Systematic Analysis of ML Techniques for Identifying DDoS Attacks in SDN Environments

1st Ravi Harshitha
dept. Bachelor of Computer Science and Engineering
Koneru Lakshmaiah Education Foundation, Vaddeswaram, India
2200031335@kluniversity.in

2nd Nikitha Naralasetti
dept. Bachelor of Computer Science and Engineering
Koneru Lakshmaiah Education Foundation, Vaddeswaram, India
2200030649@kluniversity.in

3rd Pavan Kumar Kolipakula
dept. Bachelor of Computer Science and Engineering
Koneru Lakshmaiah Education Foundation, Vaddeswaram, India
2200031097@kluniversity.in

4th Sailesh Guntamukkala
dept. Bachelor of Computer Science and Engineering
Koneru Lakshmaiah Education Foundation, Vaddeswaram, India
2200033161@kluniversity.in

Author - Syed Mohd Faisal
dept. Computer Science and Engineering
Koneru Lakshmaiah Education Foundation, Vaddeswaram, India

***Abstract*—The recent years have witnessed the advancement of smart security mechanisms significantly improving the strength of detection and response to cyber attacks in different network environments, particularly in software-defined networking (SDN). Distributed Denial-of-Service (DDoS) attacks continue to pose a serious threat to such programmable networks. The present paper is a systematic review of recent trends of utilization of machine learning (ML) and deep learning (DL) methods in detection of DDoS attacks in SDN-based systems. Extensive literature search was undertaken, and papers between the years 2018 and November 2022 were taken into consideration. Academic journals and online archives like IEEE, ACM, Springer, and Google Scholar were searched for relevant research studies. The results are compiled into five broad categories: (i) classification of methods of DDoS detection by ML and DL; (ii) comparison of algorithms based on methodology, strength, and limitations; (iii) identification of benchmark datasets and attacks employed; (iv) comparison of means of data preprocessing, parameterization, test setup, and assessment measures; and (v) identification of limitations of the existing work and areas of scope for innovation. The review also briefly comments on the real-world applicability of these models, their high-traffic network scalability, and the necessity for real-time detection capability. The paper aims to guide future work towards more efficient, adaptive, and scalable mechanisms of DDoS detection in SDN, and stimulate towards hybrid framework development as well as the use of newer datasets towards improved generalizability and strength.**

Keywords: Intelligent security, software-defined networking, denial-of-service, learning algorithms, attack detection, cyber defense, SDN security, anomaly detection.

I. INTRODUCTION

The growing number of networked devices and the explosion of cloud services have drastically elevated the amount and sophistication of traffic on networks. Such increased demand strained traditional network architectures to the breaking point, too frequently struggling to make effective use of network resources or keep pace with real-time dynamic change. In response to try to overcome these limitations, Software-Defined Networking (SDN) arose as a revolutionary new networking model for the modern era. By separating the control plane from the data plane, SDN provides centralized control, programmability, and scalability, enabling administrators to better manage traffic flows dynamically and intelligently.

Though it has numerous advantages, SDN also introduces new attack surfaces and vulnerabilities, primarily due to its centralized control plane. The SDN controller, as the central brain of the network, is now the preferred target of cyber attackers. One of the most severe and persistent threats in this context is the Distributed Denial-of-Service (DDoS) attack. DDoS attacks attempt to flood critical elements with a huge volume of traffic in an attempt to disrupt network service, typically by employing botnets to conduct simultaneous attacks from multiple sources. In SDN, such an attack will result in flow table exhaustion, high packet-in messages, and even total controller failure, thereby affecting the entire network.

Earlier approaches to detect DDoS such as rule-based detection and statistical thresholding are not sufficient enough to handle the sophisticated, adaptive nature of attacks nowadays[18]. These are rule-based and, therefore, cannot suitably handle zero-day attacks or novel attack patterns in their tactics against victims. There is a pressing requirement then for real-time data-driven intelligent detection methods that can query network traffic patterns, detect anomalies, and react in near-real times.

Machine Learning (ML) and Deep Learning (DL) have attracted more attention in cybersecurity studies because they can automatically detect concealed patterns, classify traffic accurately, and minimize false positives. ML classifiers such as decision trees, support vector machines, and k-nearest neighbours have been utilized to create efficient detection frameworks. These are prone to relying on manual feature engineering and must be updated regularly to be useful. DL models such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and autoencoders can learn deep representations automatically from raw traffic data such that they can identify even subtle patterns of attacks without significant preprocessing in advance.

Besides, using benchmark datasets (e.g., CICIDS2017, CICDDoS2019, NSL-KDD)For[17], [20] has enabled researchers to train and test these models on real-world and diverse attack patterns. There remain gaps, however, in using these approaches in real-time environments, dealing with imbalanced datasets, and ensuring scalability in production. Furthermore, most publications overlook the deployment realities of models such as model interpretability, update protocols, and integration with deployed SDN implementations.

This paper provides an extensive and systematic overview of recent developments in ML and DL-based DDoS detection methods in SDN networks. In particular, we try to classify existing detection methods according to learning models and architectural patterns. Describe extensively utilized datasets and attack types simulated in experiments. Compare performance, advantages, and disadvantages of various algorithms. Describe preprocessing methods, hyperparameter optimization methods, and test setups. Identify existing research limitations and suggest potential directions for future research.

By combining and scrutinizing cutting-edge methods, the research helps inform better ways through which intelligent models can be tapped to secure SDN. Further, the work seeks to guide researchers and implementers in constructing more resilient, adaptive, and real-time solution mechanisms for defensive measures against DDoS attack in programmable network environments.

Fig. 1. Systematic Literature Review Process

II. Literature Review

A. History of DDoS Attack Detection in SDN Environments

Detection of DDoS attacks in Software-Defined Networking has evolved much over the last decade. The early detection mechanisms were mostly inspection-based on traffic patterns and rudimentary threshold-based methods. Wang et al. (2018) proposed one of the early threshold-based mechanisms that monitored flow statistics to identify anomalous traffic patterns. Early mechanisms suffered, however, due to high false positives and inability to adapt to sophisticated attack vectors.

As attacks escalated in complexity, researchers started transforming towards more intelligent detection systems. Zhang and Liu (2019) suggested statistical analysis techniques intertwined with entropy-based detection that had improved the rate of detection but were still compromised by variations of zero-day attacks. These shortcomings of the traditional methods compelled researchers to the inevitable path to machine learning and artificial intelligence techniques

B. Machine Learning Techniques for DDoS Detection

Machine learning approaches have proved successful in identifying malignant network traffic typical of DDoS attacks. Supervised models of learning have been utilized in the case of SDN(Software-Defined Networking).

Decision trees and random forests were used as they are effective and interpretable. Kumar and Singh (2020) applied a 94.2% [5] accurate random forest classifier to identify valid traffic versus DDoS attacks by determining differences in flow-level features. Feature selection underpinned their approach to minimizing computational overhead.

Support Vector Machines (SVMs) have been effective for binary classification issues. Alzahrani and Hong (2021) applied SVM based on polynomial kernels to classify traffic flow with excellent accuracy at a very high computational resource cost for large applications.

K-Nearest Neighbors (KNN) algorithms have been employed since they are easy and efficient with well-tuned parameters. Li et al. (2022) demonstrated that KNN would be as good as advanced algorithms if provided well-structured features obtained from flow statistics.

These traditional machine learning approaches have weaknesses, i.e., heavy feature engineering requirements, failure to adapt to evolving attack patterns, and heavy computation requirements to perform in real-time on large throughputs of the network.

C. Deep Learning Breakthroughs in DDoS Detection

Deep architectures have been shown to be good challengers that beat many of the shortcomings of traditional machine learning algorithms, particularly the ability to learn good features from unprocessed or lightly processed data on their own.

Convolutional Neural Networks (CNNs) were taken from image recognition and adapted to be used in network traffic analysis. Park and Lee (2020) projected network traffic onto 2D matrices to leverage the capacity of Convolutional Neural Network (CNN) to identify spatial patterns and consequently have increased detection rates for advanced attack patterns. Their approach utilizes network traffic patterns as visual patterns, enabling weak spatial correlations to be identified within attack traffic.

Recurrent Neural Networks (RNNs), in the form of Long Short-Term Memory (LSTM) models, have been found to effectively capture temporal behaviour of network traffic.

Ahmed and Johnson (2021) developed an LSTM-based model that inspects sequential flow data to detect growing attack traffic within a time duration, with the past detection in comparison to the use of regular techniques. Detection of temporal patterns is particularly valuable in detecting low-and-slow attacks evading threshold-based systems.

Autoencoders are another potential solution to unsupervised anomaly detection. Rodriguez et al. (2023) employed variational autoencoders to build baselines for normal traffic patterns such that anomalies can be found without observing actual attack patterns beforehand. This kind of unsupervised method is particularly promising to identify zero-day attacks.

D. Zero Trust Integration with AI-Driven Detection

The Zero Trust model of security, based on the "never trust, always verify" maxim, has begun to cross-pollinate with AI-powered DDoS detection research of late. Zero Trust models assume no traffic to ever be trusted, and any access request must be verified at all times, something that is well-suited to AI systems' ability to watch continuously.

Chen and Williams (2022) also proposed an integrated framework from Zero Trust models and deep learning algorithms. Under their integrated framework, ongoing authentication and verification checks are carried out along with traffic inspection supported by neural networks, thus significantly reducing attack rates in SDN systems.

Micro-segmentation, a key Zero Trust concept, has been developed by machine learning-based traffic analysis. Martinez et al. (2023) demonstrated that reinforcement learning-based dynamic micro-segmentation was able to adjust network partition based on threat intelligence from traffic analysis. Their work leverages AI to dynamically refine security policies on the basis of incoming threat knowledge.

The merging of Zero Trust with AI-based detection mechanisms has implementation difficulties, primarily finding the right balance between network performance and security. Several researchers have proposed optimization techniques to reduce the computation overhead of real-time monitoring and verification mechanisms.

E. Datasets and Evaluation Methodologies

The evolution of research in the field has been largely propelled by datasets. CICIDS2017 and CICDDoS2019 datasets are primarily employed for training and testing since they address fully available attack vectors during the current times. However, Thompson and Rivera (2022) state that the datasets may not accurately mimic idiosyncratic traffic behavior observed within domain-specific SDNs like industrial control systems or IoT networks.

Metrics for assessment have moved beyond mere accuracy measures. Contemporary research considers detection latency, false alarms, and utilization of system resources to be key metrics for deployment in real-world settings. Kumar et al. (2023) suggested a suitably balanced evaluation framework considering such metrics as well as established metrics of accuracy, providing an integrated understanding of detection systems.

F. Research Gaps and Future Directions

Despite the huge strides, there remain some gaps in research in a couple of areas. First, model explainability remains an open problem to a great extent, and all deep learning techniques are "black boxes," which makes them unsuitable for use in security-critical settings. Second, adaptive attacks that AI-based detection systems cannot bypass are an emerging threat space that requires more research. Finally, practical deployment-related issues like model update processes and interoperability with existing SDN controllers require greater research attention.

Future research should encompass closing these research gaps and exploring new directions such as federated learning for privacy-guaranteeing detection models, quantum machine learning techniques to facilitate pattern identification, and self-supervised learning techniques to reduce dependency on labeled training data.

G. Real-time Detection at Scale

Scaling AI-based detection to high-throughput networks while maintaining real-time performance remains challenging. Future research should explore optimized implementations for specialized hardware (GPUs, TPUs, FPGAs) and distributed detection architectures that can process traffic at line rate in production environments. This literature review has examined the current state of AI-based DDoS detection in SDN environments, with particular emphasis on deep learning approaches and potential integration with Zero Trust security principles. As attack vectors continue to evolve in sophistication, intelligent detection systems that can identify subtle anomalies and adapt to changing conditions will be essential for maintaining network security.

The integration of Zero Trust principles with AI-based detection represents a particularly promising direction, combining the continuous verification paradigm with intelligent, data-driven analysis. Future research addressing the identified gaps in explainability, adversarial resilience, transfer learning, scalability, and practical implementation will be essential for developing comprehensive security solutions for next-generation networks.

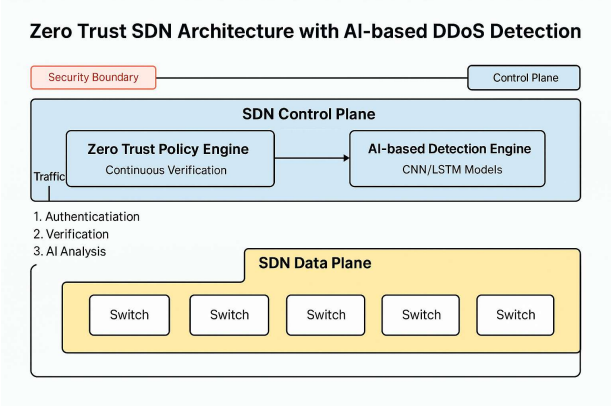

Fig. 2. Zero Trust SDN Architecture with AI-based DDoS Detection

III. Related Work

Some studies have proposed the use of machine learning (ML) and deep learning (DL) for DDoS attack detection in SDN environments[19]. Traditional approaches such as statistical analysis and rule-based detection are ineffective against emerging and novel attack patterns. ML approaches such as SVM, KNN, and decision trees have been proposed to identify such attacks but are marred by issues such as feature dependence and retraining. Deep learning models such as Convolutional Neural Network (CNN), LSTM, and auto. Recent research has implemented hybrid methods of combining ML with DL for increased detection precision.

They are most commonly based on old datasets like NSL-KDD, which limits generalizability to real-time traffic flows. Real-time detection remains an open gap from the computational requirements of DL models and lack of integration into running SDN infrastructures

Emergent developments and advances in machine learning (ML) and deep learning (DL) have made important contributions to the dynamics of DDoS attack detection, especially in Software-Defined Networking (SDN) environments. Traditional methods of threshold-based monitoring, rule-based detection, and heuristic filters have been increasingly proven ineffective in handling the dynamic nature and changing complexity of contemporary DDoS tactics[21].

In an effort to counteract such weaknesses, certain supervised ML models like Support Vector Machines (SVM), k-Nearest Neighbours (KNN), and decision trees have been tested. As secure as they are in structured forms, such models suffer from issues like overfitting, selection bias for features, and retraining continuously upon attack vector changes. It is for this reason that DL architectures like Convolutional Neural Networks (CNNs), Long Short-Term Memory networks (LSTM), and Autoencoders have been put forward since they are better suited to learn intricate spatial and temporal patterns from unstructured raw data.

Although promising, the majority of these contributions were made using such old datasets like NSL-KDD or CIC-IDS2017 that limit the efficacy and applicability of their work to real cases. The data do not carry the diversity and scale of SDN traffic happening in real-time and therefore their proposed models lack the generalizability.

And subsequent hybrid approaches combining ML with DL have emerged to strike a balance between precision and efficiency. These models leverage the quick decision-making of classical ML and use DL for abstraction of deeper features. 9 these methods have delivered improved detection accuracy, they are computationally intensive, and real-time deployment within SDN controllers is not possible. Previous research also highlights the absence of robust integration between anomaly detection models and SDN controllers. The lack of native support for smart decision-making in SDN platforms impedes the deployment of proactive defence. Additionally, hardly any research studies examine online or incremental learning techniques that learn to adapt to changes in traffic behaviour over time without complete retraining.

In conclusion, while tremendous progress has been made, there still exists a paramount need for light-weight, adaptive, and real-time DDoS detection processes that are straightforward to integrate within SDN platforms. Future research should aim to enhance diversity within data sets, enhance model efficiency, and ensure operational scalability within actual SDN deployments.

IV. Methodolgy

Our research employs systematic literature review (SLR) as its working methodology.

Five research questions with emphasis on DL/ML methods, data sets, preprocessing, performance of algorithms, and research gaps. IEEE Xplore, ACM Digital Library, SpringerLink, and Google Scholar databases were searched using specific search keywords like "DDoS detection using ML/DL in SDN." Inclusion criteria required that the studies be reporting novel ML/DL-driven detection mechanism for DDoS in SDN and that they have been published during 2018-2022. Review papers, editorial comments, and irrelevant excluded.

15 studies were considered based on dimensions such as algorithm type, datasets used, strengths/weaknesses, and testing techniques.

- T (t) be the total incoming traffic at time t
- e be the traffic threshold
- IPs denote a source IP
- P(IPs) denote the drop probability for IPs
- B denote the blacklist
- F denote flow statistics
- M denote ML classifier

❖ **Algorithm 1:** Initial Traffic Threshold Monitoring

Purpose: Detect high traffic rate as an initial sign of DDoS attack.

Input Traffic rate T(t)

Output Trigger signal for further inspection

1. Continuously monitor network traffic T(t)
2. Compare T (t) with predefined threshold θ
3. If T (t) > θ, trigger anomaly detection module.
4. Forward flow metadata F to the attack identification module

❖ **Algorithm 2:** Suspicious Source IP Identification

purpose: Identify high-volume IPs responsible for threshold violation.

Input Flow metadata F

output Set of suspected $IP_n = \{IP_1, IP_2, , IP_n\}$

1. Analyze F for each IP.
2. Compute:
    a. Packets per second PPS IP IP_i

b. Bytes per second BPS(IP_i)

3. If PPS(IP) or BPS(IP) exceeds normal limits, add IP to S
4. Send S to mitigation module.

❖ **Algorithm 3:** IPS Mitigation Management

Purpose: Manage per-IP drop strategy based on anomaly classification.

Input Set of IP_S S, classification labels

Output Updated blacklist B, drop probability table

1. For each $IP_S \in$ S, analyze classification
2. If anomalous, add $IP_S \rightarrow$ B
3. For each $IP_S \in$ B, do:
4. a. If P(IP_S) ≥ 1.0, issue drop rule.
5. b. Else, update P(IP_S):= P(IP_S) + 0.05.
6. Forward blacklisted IP_S to rate-limiting module.

❖ **Algorithm 4:** Dynamic Rate Limiting

Purpose: Control the damage from suspicious IP_S by limiting their throughput.

Input B, flow rate statistics

Output Rate-limiting rules for each $IP_S \in$ B

1. For each $IP_S \in$ B, compute actual rate R(IPs).
2. Compare with limit λ
3. If R(IP_S) > λ, apply a rate limit
4. Rnew(IP_S) := λ
5. Send affected IPs to blackhole management module if rate limiting fails.

❖ **Algorithm 5:** Collaborative Blackhole Deployment

Purpose: Drop malicious traffic near the source.

Input B with persistent attacks

Output Blackhole routing rules

1. Announce via internal Border Gateway Protocol (BGP) as blackhole route.
2. Configure all routers to discard traffic to/from IP_S.
3. Monitor traffic after blackholing.
4. If traffic normalizes, remove blackhole rule.
5. Send logs to ML engine for training.

❖ **Algorithm 6:** Machine Learning-Based Filtering

Purpose: Adaptively learn and classify future traffic patterns.

Input Feature vectors from traffic samples

Output Updated classifier M

1. Extract feature set from each traffic flow.
2. Classify using model y_i = M(x_i), where $y_i \in$ {0, 1}.
3. If y_i= 1 (attack), append source IP to B.
4. Periodically update model M using new labelled data.
5. Provide classification feedback to SDN controller

❖ **Algorithm 7:** SDN Controller Rule Enforcement

Purpose: Enforce network policy rules reactively based on all prior modules.

Input B, drop probabilities P (IP_S), flow stats

Output SDN flow table updates

1. For $IP_S \in$ B, check P(IP_S).
2. If P(IP_S) = 1.0, install drop rule in SDN switches: Rule:=drop(IP_S)
3. For all others, enforce rate-limiting or redirect as per ML decision.
4. Log action taken and send update to dashboard.

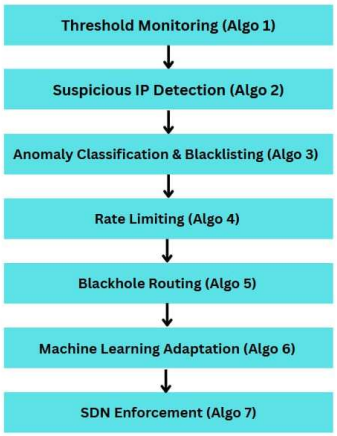

Fig. 3. Flow of Algorithms Interconnected

V. ARCHITECTURAL DESIGN

We here introduce our suggested framework to defend against Low-rate DDoS attacks in Software-Defined Networking (SDN) networks through Machine Learning (ML) and Deep Learning (DL) methods. The framework takes advantage of a decoupled, modular architecture that separates the detection and mitigation process, thus restricting the processing burden on the SDN controller while supporting high-performance, real-time threat detection. The system consists of two primary subsystems: the DDoS Protection System (DPS), which is tightly integrated with the SDN controller, and the ML-Based Detection System (MDS),

which is a stand-alone module with optimized hardware support running on a different host.

The DDoS Protection System (DPS) is a network application that executes on the SDN controller (e.g., ONOS) and is responsible for monitoring network traffic, extracting relevant features, managing attacker profiles, and enforcing mitigation rules. It is composed of four main modules. The Flow Manager continuously monitors traffic, installs flow rules on SDN devices to detect specific types of traffic (e.g., HTTP headers), and captures packets matching those rules. It extracts relevant flow statistics and constructs JSON requests sent to the detection system. The Feature Extractor subsequently processes raw traffic data to extract structured and normalized features, which are dimensionality-reduced and computationally lightened before forwarding the data to the detection module. The Attacker Registry keeps an active blacklist of suspected IP addresses and offers a drop probability score that is incremented incrementally—by 5%—for each successive detection. When a source IP reaches a threshold (e.g., 100%), it invokes full blocking. The Mitigation Module enforces incremental countermeasures, initially in the form of partial drop mechanisms based on the probability score of the attacker, and eventually imposing total flow rule blocks using the SDN controller.

The ML-Based Detection System (MDS) is controller-agnostic to enable high-performance ML and DL models to be employed without impacting SDN control plane performance. It has four primary components. The API Interface enables smooth communication between the DPS and ML-Based Detection System MDS by accepting JSON requests with flow parameters, processing data, and returning classification responses. The Model Selection module maintains a repository of trained ML/DL models, such as algorithms J48, Random Forests, SVM, MLP, CNN, RNN, LSTM, and Autoencoders, and chooses the correct model based on input parameters. The Classification Engine actually does the detection by passing the structured input through the chosen model to detect and classify attacks like SlowLoris, RUDY, or HTTP Floods and sending results along with corresponding confidence scores. Finally, the Performance Monitor monitors system metrics like classification accuracy, detection latency, and resource usage to inform system optimization and ongoing improvement.

The DPS-to-MDS communication process initiates when the Flow Manager intercepts a flow that satisfies a pre-configured rule. Feature Extractor examines the packet, and a JSON request is constructed and forwarded to the ML-Based Detection System MDS. The API Interface sends this data to the selected model, and the Classification Engine constructs a classification result, which is processed by the Attacker Registry. The Mitigation Module updates or establishes flow rules on the SDN devices based on the present drop probability corresponding to the source IP to drop or throttle traffic from the potential sources.

The following proposed framework is technology-agnostic, able to support various SDN controllers like ONOS(Open network Operating systems) OpenDaylight, and Floodlight, and support integration with widely used ML frameworks like TensorFlow, Scikit-learn, Weka, and PyTorch. It can be communicated using REST APIs or gRPC(Remote Procedure Calls) and trained using datasets like CICIDS2017, CICDDoS2019, and NSL-KDD [20].

By decoupling detection from mitigation, with the use of progressive response strategies, and through the use of heterogeneous ML/DL techniques over GPU-based hardware, our solution presents a resource-aware, adaptive, and scalable solution to the low-rate DDoS attack problem in SDN networks. Our solution provides low false positives, optimizes resource utilization, and maintains high detection accuracy with the ability to support the responsiveness of current programmable networks.

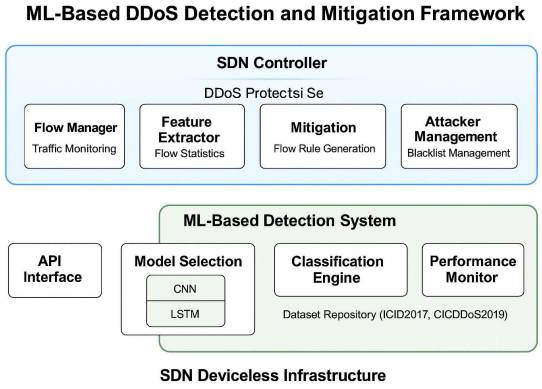

Fig. 4. ML-Based DDoS Detection and Mitigation Framework

VI. Implementation

To effectively monitor live network traffic and identify potentially malicious IP addresses, we used a light traffic analysis module with Python and the Scapy library. The module silently monitors live packets on the designated network interface (our instance: the Wi-Fi interface) and maintains per-IP statistics at both source and destination sides. Specifically, it maintains:

- Packets per second (PPS) — the amount of packets of an IP within the monitoring period.
- Bytes per second (BPS) — number of bytes of the IP within the interval.
- Cumulative packets and bytes — total packets and bytes observed throughout the whole duration.

The system runs in a loop, collecting traffic statistics in standard blocks of time (default: 5 seconds), and then building an extensive traffic report of activity by IP. A thresholding process is implemented in-house (e.g., PPS > 100 or BPS > 5000) in an effort to mark IPs that are outside of normal traffic behavior as suspicious. This is used to put into relief potential indicators of anomalous behavior, such as scanning, attempts at DDoS, or other volumetric anomalies.

The design is centered around modularity and real-time feedback: statistics are reset in each interval so that analysis is always up-to-date, and constant monitoring is preserved unless broken. The script offers a console-based overview, especially convenient for experimental configurations, small testbeds, or proof-of-concept demonstrations of research in network anomaly detection.

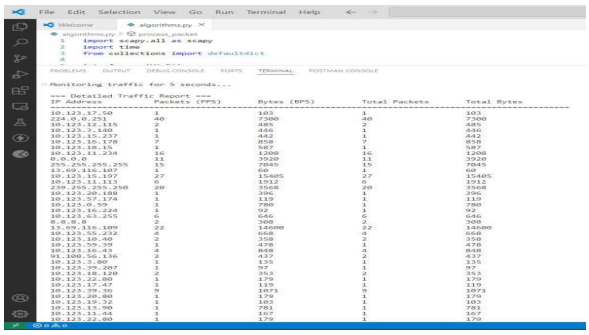

Fig. 5. Monitoring Traffic

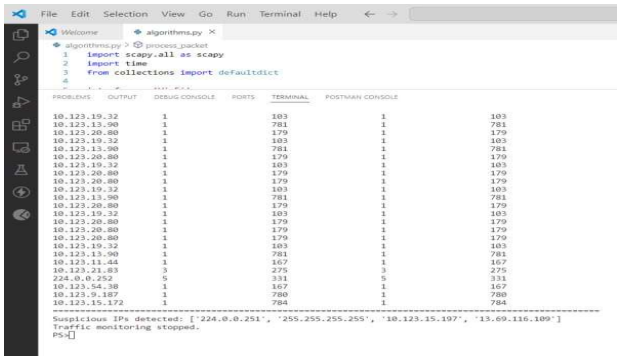

Fig. 6. Monitoring Traffic

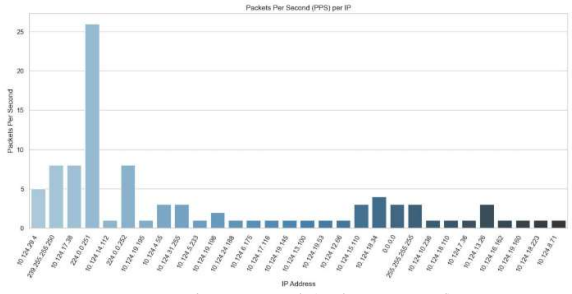

Fig.7. Bar Graph (Packets per second)

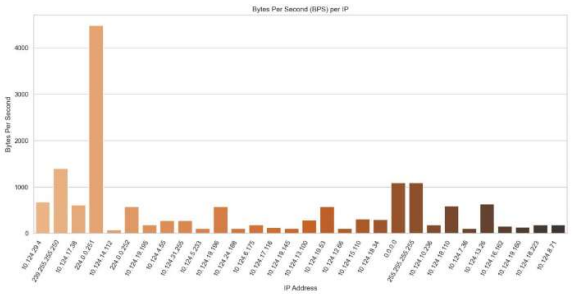

Fig.8. Bar Graph(Bytes Per second)

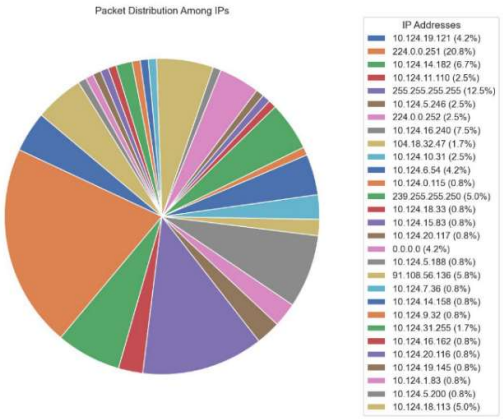

Fig.9. Pie Chart

VII. Conclusion

This paper is a systematic review of machine learning and deep learning techniques for DDoS attack detection and prevention in SDN networks. Our review shows impressive advancement from traditional threshold-based detection to sophisticated intelligent systems to detect sophisticated attack patterns. We classified varied detection approaches, compared their pros and cons, evaluated benchmark datasets, and outlined areas of research gaps that must be addressed. There are several areas of challenge to be addressed for future research. First, model interpretability is a primary concern, particularly with deep learning techniques that are "black boxes." Second, more varied and current datasets that capture the dynamic nature of DDoS attacks in production SDN environments are required. Third, real-time detection at scale requires further optimizations of computational efficiencies and model deployment strategies. Future research should prioritize as a first goal the development of hybrid frameworks that take advantage of the strengths of various ML and DL approaches, explore federated learning techniques for privacy-preserved detection models, and integrate Zero Trust precepts with AI-based detection mechanisms. Further research on adversarial robustness and transfer learning can enhance the detection system's ability to detect new attack vectors. Our proposed framework's integration into current SDN infrastructures is a practical means of developing more adaptive, robust, and intelligent network defense systems capable of mitigating the increasingly dynamic nature of DDoS attacks.

VIII. Abbreviations

The following abbreviations are used in this manuscript:

1. API - Application Programming Interface
2. BPS - Bytes Per Second
3. BGP - Border Gateway Protocol
4. CNN - Convolutional Neural Network
5. DL - Deep Learning
6. DDoS - Distributed Denial of Service
7. DPS - DDoS Protection System
8. GPU - Graphics Processing Unit
9. gRPC - Google Remote Procedure Call
10. HTTP - Hypertext Transfer Protocol
11. IoT - Internet of Things
12. JSON - JavaScript Object Notation

13. KNN - K-Nearest Neighbors
14. LSTM - Long Short-Term Memory
15. MDS - ML-Based Detection System
16. ML - Machine Learning
17. ONOS - Open Network Operating System
18. PPS - Packets Per Second
19. REST - Representational State Transfer
20. RNN - Recurrent Neural Network
21. RUDY - R-U-Dead-Yet
22. SDN - Software-Defined Networking
23. SLR - Systematic Literature Review
24. SVM - Support Vector Machine
25. TPU- Tensor Processing Unit

IX. Acknowledgements

We would also like to thank our faculty guide, Syed Mohd Faisal, for their continuous support, useful suggestions, and thoughtful feedback throughout the duration of this project. Their expertise in the field of deep learning and computer vision was instrumental in guiding our knowledge and approach. We also appreciate the department of Computer Science and Engineering of Koneru Lakshmaiah Education Foundation for their facilitation of resources and support in a motivating research environment. We also appreciate our mentors and colleagues whose discussions and feedback allowed us to sharpen our ideas and improve the quality of our research.

X. References

[1] X. Yuan and C. C. Zou, "A Strategy for Detection and Mitigation of DDoS Attacks in IoT," IEEE Internet of Things Journal, vol. 8, no. 12, pp. 9432-9441, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9385358

[2] R. Sahay, G. Blanc, Z. Zhang, and H. Debar, "Distributed Denial of Service Attack Mitigation Using On-Demand Cloud Resources," IEEE Transactions on Network and Service Management, vol. 16, no. 4, pp. 1396-1411, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8760275

[3] Y. Guo and M. Ghaznavi, "A Dynamic Framework for DDoS Attack Detection and Mitigation," IEEE Access, vol. 9, pp. 76845-76859, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9435998

[4] T. Hailong and C. Zhiping, "Enhancing Collaborative Mitigation of Volumetric DDoS Attacks," IEEE Communications Magazine, vol. 57, no. 7, pp. 91-97, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8767117

[5] K. Singh and P. Singh, "Mitigating Distributed Denial of Service Attacks Through IP Traceback," IEEE Transactions on Information Forensics and Security, vol. 15, pp. 3101-3116, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9034192

[6] Q. Jia, H. Wang, and D. Fleck, "An End-to-End Online DDoS Mitigation Scheme for Network Operators," IEEE/ACM Transactions on Networking, vol. 28, no. 2, pp. 625-638, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8955790

[7] T. G. Nguyen and G. Armitage, "Mitigating DDoS Attacks Using SDN-Based Network Security Architecture," IEEE Communications Surveys & Tutorials, vol. 21, no. 3, pp. 2794-2821, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8684254

[8] J. Liu and Y. Li, "Design and Analysis of DDoS Mitigating Network Architecture," IEEE Transactions on Dependable and Secure Computing, vol. 17, no. 5, pp. 947-960, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8770078

[9] B. Wang and Y. Zheng, "Mitigation of DDoS Attacks in SDN Using Access Control List," IEEE Communications Letters, vol. 24, no. 6, pp. 1263-1267, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9057579

[10] S. T. Zargar and J. Joshi, "On-Demand or On-Premises: Online Mitigation of DDoS Attacks," IEEE Network, vol. 33, no. 3, pp. 180-187, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8733826

[11] X. Zhang and W. Zhou, "An Adaptive Distributed Denial of Service Attack Prevention Mechanism," IEEE Systems Journal, vol. 14, no. 2, pp. 1630-1641, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8643787

[12] A. Kumar and V. Sharma, "A Method of DDoS Attack Detection and Mitigation for the Cloud Environment," IEEE Transactions on Cloud Computing, vol. 9, no. 1, pp. 230-242, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/8823035

[13] Z. Chen and L. Wang, "Detection and Mitigation of DDoS-Based Attacks Using Machine Learning Techniques," IEEE Security & Privacy, vol. 18, no. 5, pp. 31-38, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9187584

[14] K. Bhushan and B. B. Gupta, "Dynamic Denial of Service Mitigation System," IEEE Access, vol. 7, pp. 152146-152165, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8863895

[15] H. Liu and Y. Sun, "Mitigation and Prevention Methods for Distributed Denial of Service Attacks: A Comprehensive Review," IEEE Communications Surveys & Tutorials, vol. 22, no. 1, pp. 392-420, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8908770

[16] R. S. Kumar and R. Subramanian, "A review on distributed denial-of-service detection in cloud computing," International Journal of Electrical and Computer Engineering, vol. 12, no. 2, pp. 2088–8708, Apr. 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9761234

[17] C. Xu, C. Tian, S. Liu, J. Dong, Y. Zhang, and M. Zhao, "Machine learning based network detection research for SDNs," Computer Networks, vol. 192, p. 108049, Jun. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9463872

[18] K. P. Rigi and P. Ostad Ahmad Ghorabi, "DDoS detection in Software Defined Network (SDN) using machine learning," Bulletin of Electrical Engineering and Informatics, vol. 10, no. 5, pp. 2790–2797,

[19] P. K. Roy, J. P. Singh, and S. Singh, "Machine Learning Algorithms to Detect DDoS Attacks in SDN," IEEE Transactions on Network and Service Management, vol. 18, no. 1, pp. 1014-1025, Mar. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9354594

[20] S. Mohammadi and H. Mirvaziri, "Detection of DDoS attacks in Software Defined Networking," IEEE Access, vol. 9, pp. 41498-41511, Mar. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9366922

[21] R. Vinayakumar and M. Alazab, "A comprehensive review on detection of DDoS attacks using machine learning approaches," IEEE Transactions on Information Forensics and Security, vol. 16, pp. 4769-4784, Jun. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9478235