained model is then applied online to classify future data. In our benchmark datasets, $d_1(I)$ contains the network traffic that is captured during the first 5 minutes of the monitoring interval. It is considered to be the historical data and used to form T_r. T_s is composed of the traffic captured over the remaining 13 5-minute intervals. The 13 traffic sets are tested independently for each evaluated technique, using the accuracy, sensitivity and specificity metrics. Boxplots are used to illustrate the assessed performance, so that the performance variation of these evaluated DDoS detection techniques is also quantified.

1) TCP Traffic: Results for TCP traffic using packet-header features are shown in Fig. 2. The results show that DT outperforms all other techniques, with respect to the defined performance metrics. Furthermore, the results show that DT performs consistently across different testing sets. These results show the high potential of rule-based strategies to efficiently detect DDoS TCP attack traffic in the packet level. In this experiment, NB and RBF-SVM exhibit the worst performance when it comes to distinguishing attack packets from legitimate traffic packets.

The NB classifier is a simple classifier based on Bayes' theorem, assuming a strong independence among the features. As such, given the packet-header features, the NB classifier is biased towards labeling most samples as attack traffic during the testing phase. This results in an extremely high sensitivity score, but a poor specificity score.

The RBF kernel maps features into an infinite dimensional space to solve non-linearly separable samples. This may lead to a lose of generalization, if the training samples are underrepresented. In other words, the trained model fits the training samples too closely, causing the model to become very sensitive to the input data. Fig.3 presents the performance of RBF-SVM in each test case. For the first four test cases k, ($k = 2, 3, 4, 5$), RBF-SVM shows a good performance. However, for test cases k, $k \geq 6$, its accuracy and sensitivity scores are significantly decreased. Recall that in the attack event, presented in Fig.1, the attack volume increases significantly starting from $k = 6$. The increase in volume affect the distribution of traffic attributes, which in turn causes the underlying patterns of the input dataset to become significantly different from that was learned during the training phase.

Focusing on the sensitivity score, the results achieved by SVM, Poly-SVM, KNN and KMeans, shown in Fig. 2, are comparable to those achieved by DT. They all outperform D-WARD. However, the poor specificity scores of the these ML-based techniques suggest a potentially high rate of false alarms which can incorrectly prevent legitimate users from accessing resources.

Results for TCP traffic using flow-level features are shown in Fig.4. DT no longer demonstrates the observed superiority over all other ML-based techniques. Moreover, its accuracy and sensitivity scores are highly variable across all datasets. Examining closely its performance on each test case, DT suffers the over-fitting problem, similar to the pattern of the RBF-SVM behavior, shown in Fig. 3. Additionally, using flow-level features, which represent features from the aggregated packet data between a source and a destination, significantly improves NB's accuracy and specificity. The performance consistency of NB across multiple test cases is also enhanced by using flow-level features. In comparison with packet-header features, flow-level features provide statistic attributes that capture the traffic behavior. Hence, the distribution assumption and the probability inferring of NB classifier is more reasonable.

Finally, it is worth noting that the traditional method, D-WARD, performs competitively in comparison to ML-based

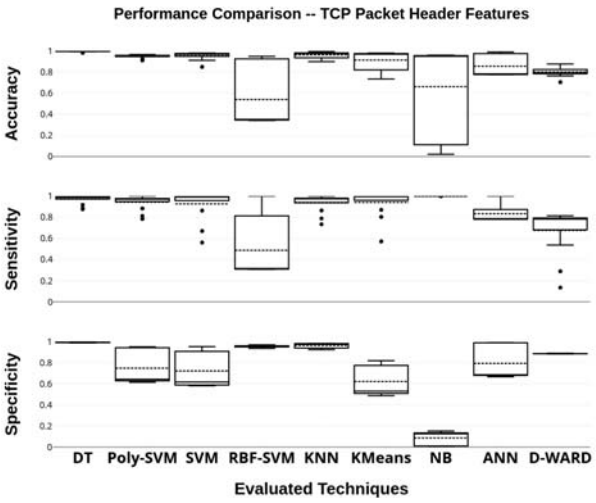

Fig. 2. Performance Comparison–TCP Traffic with Packet Header Features Boxplots are used to illustrate the performance of each evaluated technique. Values of minimum, maximum, median (solid line in box), mean (dot line in box), first quartile, and third quartile are displayed by the box for the 13 test cases. X-axis presents each evaluated detection technique. Subplots present Accuracy, Sensitivity and Specificity, respectively.

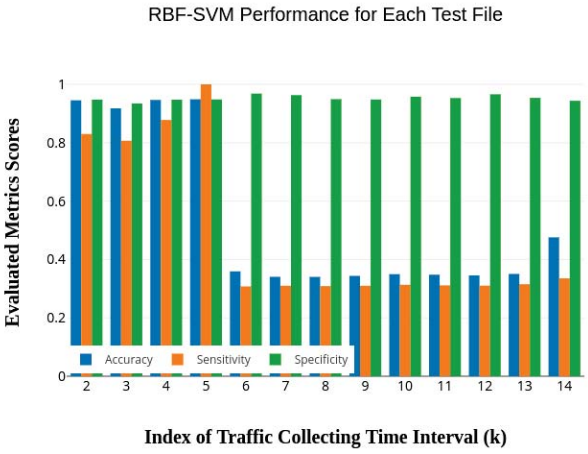

Fig. 3. RBF-SVM Performance Using Packet Header Features X-axis presents the index of the time interval (k), and Y-axis presents the evaluated metrics values. Accuracy, sensitivity and specificity are presented in different colors.

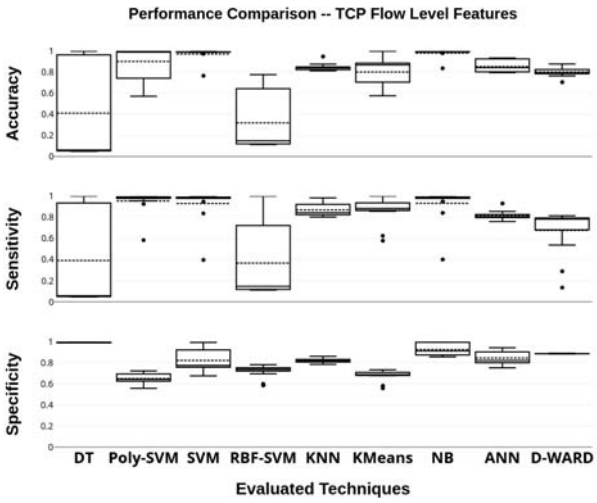

Fig. 4. Performance Comparison–TCP Traffic with Flow Level Features

techniques for all metrics. However, D-WARD fails to identify attack traffic in two datasets, as indicated by the poor sensitivity scores for these two cases. Overall, it can be concluded that ML-based techniques can achieve high performance in detecting TCP attack traffic. Furthermore, ML-based techniques outperform D-WARD in most test cases.

2) ICMP Traffic: Results for ICMP traffic using packet-header features are shown in Fig. 5. The results show that all ML-based techniques achieve near optimal values with respect to all performance metrics, but only for some datasets. On the other hand, the performance consistency of D-WARD is relatively higher. For ICMP traffic, D-WARD detects attack by monitoring the paired messages of ICMP requests and the corresponding replies. If the monitored ratio exceeds the threshold, the alarm is raised. The results show that this simple mechanism works effectively. Features that capture this type of information should be able to improve the performance of ML-based techniques.

Results for ICMP traffic, using flow-level features, exhibit a similar performance pattern as the one in TCP traffic, underscoring the fact that using sophisticated features does not necessarily improve the performance of all ML-based techniques. As indicated by the results, only specific ML-based techniques, such as Poly-SVM and NB, are improved. The use of these features did not significantly improve the performance of the other schemes. In some cases, the performance of these latter schemes has decreased.

In summary, the outcome of this experiment shows that it is not clear that a single technique outperforms all others in all test cases, especially when focusing on the ICMP traffic. The experiment shows that different techniques perform better when using certain types of features, suggesting that feature selection should be method specific. Furthermore, the capability of detecting attack traffic shown by ML-based techniques is evident. On the other hand, the performance inconsistency exhibited by ML-based techniques in dealing with different types of attack traffic raise doubts about their ability to efficiently detect DDoS attack in real world scenarios.

B. Experiment 2 – Impact of Observable Traffic Proportions

In experiment 1, it is assumed that the entire network traffic is available for each DDoS detection scheme. In practice, however, it is infeasible for a detection scheme to have access to the entire network traffic. Actually, the detection scheme is usually deployed on, or attached with, routers or switches. Consequently, a detection scheme can only observe network traffic passing through the network device on which it is deployed. The purpose of experiment 2 is to investigate the ability of a detection scheme to only access a limited portion of network traffic.

To emulate a realistic network environment, we randomly select a proportion p of the total traffic for testing. The selected traffic is then analyzed by the detection techniques, and the performance for each metric is evaluated. Two selection criteria, namely packet- and flow-level, are used to generate a specified portion, p, of the network traffic. For packet-level, a proportion, p, of the total traffic packets is randomly selected without any consideration of the flows to which the selected packets belong. For flow-level, however, a proportion, p, of the total traffic flows is randomly selected and only packets belonging to these flows are made available to the DDoS detector.

The same training data set T_r, used in experiment 1, is also used for this experiment. The traffic proportion p is selected from a set $P = \{1\%, 5\%, 10\%, 20\%, 50\%, 75\%\}$. We do not use separate symbols to differentiate packet- and flow-level datasets, since they are structurally the same. The random traffic selection of DDoS detector observed traffic is repeated 10 times, with different random seeds, to avoid data bias. The dataset used for testing is denoted by D_{exp2}, where $D_{exp2} = d^p_{k,r}(I)$, where $1 \leq r \leq 10$, $p \in P$ and $2 \leq k \leq 14$. In total, 780 test cases are used to assess the performance of

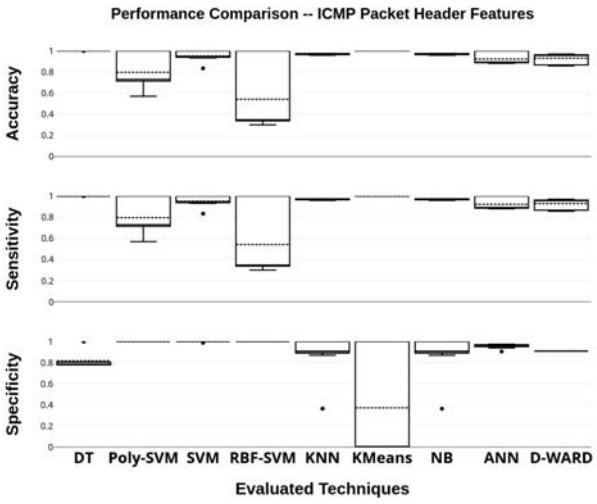

Fig. 5. Performance Comparison–ICMP Traffic with Packet Header Features

each DDoS detection technique, for both packet- and flow-level selections.

To quantitatively evaluate the impact of the observable traffic proportions, Pearson Correlation Coefficient test is applied to measure the strength of the linear correlation between each evaluated metric and the observed proportion. For both packet- and flow-level traffic selections, the performance of D-WARD presents a strong correlation with the increasing observed traffic proportion. Conversely, only a weak correlation between observed traffic proportions and performance is exhibited by ML-based techniques. Table IV shows the correlation scores for flow-level traffic selection. ML-based techniques present even weaker correlation in packet-level traffic selection. Due to the limited space, results of packet-level selections is omitted.

Recall that traditional detection techniques, represented by D-WARD, usually infer network status through monitoring two-way traffic. As such, D-WARD gains a relatively complete picture of the network status given a higher proportion of observed traffic packets or flows. This leads to a more accurate attack detection. Ideally, D-WARD should be deployed at the only boarder router, so that both directions of flows can be observed by the detector. However, this is impractical, a limitation also reported by the authors of D-WARD [16]. Comparing to the traditional detection method, the weak correlation, presented by ML-based techniques, shows that the deployment location does not impact the performance of ML-based detection techniques.

TABLE IV
CORRELATION COEFFICIENT SCORES BETWEEN OBSERVABLE TRAFFIC PROPORTIONS AND PERFORMANCE – FLOW-LEVEL TRAFFIC SELECTION

Techniques	Correlation Scores between Traffic Proportions and		
	Accuracy	Sensitivity	Specificity
DT	0.0549	-0.0159	0.0253
SVM	-0.0177	0.2045	0.0147
RBF-SVM	-0.2732	-0.2384	0.2769
Poly-SVM	-0.2473	0.0666	-0.0661
KNN	-0.1760	0.0885	0.01744
KMeans	-0.0912	0.0300	-0.1349
NB	-0.0946	0.1870	-0.4023
ANN	-0.1283	-0.0456	-0.0857
D-Ward	0.7703	0.8055	0.6329

C. Experiment 3 – Impact of the Class Imbalance Problem

The class imbalance problem is frequently encountered in practice, where the number of observations of one class is far less than the other class. When this problem occurs in the testing phase, accuracy alone is no longer enough to assess the performance of the detection scheme. Different types of evaluation metrics, such as sensitivity and specificity used in this work, need to be used to complement the accuracy to better assess performance [20]. When the class imbalance problem occurs in the training dataset, it may hinder the learning process of classification algorithms [21]. Practically, if the imbalanced class distribution in the training dataset matches the native class prevalences in the test scenario, then the dataset bias in the learning process can be neglected. Yet, the described scenario does not apply to DDoS attack detection.

From the perspective of a DDoS attack detector, attack traffic usually represents a very small subset of all network traffic it observed, particularly in stealth attack. However, when an attack happens, the attack traffic may become the majority class among the traffic that is observed by the detector. Hence, it is common that the detection scheme is dealing with highly imbalanced dataset, and the dominant class is non-stationary. Represented in our benchmark, the attack traffic is the minority class during the training phase, but it becomes the majority class after the attacker increases the attacking volume, shown in Fig. 1.

To assess the impact of the class imbalance problem in the training datasets, we generate a set of training data with different degrees of imbalance. We apply a simple random under-sampling method to create five subsets from the training dataset T_r. Each subset contains 70,000 packets, and the percentage of attack traffic in each subset is drawn from $\{10\%, 30\%, 50\%, 70\%, 90\%\}$. All ML-based techniques are trained with each subset independently. Five models are then built for each technique. Due to the limited number of ICMP legitimate traffic, only TCP traffic is considered in this experiment. For testing, we apply five trained models on the testing dataset T_s from experiment 1. The results show that the family of SVM techniques exhibits the strongest sensitivity to the imbalanced training datasets, while other techniques are affected slightly.

To further study the correlation between the class imbalance of the training data and the performance, we generate nine subsets from T_s with different ratios of attack to legitimate traffic packets. Each subset contains 100,000 packets, and the percentage of attack traffic in each subset is drawn from $\{10\%, 20\%, 30\%, ..., 90\%\}$. The procedure of generating testing samples is repeated 10 times. In total, 90 test cases are used to assess the impact of the imbalanced and balanced training models on the performance of each technique.

Fig. 6 and 7 display the results of linear SVM and RBF-SVM. Trained with small percentage of attack traffic, all SVM models tend to label most samples as legitimate traffic. Using a relatively balanced training dataset, the results show that the robustness of the model is improved. It is worth noting, however, that a balanced dataset does not necessarily achieve the best performance. A sophisticated kernel, such as RBF-SVM, can identify the hidden information by mapping features into higher dimensions. Specifically, RBF-SVM outperforms linear SVM when extremely imbalanced training data is used, although its overall performance remain less than optimal. On the other hand, the sophisticated kernel exhibits higher sensitivity to the data balancing, making it difficult to optimize its performance without degrading the robustness of the model.

In summary, the results clearly show that the impact of the class imbalance problem in datasets should not be neglected. Carefully designing the training process, analyzing the application scenario and choosing the appropriate method are critical for a successful intelligent DDoS detection scheme.

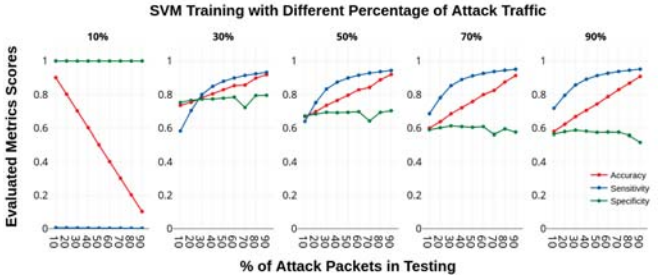

Fig. 6. Class Imbalance Problem Analysis – Linear SVM
The subtitle of each figure describes the percentage of attack traffic in the training sets. X-axis presents the percentage of attack traffic in the testing sets. Different evaluated metrics are represented in different colors.

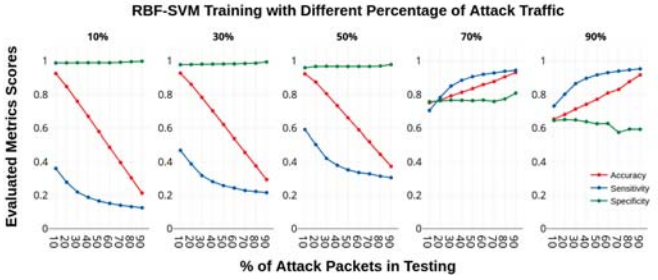

Fig. 7. Class Imbalance Problem Analysis – RBF-SVM

Additionally, detection DDoS attacks in dynamically changing environment remains a challenge for ML-based detection methods.

V. Conclusion

In this work, we conduct a series of experiments to explore the advantages, limitations and influential factors for ML-based DDoS detection techniques. Instead of studying specific solutions, our work focus on empirically evaluating the "building blocks", which are commonly shared among intelligent DDoS detection schemes. A comparative analysis of the overall performance of these techniques is carried out, to provide a better understanding of the techniques' capabilities to detect DDoS attacks. Although the comparative results show that no single technique that outperforms all others in all test cases, the detection capabilities exhibited by ML-based techniques are evident. Additionally, different techniques exhibit different preferences over feature types, emphasizing the significance of feature selection and suggesting that feature selection should be model oriented. A sensitivity analysis illustrates the influence of the observed traffic proportions on the performance of these techniques. As expected, the observed traffic proportions severely impact the performance of traditional detection methods that rely on monitoring the two-way traffic, while ML-based techniques display weak correlation with the proportion of the observed traffic. Lastly, we explored the impact of the class imbalance problem on the performance of ML-based techniques. The results show that the impact of the class imbalance problem should not be underestimated, especially with respect to the dynamically evolving nature of DDoS attacks. Future work can be focused on investigating an ensemble of intelligent schemes, strategically distributed across the network, using an appropriate feature selection model for an adaptive and efficient DDoS detection.

References

[1] "DDoS attacks in Q2 2018." [Online]. Available: https://securelist.com/ddos-report-in-q2-2018/86537/
[2] "2016 Dyn cyberattack," accessed: Aug.2018. [Online]. Available: https://en.wikipedia.org/wiki/2016_Dyn_cyberattack
[3] M. Broersma, "UK National Lottery Hit By Peak-Time DDoS Attack." [Online]. Available: https://www.silicon.co.uk/security/uk-national-lottery-ddos-222601?inf_by=5b7ee807671db80d6d8b4982
[4] "Github–February 28th DDoS Incident Report." [Online]. Available: https://githubengineering.com/ddos-incident-report
[5] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman *et al.*, "Understanding the mirai botnet," in *USENIX Security Symposium*, 2017, pp. 1092–1110.
[6] M. H. Bhuyan, H. J. Kashyap, D. K. Bhattacharyya *et al.*, "Detecting distributed denial of service attacks: methods, tools and future directions," *The Computer Journal*, vol. 57, no. 4, pp. 537–556, 2013.
[7] O. Osanaiye *et al.*, "Distributed denial of service resilience in cloud: review and conceptual cloud DDoS mitigation framework," *Journal of Network and Computer Applications*, vol. 67, pp. 147–165, 2016.
[8] G. Somani, M. S. Gaur, D. Sanghi, M. Conti, and R. Buyya, "DDoS attacks in cloud computing: Issues, taxonomy, and future directions," *Computer Communications*, vol. 107, pp. 30–48, 2017.
[9] J. Mirkovic and P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 39–53, 2004.
[10] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against Distributed Denial of Service flooding attacks," *IEEE communications surveys & tutorials*, vol. 15, no. 4, pp. 2046–2069, 2013.
[11] Q. Yan, F. R. Yu, Q. Gong, and J. Li, "Software-defined networking (SDN) and distributed denial of service attacks in cloud computing environments: A survey, some research issues, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 602–622, 2016.
[12] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández *et al.*, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *computers & security*, vol. 28, no. 1-2, pp. 18–28, 2009.
[13] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: methods, systems and tools," *IEEE communications surveys & tutorials*, vol. 16, no. 1, pp. 303–336, 2014.
[14] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.
[15] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, 2016.
[16] J. Mirkovic, G. Prier, and P. Reiher, "Attacking ddos at the source," in *Network Protocols, 2002. Proceedings. 10th IEEE International Conference on*. IEEE, 2002, pp. 312–321.
[17] "The CAIDA 'DDoS Attack 2007' Dataset." [Online]. Available: https://www.caida.org/data/passive/ddos-20070804_dataset.xml
[18] "1999 DARPA Intrusion Detection Evaluation Data Set." [Online]. Available: https://www.ll.mit.edu/ideval/data/1999data.html
[19] "Tcpreplay," http://tcpreplay.synfin.net/.
[20] M. Galar, A. Fernandez, E. Barrenechea *et al.*, "A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp. 463–484, 2012.
[21] B. Krawczyk, "Learning from imbalanced data: open challenges and future directions," *Progress in Artificial Intelligence*, vol. 5, no. 4, pp. 221–232, 2016.