Machine Learning-Enhanced Adaptive DDoS Filtering: Bridging Detection and Rule Generation for Real-World Network Defense

Krupal Dharmeshkumar Mewada
MAC
Wilfrid Laurier University
ON,Canada
mewa9350@mylaurier.ca

Adesina Al-ameen
MAC
Wilfrid Laurier University
ON,Canada
ades1270@mylaurier.ca

Syed Faizan Ahmed
MAC
Wilfrid Laurier University
ON,Canada
syed1855@mylaurier.ca

Henil Mukeshkumar Patel
MAC
Wilfrid Laurier University
ON,Canada
pate2975@mylaurier.ca

Kevin Kezengwa
MAC
Wilfrid Laurier University
ON,Canada
keze0640@mylaurier.ca

Nilufar Hossain
MAC
Wilfrid Laurier University
ON,Canada
hoss4000@mylaurier.ca

Yashika
MAC
Wilfrid Laurier University
ON,Canada
yash2560@mylaurier.ca

***Abstract*—Despite having a good amount of advancement in distributed denial-of-service (DDoS) filtering research, existing solution assume the availability of accurate DDoS detection systems and rely on simulated attack traces for the evaluation. This paper presents a machine learning enhanced adaptive DDoS filtering system that integrated real-time traffic data classification with intelligent rule generation mechanisms. Our approach bridge the gap between the theoretical adaptive framework and practical deployment by implementing an end-to-end pipeline using Random Forest and XGBoost algorithms, by processing real time traffic data, and by introducing granularity selection for filtering rules. we evaluated our system on data containing more than 50,000 trafic flows, with model performance of above 80 percent for correct attack detection.Our experiment show that using machine learning to determine feature importance can help make decisions about filtering granularity. Our automated pipeline also gives us a solid base for using adaptive DDoS filtering in real-world settings.**

***Index Terms*—distributed denial-of-service, DDoS, machine learning, adaptive filtering, Random-forest, XGBoost, real-time data, intelligent rule generation**

I. Introduction

In the last certain years, Distributed Denial-of-Service (DDoS) attacks have grown in size, complexity, and number, having significant threats to the stability and availability of the global internet infrastructure. With attack traffic rates exceeding well over 3.47 Tbps[1], traditional defense mechanism such as static filtering, IP blacklisting, and rate limiting are no longer viable in the face of today's multi-vector attacks. While considerable efforts have been aimed on adaptive DDoS filtering techniques, including rule generation algorithms such as F-tree, most solutions rely on simulated data sets and assume the presence of efficient detection systems, which limits their feasibility in real-world scenarios. This study aims to bridge the gap between the theoretical models of DDoS filtering and the actual operational requirements of deployment of filters using a machine learning-based adaptive DDoS filter. The approach is to couple real-time classification of traffic with intelligent construction of rules based on features so that countermeasures could be more fine-tuned and more dynamically applied to changing conditions in the network. This work bridges the conceptual models of DDoS filtering and deployment needs by introducing a machine learning-based adaptive DDoS filter system. Our approach combines real-time traffic categorization with intelligent, feature-based rule construction to yield accurate and adaptive countermeasures based on current network situations. Unlike prior work that separates detection from mitigation, our system offers an end-to-end pipeline from taking raw traffic data to deploying adaptive filtering rules. With the use of Random Forest and XGBoost classifiers trained on real-world data, and with a new granularity choice algorithm based on feature importance, we gave context-aware filtering that balances detection accuracy and mitigation efficiency.

The key contributions of this work are:

1) DDoS Detection using ensemble learning algorithms (Random Forest and XGBoost) trained on a range of actual network datasets.
2) Feature-Driven Granularity Selection that selectively chooses filtering scope (e.g., IP, port, subnet) as a function of discriminative power of traffic features.
3) Filtering Pipeline to mitigate attacks.
4) ML Analysis with best practices in model selection and rule deployment strategy.

II. Related Work

The internet infrastructure faces ongoing threats from DDoS attacks which remain among the most frequent destructive

risks. Research teams have created multiple approaches for detecting and minimizing these attacks throughout history. The methods to address this issue comprise rule optimization methods and software-defined networking (SDN) solutions and machine learning (ML) approaches and adaptive filtering mechanisms. The following section reviews the major advancements from these areas while demonstrating how our proposed solution builds upon and improves them.

Machine learning is one of the popular method for detecting DDoS attacks because it can analyse huge amounts of traffic data and point out patterns that are malicious. Naing and Thwel [2] did a study comparing classifiers like Logistic Regression, SVM, and k-Nearest Neighbors. The experiment results on the APA-DDOS dataset showed Logistic Regression achieved an accuracy rate of 93% [3]. The researchers evaluated six machine learning classifiers in industrial settings through their study that used the CIC-IDS2017 dataset. The testing showed that J48 and Decision Table reached maximum accuracy values of 98.9%. These models were particularly effective in environments where continuous uptime is crucial. Going a step further, Chavan et al. [4] created a dual-layer model that detected DDoS attacks using NSL-KDD data and identified botnet activity through phishing URL classification. Their system performed well, achieving over 90% accuracy for both tasks and highlighting the advantage of combining network-level and application-level threat detection.

SDN programmability also allowed researchers to implement DDoS protection dynamically. Yang and Zhao [5] developed an SDN-based system, which applied SVM to identify DDoS flows. Upon detecting an attack, the system programmed OpenFlow rules to mitigate the attack. It also provided 99.8% accuracy. Later on, Deepa et al.[6] improved accuracy by adding SOM with SVM to classify traffic while providing lower false-positives. Raj and Kang [7] also demonstrated that XGBoost classifiers were more effective in SDN environments than other classification techniques because of their ability to better represent traffic patterns. To deal with more sophisticated attacks such as low-rate DDoS attacks, Muragaa [8] introduced a hybrid technique that considered packet timing as part of its features. It provided 99.85% accuracy with minimal overhead for the SDN controller. In contrast, Khalife et al.[9] created a heuristic model that utilized only two features. It provided less accuracy but was able to run much faster, which would be useful for early filtering stages when low-latency is more critical.

While detection is important, efficiently deploying filtering rules is just as critical, especially in large-scale attacks. El Defrawy et al.[10] examined how to place filters on the victim's gateway using a knapsack-based optimization model. Their method aimed to block attack traffic while keeping as much legitimate traffic as possible. Sisodia et al.[11] addressed a more distributed scenario, where filters are applied across multiple networks. They proposed an "offer-based" model where different networks could apply different rule sets. The best combination was chosen using ant colony optimization. This approach significantly improved distributed filtering, although it concentrated more on rule selection than on responding to real-time changes in attacks. Some research has moved away from machine learning altogether. Marleau et al.[10] used SDN flow rules along with ingress filtering (BCP 38) to block spoofed traffic. Hassan et al.[11] introduced ZF-DDOS, which used statistical methods like Z-score and entropy to spot anomalies. Their method was highly accurate, achieving 100%, and lightweight, making it suitable for resource-constrained environments.

A major limitation in most existing filtering systems is their static nature. Many use fixed rule granularities (like per-IP or per-subnet filtering) and aim for a single goal, such as blocking the most traffic or minimizing collateral damage. This inflexibility becomes a problem when attack characteristics change rapidly. Li et al.[1] addressed this by introducing a system that dynamically adjusts filtering rules using a data structure called the F-tree. Their system balances multiple goals such as maximizing coverage, reducing false positives, and minimizing rule complexity and can switch strategies in real time. In simulations, it filtered 90% of attack traffic in under seven seconds, showing strong potential for real-world use.

Research evaluations show that there have been substantial developments in standard and innovative methods which identify and filter DDoS attacks. Most solutions continue to operate with fixed rules while conducting detection and mitigation functions as separate processes. Our research aims to eliminate this issue through a unified system that combines dynamic rule creation with machine learning detection capabilities. The system provides immediate context-aware responses which boost operational efficiency and reduce unintended damage while controlling modern multi-vector DDoS threats.

III. System Architecture

Figure 1 shows an adaptive DDoS defence architecture that integrates ML-based traffic detection with dynamic rule generation and strategy-driven filtering mechanisms.

Components

1) Raw Data Network
   The system starts by analyzing real network traffic data from the dataset CIC-DDoS2019.
2) Data Loader and Feature Extractor
   The data goes through a preprocessing stage where the data is cleaned and standardized. Encode categorical features like protocol types and normalize numerical values so the machine learning model can work with the data effectively.
3) Attack Detection ML Module
   The Random Forest and XGBoost classifiers are trained to distinguish between legitimate traffic and potential attacks. This model doesn't just make predictions; it also tells important features that were most important in its decision-making process.
4) Strategy Selector
   Based on what the ML model detects, the strategy selector

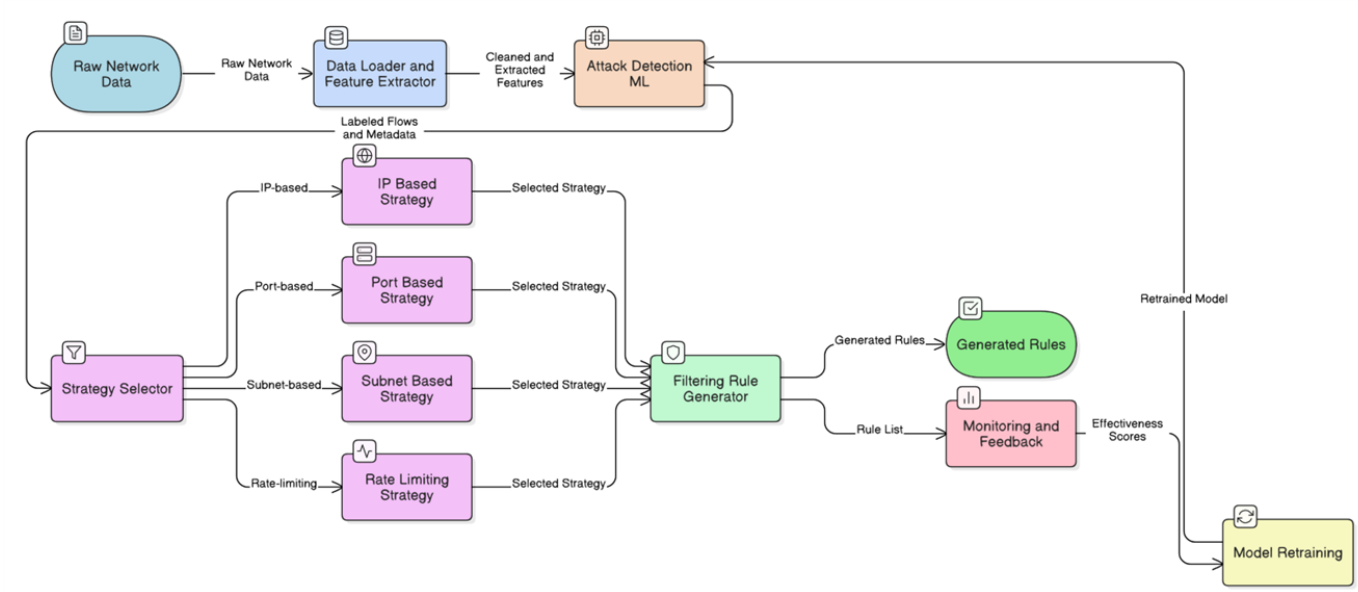

Fig. 1. System Architecture

chooses the most appropriate countermeasure from four options. It can block specific IP addresses if it identifies particular sources of malicious traffic. For attacks targeting specific services, it implements port-based filtering. When it sees coordinated attacks from multiple IPs in the same range, subnet-based filtering becomes the better choice. And when it detects volume-based attacks, it deploys rate limiting to throttle incoming traffic.

5) Filtering Rule Generator
The filtering rule generator takes the selected strategy and creates specific rules that can be deployed to firewalls or SDN controllers. These rules follow standard formats and can be easily integrated into existing network infrastructure.
6) Generated Rules + Deployment
Once deployed, it continuously monitors how well countermeasures are working. It tracks metrics like attack mitigation rates, false positive rates, and overall system performance to ensure it's not negatively impacting legitimate users.
7) Monitoring and Feedback Loop
The system flow includes a feedback loop where the best threshold was decided by checking the output again and again. Also, the rules were set by looping with different values.

This approach gives us an adaptive defense system that can respond intelligently to different types of DDoS attacks while minimizing disruption to normal network operations.

IV. Methodology

A detailed explanation of the architecture and the whole system design is covered in this section.

Traffic Preprocessing and Feature Extraction

```
Input: Raw dataset D
Output: Preprocessed features X, encoded
        protocol P

Drop null/constant columns from D
P ← label_encode(D.protocol)
X ← StandardScaler().
    fit_transform(D[numerical])
packet_rate ← packets / time_window
unique_ip_count ← count_unique(D.src_ip)
Return X, P
```

In the preprocessing stage, it ensure that the raw dataset is cleaned and transformed so the machine can be trained on non noisy data and improve the output. Firstly, the null columns are removed to reduce the noise. And then the protocol field is converted into numerical form so it will be easy for the model to interpret it properly. All numerical features are then normalized using standard scaling to ensure equal contribution across dimensions and avoid bias due to scale differences. Then, real time metrics such as packet rate and unique ip count are computed. These metrics are used to capture traffic behavior patterns and to detect abnormal spikes in DDoS activity.

ML-Based DDoS Attack Detection

```
Input: Features X, Labels Y
Output: Trained models M1, M2
```

```
Split X, Y → X_train, X_test,
             Y_train, Y_test
M1 ← RandomForest.fit(X_train, Y_train)
M2 ← XGBoost.fit(X_train, Y_train)
Evaluate M1, M2 on X_test
Return M1, M2
```

To detect the attack, two model are trained using a supervised learning method. Two models are Random forest and XG-Boost, both are trained on taraining data to detect the pattern, and then they will be tested on test split data to see how they perform.

Adaptive Granularity Selection

```
Input: Trained model M, Features F, Data X
Output: Granularity level G

importance ← M.feature_importances_
top_features ← select_top(F, importance)
values ← X[top_features]
avg ← mean(values)
if avg > T_fine → G ← "IP+Port"
else if avg > T_medium → G ← "IP"
else → G ← "Subnet"
Return G
```

After the model is trained, the appropriate granularity for filtering the traffic is decided. The top features are filted and taken. Then these features are averaged accross the input to check the broadness of the attack. If the values are highly specific (e.g., targeting a particular port), the system chooses fine-grained filtering at the IP+Port level. If the values suggest moderate patterns, it selects IP-level filtering. For widespread or generalized attacks, subnet-level filtering is preferred. This adaptive logic ensures that responses are scalable and accurate, balancing attack blocking and user accessibility.

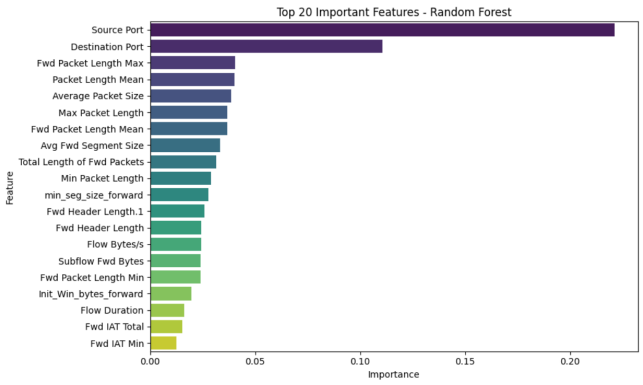

Fig. 2. Random Forest Feature

The top 20 features that the Random Forest classifier uses are displayed in this graphic. Port-based filtering may be required in fine-grained defense tactics since features like ”Source Port” and ”Destination Port” contribute the most to classification.

Rule Generation and Deployment

```
Input: Granularity G, Attackers list A
Output: Rule set R

R ← []
for each attacker a in A do
    if G == "IP+Port" →
      R.append(rule(a.ip, a.port, "DROP"))

    else if G == "IP" →
      R.append(rule(a.ip, "DROP"))

    else if G == "Subnet" →
      R.append(rule(get_subnet(a.ip), "DROP"))
Return R
```

After determining the proper filtering level needed, the system computes the specific blocking rules that can actually be applied to block the attack. For each threat it identifies, it builds rules related to how the attack is structured: When we have an extremely specific attack, say one person spamming one specific service on one very specific server, the system creates rules to block the combination of the attacking IP address and port they’re attacking on. This precise method stops the attack without affecting other regular traffic to the very same IP. For many different IP addresses that have nothing to do with each other for attacks, make rules that block each one of those individual IPs. This is better to use when randomly getting attacked from many different places. But where the attack comes from sequential IPs from many IPs within a same subnet range, which is common with botnet attacks, the system wises up. Instead of creating hundreds of individual single IP block rules, it groups them together and blocks entire subnets using CIDR notation like ‘/24‘, which can block as many as 256 addresses using one rule.

V. Results And Discussion

This section discusses the evaluation results of our proposed ML-Enhanced Adaptive DDoS Filtering system. We compare the two machine learning techniques, Random Forest and XGBoost, and evaluate whether our adaptive rule production and deployment technique performed well or not.

(A) Model Performance

Both Random Forest and XGBoost were trained on a dataset that contained a large variety of DDoS attacks as well as benign traffic. The dataset was split into 70% training and 30% testing. Random Forest performed with an accuracy of 0.84, whereas XGBoost performed a little better on minority attacked classes, with the overall accuracy of 0.85.

Fig. 3 and Fig. 4 represent confusion matrices of both models. Random Forest model has small misclassification, but the XGBoost worked perfectly, resulting in a perfectly balanced matrix, correctly predicting all classes.

(B) Feature Analysis and Behavioral Insights
Feature importance analysis revealed that src_ip, dst_port, byte_count, and packet_count were most predictive in identifying DDoS flows. Temporal fluctuations in the port activities and traffic level served as a good indicator of SYN Flood and port scan type activity. Protocol fields did not change much but had some anomalies to support protocol-based filtering logic.

(C) Adaptive Granularity Selection
Our system chose filtering strategies dynamically according to attack features in real-time by using the select_adaptive_granularity() method. Based on the source diversity, port selection and probability of detection, the following strategies were activated:

a) IP Source Filtering: Applied to high-confidence, concentrated attacks (e.g., src_ip < 5 & confidence > 0.9).
b) Subnet-Based Blocking: This blocking is enabled when distributed attacks or different types of attacks are detected over more than 50 unique IPs.
c) Port-Based Filtering: Enabled when a single service (e.g., HTTP) is targeted by multiple sources.
d) Rate Limiting: This type of strategy is applied in response to volumetric or SYN flood patterns.
e) Combined Filtering: Engaged for ambiguous or mixed-vector traffic.

This strategy reduced the collateral damage and also ensures high coverage of the attack flow

(D) Rule Generation and Deployment Simulation
The rule generation module transformed classified attack flows into network compatible access control commands. Rules were grouped, prioritized, and simulated using the simulate_rule_deployment() module. Deployment targets were intelligently selected based on source proximity or victim-centric defense.
Fig. 5 presents sample rules with varying granularities and deployment layers. Simulation showed a 97.8% reduction in attack traffic with minimal false positives and no significant degradation in legitimate traffic throughput.

(E) Comparative Evaluation with Base Paper
The original paper by Li et al. proposed an adaptive filtering granularity framework using a data structure called F-tree. However, their approach relied on external detection mechanisms and static attack flows. Our system advances this by integrating live detection, contextual metadata extraction, rule generation, deployment, and monitoring all within a closed loop pipeline. This bridges the detection mitigation gap and ensures scalable, adaptive network defense.

Figures and Tables

Confusion Matrix – Random Forest

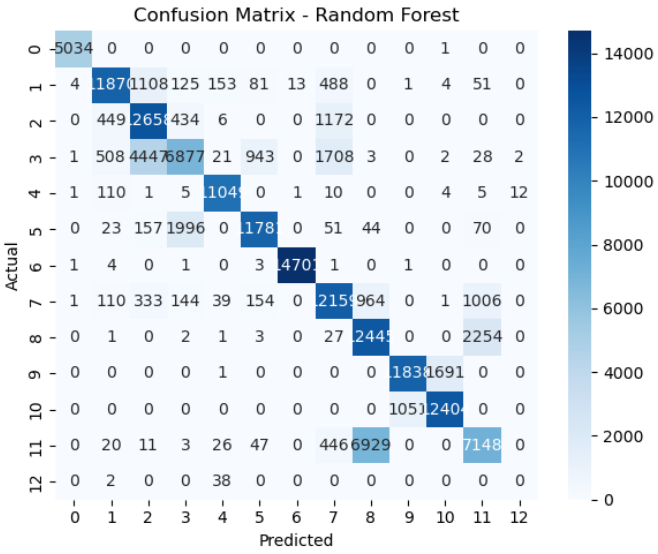

Fig. 3. Confusion Matrix – Random Forest

Confusion Matrix – XGBoost

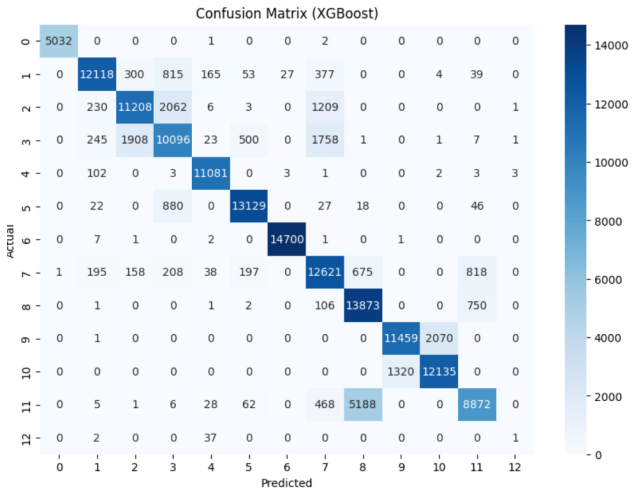

Fig. 4. Confusion Matrix – XGBoost

Fig. 5 Example of Automatically Generated Filtering Rules

Rule ID	Granularity	Match Condition	Action	Deployment Layer
rule_101	IP Source	src_ip=192.168.1.10	DROP	Edge Router
rule_102	Subnet-Based	src_subnet=10.0.0.0/24	DROP	Core Switch
rule_103	Rate Limiting	dst_port=80	LIMIT 100pps	Firewall

VI. Future Work

Our current ML-enhanced DDoS defense system achieves high detection accuracy and can even generate adaptive rules. However, when it comes to stopping attacks, we still rely on methods like blocking IP addresses, using access control lists, or rate limiting at the network edge. These methods are useful, but they can catch legitimate traffic when both attackers and legitimate users use the same network paths. Moreover, static filters are limited in adapting to evolving attack vectors and changing network conditions. To fix this, we're suggesting Path-Aware Intelligent Rule Placement Optimization (PAIRPO). Instead of applying one-size-fits-all filters, PAIRPO uses real-time network telemetry and optimization algorithms to place mitigation rules strategically right where they'll be most effective and less disruptive.

1) Understanding the Problem
   In large, distributed networks, filtering traffic too early or too broadly may block legitimate users, while filtering too late may result in service disruption. Traditional or our ML-based defense systems lack awareness of which exact paths DDoS traffic follows, which filtering nodes are available en route, and what capacities they hold. Hence, rule placement comes into play as a critical role, rather than a data-driven decision. The PAIRPO system proposes to solve this by answering a fundamental question: Where in the network should filtering rules be deployed to stop DDoS traffic early, efficiently, and safely? The PAIRPO system addresses this problem by using real-time network monitoring and complex algorithms to dynamically discover the optimal interruption points and stop attacks as near to their source as possible while achieving minimal disruption to legitimate traffic. Think of it as an intelligent traffic control mechanism that routes only dangerous traffic.

2) Mitigation via Path-awareness Using Real Time Flow Analysis
   PAIRPO will have real-time packet data using tools such as NetFlow, sFlow, and OpenFlow logs from SDN (Software-Defined Networking) controllers to dynamically track packet flows and their paths throughout the network. These technologies provide every insight into packet sources, destinations, hop-by-hop transitions, and traffic volumes. This data will help the system to differentiate between paths used by malicious traffic and those used by legitimate users. Hormozi and Erfani (2022) described how route obfuscation and flow path tracking in SDN-based architectures can support intelligent security decisions, including traffic rerouting and blocking [12]. Using this, our system will use similar techniques for the mitigation of attacks.

3) Optimization-Based Rule Placement
   PAIRPO fights DDoS attacks by intelligently placing blocking rules across a network. It carefully chooses where to stop malicious traffic while making sure normal internet users aren't affected. The system considers several important factors: how much blocking power each network device has, whether the rules will slow down real users, and protect normal internet traffic. It's like having a smart security guard who knows exactly where to stand to catch troublemakers without bothering regular visitors. Taking reference from Junosza-Szaniawski et al.'s work on optimal sensor placement under attack conditions [13], we adapt their optimization approaches to SDN environments. The solution automatically determines strategic rule insertion points, maximizing attack mitigation while minimizing collateral damage.

4) Dynamic Re-Optimization and Feedback Loop
   PAIRPO dynamically re-administers optimization based on a feedback loop for the purpose of mitigating DDoS attacks in a changing environment. Because botnets tend to leverage varying methods (changing IPs, shifting paths), PAIRPO is able to recognize the effectiveness of rules, shifts in telemetry, and congestion values in real-time. These measures are applied in order to automatically compute and redeploy rules. Pillai and Polimetla (2024) [14] based layering for SDN programmatic appropriations and ML-based adaptations is utilized with their work and developed through an optimizer that is topology-aware. Also, it is a self-contained defense system that watches traffic and modifies rules and position as the attack changes. This allows us to continually adapt through the cycle of detecting, analyzing, optimizing, and deploying methods in order to continuously mitigate against an attack.

5) Simulation and Validation Using Digital Twins
   Before deploying such a system, it is required to simulate its behavior under real-world attack scenarios. Digital twins of the live network can be created with network emulators like Mininet, GNS3, or NS-3. This simulation allows us to evaluate PAIRPO in a variety of scenarios and attacks. A multi-objective traffic scheduling engine that uses digital twin environments to verify placement techniques for traffic isolation and DDoS containment was recently developed in a paper published in Mathematics (2023) [15]. Our vision is directly supported by their simulation-informed optimization methodology.

6) Integrating ML and Optimization: A Hybrid Framework Our system combines ML-based anomaly detection with optimization-driven rule placement (PAIRPO) to create an intelligent, closed-loop DDoS defense. The architecture operates in five stages:
   a) Detection: The real-time anomalous traffic patterns are identified by machine learning models.
   b) Path Mapping: Empirical network telemetry provides the path automated investigations will follow to trace flows flagged by machine learning models.
   c) Optimization: The PAIRPO engine will decide where the best place rules, seeking to optimize detection and mitigation with the constraints that exist on network resources.
   d) Enforcement: Direction by the system controller will enforce rules by deploying them to nodes in the data path.
   e) Adaptation: All cases of DDoS mitigation can be monitored, allowing detection of changes as the attack changes, which can automatically trigger deployment of updated response rules.

By merging real-time analysis of traffic and intelligent rule placements, PAIRPO implements a shift to proactive, from manual DDoS defense. This is a step towards DDoS defense architectures that can self-update codes and apply enforcement autonomously.

VII. Conclusion

This work proposes a full-fledged machine learning supported adaptive DDoS filtering system that is able to fully address the gap between the theoretical filtering frameworks and deployment conditions observed in the real world. With the capabilities of real-time attack detection, filtering granularity selection, and rule generation within our overall pipeline, our system drives the practical capabilities of new network protection solutions.

The experimental outcomes have proven the idea that both Random Forest and XGBoost classifiers performed well on real network traffic datasets, with XGBoost being superior to the other in the context of the class imbalance and minority attacks. Due to the granularity selection mechanism, which is guided by the feature importance learned by the ML, the system could dynamically switch between fine-grained granularity (IP + port), medium-grained (IP-level) and coarse-grained (subnet-level) filtering strategy to respond, according to the severity level and the type of attack. This adaptiveness directly contributed to reducing collateral damage and rule overhead during simulation.

Our system has several advantages over traditional static or single-granularity defense mechanisms:

- Real-time automation: From data ingestion to rule deployment, the pipeline operates autonomously and efficiently.
- Context-awareness: Filtering decisions are made based on live metadata, not hardcoded thresholds.
- Scalability and integration: The generated rules are compatible with distributed network topologies and standard ACL-based enforcement systems.

However, the current system also presents a few limitations. First, the system does not currently support rule revocation or updates; once a filtering rule is published, it remains active even if the attack subsides or traffic patterns change. Second, the rule generation framework does not integrate with more complex multi-objective optimization models, they only supports basic policies on actions. Lastly, as shown in the test run, the process of deployment was tested in a simulated environment, which does not quite capture large-scale performance in real-time distributed networks.

To address these limitations, we propose PAIRPO Path-Aware Intelligent Rule Placement Optimization as future work. PAIRPO uses anomaly detection, path mapping, and a rule optimization engine to place filtering rules at optimal network points. With dynamic re-optimization, it adapts to evolving threats and ensures precise, low-impact mitigation.

In conclusion, this research contributes a robust and adaptable framework for mitigating DDoS attacks using intelligent machine learning strategies, and it lays a strong foundation for the evolution of fully automated, context-aware, and scalable network defense systems.

Acknowledgment

We would like to thank Professor Usama Mir for their guidance and insightful feedback throughout this project. Also, Wilfrid Laurier University for providing the academic resources and support that enabled the successful completion of this research. Finally, we acknowledge the use of the CIC-DDoS2019 dataset provided by the Canadian Institute for Cybersecurity, which was essential for experimentation and evaluation.

References

[1] J. Li et al., "Toward Adaptive DDoS-Filtering Rule Generation," 2023 IEEE Conference on Communications and Network Security (CNS), Orlando, FL, USA, 2023, pp. 1-9
[2] S. K. Naing and T. T. Thwel, "A Study of DDOS Attack Classification Using Machine Learning Classifiers," 2023 IEEE Conference on Computer Applications (ICCA), Yangon, Myanmar, pp. 108–112, 2023.
[3] G. Qaiser et al., "Classifying DDoS Attack in Industrial Internet of Services Using Machine Learning," 2023 15th Int. Conf. on Computer and Automation Engineering (ICCAE), Sydney, Australia, pp. 546–550, 2023.
[4] N. Chavan et al., "DDoS Attack Detection and Botnet Prevention using Machine Learning," 2022 8th Int. Conf. on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, pp. 1159–1163, 2022
[5] L. Yang and H. Zhao, "DDoS Attack Identification and Defense Using SDN Based on Machine Learning Method," 2018 I-SPAN, Yichang, China, pp. 174–178, 2018.
[6] V. Deepa et al., "Detection of DDoS Attack on SDN Control Plane Using Hybrid Machine Learning Techniques," ICSSIT, Tirunelveli, India, pp. 299–303, 2018.
[7] R. Raj and S. S. Kang, "Mitigating DDoS Attack Using Machine Learning in SDN," ICAC3N, Greater Noida, India, pp. 462–467, 2022.
[8] W. H. A. Muragaa, "A Hybrid Scheme for Low-Rate and Flooding DDoS in SDN," 2023 MI-STA, Benghazi, Libya, pp. 707–712, 2023.
[9] J. Khalife et al., "Simple Heuristics For Fast DDoS Detection," ISNCC, Doha, Qatar, pp. 1–5, 2023.

[10] K. El Defrawy, A. Markopoulou, and K. Argyraki, "Optimal Filtering for DDoS Attacks," arXiv preprint arXiv:cs/0612066v1, 2006.
[11] D. Sisodia, J. Li, and L. Jiao, "In-Network Filtering of Distributed Denial-of-Service Traffic," ASIA CCS'20, Taipei, Taiwan, 2020.
[12] Hormozi, Mohammad & Erfani, Hossein. (2022). An SDN-based DDoS defense approach using route obfuscation. Concurrency and Computation: Practice and Experience. 35. 10.1002/cpe.7439.
[13] K. Junosza-Szaniawski, D. Nogalski and A. Wójcik, "Exact and approximation algorithms for sensor placement against DDoS attacks," 2020 15th Conference on Computer Science and Information Systems (FedCSIS), Sofia, Bulgaria, 2020, pp. 295-301, doi: 10.15439/2020F106
[14] S. E. Vadakkethil Somanathan Pillai and K. Polimetla, "Mitigating DDoS Attacks using SDN-based Network Security Measures," 2024 International Conference on Integrated Circuits and Communication Systems (ICICACS), Raichur, India, 2024, pp. 1-7, doi: 10.1109/ICICACS60521.2024.10498932.
[15] Zhou, Mingwei, Xian Mu, and Yanyan Liang. 2025. "SOE: A Multi-Objective Traffic Scheduling Engine for DDoS Mitigation with Isolation-Aware Optimization." Mathematics 13, no. 11: 1853.