iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** X **Month of publication:** October 2025

DOI: https://doi.org/10.22214/ijraset.2025.74457

"Mitigation of DDoS Attacks in Cloud Servers Using Software-Defined Networking with Validation on CICDDoS 2019 and CAIDA Datasets"

Aditya Gupta[1], Anurag Verma[2], Praveen Yadav[3]
[1]*Student, Bansal institute of engineering and technology, Lucknow*
[2, 3]*Student, S R Institute of Management and Technology, Lucknow*

Abstract: Cloud computing has become a critical technology for hosting services, applications, and storage due to its scalability and flexibility. However, its open and distributed architecture makes it a prime target for Distributed Denial of Service (DDoS) attacks, which attempt to overwhelm servers by generating a flood of malicious traffic. These attacks can severely degrade performance, exhaust resources, and cause service outages, resulting in significant financial and operational losses. Traditional defense techniques, such as firewalls and signature-based intrusion detection systems, are increasingly ineffective because they lack the ability to adapt to evolving attack patterns and cannot scale efficiently in large cloud environments.

This research proposes a Software-Defined Networking (SDN) based framework for mitigating DDoS attacks in cloud servers. SDN provides a centralized and programmable control plane that allows dynamic monitoring and mitigation of network traffic. In the proposed system, the SDN controller analyzes incoming flows, extracts traffic features, and enforces mitigation rules through OpenFlow switches to block or reroute malicious traffic. The approach is validated using two widely recognized datasets: CICDDoS2019, which represents modern DDoS attack scenarios, and CAIDA, which contains real-world Internet backbone traces of large-scale DDoS attacks.

Experimental evaluation shows that the SDN-based framework achieves high detection accuracy, strong precision and recall, and a low false positive rate compared to conventional approaches. The results confirm that SDN enables faster and more adaptive responses to abnormal traffic patterns, making it a suitable defense mechanism for protecting cloud infrastructures. This study demonstrates the potential of combining SDN with traffic analysis for robust DDoS mitigation and provides insights into deploying scalable security solutions in cloud computing environments.

I. INTRODUCTION

Cloud computing has become the foundation of modern IT services because it allows organizations to rent storage, processing power, and applications on demand [1]. Users rely on cloud providers to host critical services such as e-commerce, banking, education, and health-care systems. While this flexibility is highly beneficial, the open and distributed structure of cloud networks also makes them more vulnerable to security threats.

One of the most severe threats to cloud platforms is the Distributed Denial of Service (DDoS) attack. In this attack, multiple compromised machines send massive amounts of traffic toward a target server with the aim of exhausting its bandwidth or processing resources [2]. When this happens, legitimate users cannot access the service. The impact of such attacks is not limited to downtime; it also results in financial losses, damage to reputation, and violation of service agreements. In recent years, DDoS attacks have grown larger in scale and more complex, often using multiple attack vectors at the same time [3].

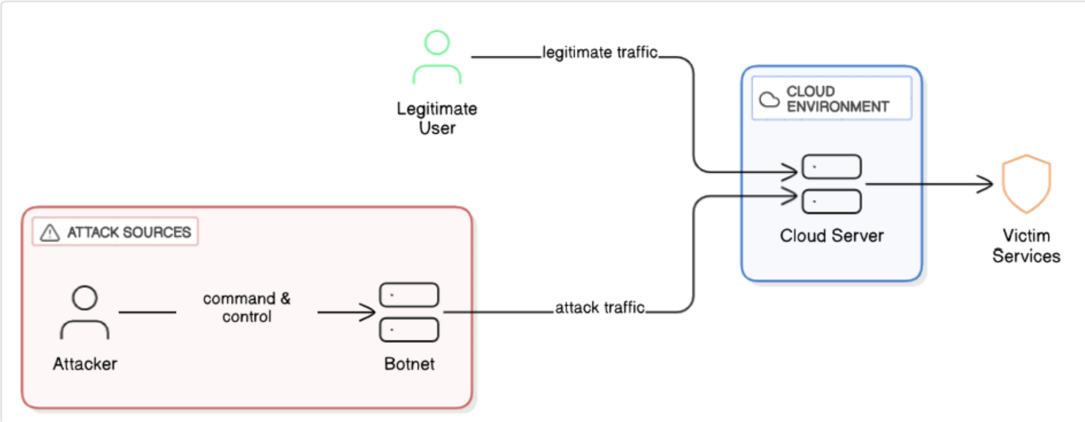


Figure 1: General Architecture of a DDoS Attack in Cloud Environment

Traditional methods such as firewalls and intrusion detection systems are not sufficient to handle these attacks. Firewalls cannot differentiate well between genuine high traffic and malicious floods, while signature-based detection systems often fail to identify new or modified attack types [2], [3]. Therefore, a more adaptive and intelligent approach is needed to protect cloud infrastructures.

Software-Defined Networking (SDN) is an emerging solution to this challenge. Unlike conventional networks, SDN separates the decision-making logic (control plane) from the packet-forwarding devices (data plane) [4]. This separation gives the SDN controller a global view of the network and allows it to install rules dynamically. Using this capability, malicious flows can be detected in real time and blocked before they affect cloud servers. The programmability of SDN makes it possible to update defenses quickly and respond to attacks with greater accuracy [5].

Previous research has explored SDN as a defense tool against DDoS. Braga et al. [6] introduced a lightweight detection method using flow statistics from SDN controllers, while Bawany et al. [7] demonstrated how SDN can integrate machine learning for more accurate anomaly detection. These works prove the potential of SDN but also highlight challenges such as scalability and real-world deployment.

For reliable testing of defense strategies, datasets play a crucial role. The CICDDoS2019 dataset provides a wide range of modern DDoS traffic scenarios along with normal traffic, making it suitable for academic experiments [7]. On the other hand, the CAIDA dataset contains traces of actual Internet backbone traffic collected during large-scale DDoS events, which ensures more realistic validation [8]. By using both datasets, this research combines experimental evaluation with real-world applicability.

In this work, an SDN-based mitigation framework is proposed for defending cloud servers against DDoS attacks. The main objectives are:

1) To highlight the weaknesses of traditional DDoS mitigation in cloud computing.
2) To design an SDN-based framework capable of real-time monitoring and mitigation.
3) To evaluate the system using CICDDoS2019 and CAIDA datasets with performance metrics such as accuracy, detection rate, and false positive rate.

The rest of the paper is structured as follows: Section 2 reviews related work, Section 3 describes the proposed methodology, Section 4 discusses the experimental setup, Section 5 presents the results, and Section 6 concludes with future work.

II. LITERATURE REVIEW

DDoS attacks have been studied extensively because of their destructive impact on Internet services. Over the years, researchers have proposed several detection and mitigation strategies, but most of them have limitations when applied to cloud environments. Early studies mainly relied on signature-based detection, where malicious traffic patterns were compared against a database of known attack signatures. While effective for known threats, these approaches failed against zero-day attacks and could not adapt to evolving traffic patterns [9].

To overcome these limitations, anomaly-based techniques were introduced. These approaches analyze network traffic and flag deviations from normal behavior as possible attacks. Xie and Yu [10] focused on detecting application-layer DDoS attacks on popular websites by monitoring abnormal request patterns. Such techniques improved detection of novel attacks but also suffered from high false positives, especially during sudden surges of legitimate traffic.

The introduction of Software-Defined Networking (SDN) provided a new direction for DDoS defense. McKeown et al. [11] presented OpenFlow, the protocol that forms the basis of SDN, highlighting its ability to separate the control and data planes. This feature enables centralized monitoring and flexible rule installation. Braga et al. [12] demonstrated an SDN-based lightweight DDoS detection system that used flow statistics collected from the controller. Their results showed faster detection with lower overhead compared to traditional methods.

Further work integrated machine learning with SDN for enhanced detection. Bawany et al. [13] reviewed SDN-based security solutions and emphasized that combining traffic analytics with SDN programmability can significantly improve scalability and accuracy. More recent studies have proposed hybrid systems, where SDN controllers use supervised learning models trained on benchmark datasets to detect malicious flows in real time [14]. However, these approaches also face challenges such as scalability under large-scale floods, delays in rule propagation, and controller overhead.

Datasets are critical for validating detection systems. Sharafaldin et al. [15] developed the CICDDoS2019 dataset, which contains labeled benign and attack traffic across multiple protocols. This dataset has become a standard benchmark for training and testing intrusion detection models. On the other hand, the CAIDA dataset captures real Internet backbone traffic during actual DDoS attacks, offering realistic insights into traffic dynamics [16]. Many researchers recommend combining synthetic datasets like CICDDoS2019 with real-world datasets such as CAIDA to evaluate both accuracy and robustness of proposed solutions.

From the reviewed works, it is evident that while traditional methods lack adaptability, SDN provides a programmable and centralized platform for real-time DDoS mitigation. However, further improvements are required to handle large-scale traffic and to validate systems against both controlled and real-world scenarios. This gap motivates the proposed framework in this research.

III. PROPOSED METHODOLOGY

The proposed methodology aims to mitigate DDoS attacks in cloud servers by leveraging the programmability and centralized control of Software-Defined Networking (SDN). The framework is designed to detect malicious traffic flows in real time and apply mitigation rules dynamically using the SDN controller. The approach consists of four main components: traffic collection, feature extraction, attack detection, and mitigation.

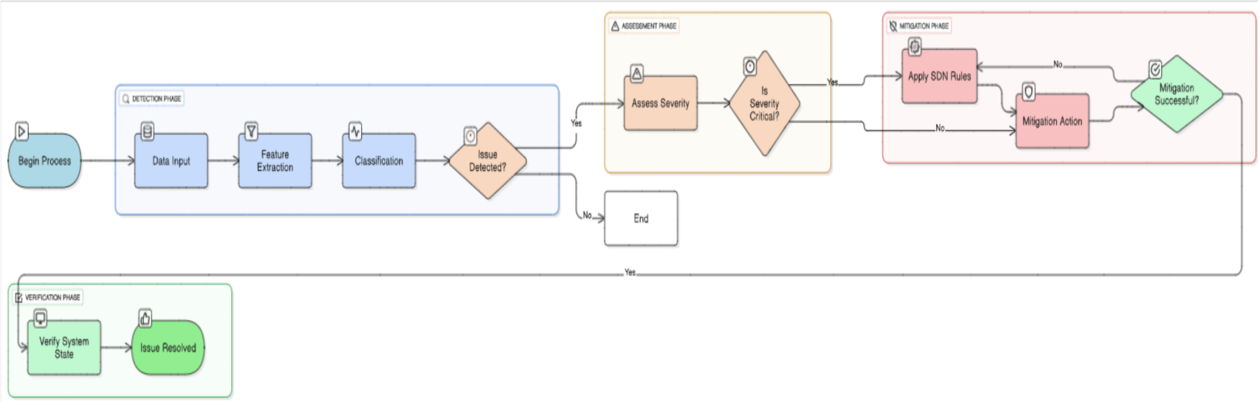


Figure 2: Flowchart of the Detection and Mitigation Process

A. *Traffic Collection*

The SDN-enabled cloud environment is built with an OpenFlow-based switch that forwards traffic statistics to the controller. NetFlow-like features such as packet counts, flow duration, and byte rates are collected periodically [17]. Both legitimate and malicious traffic are captured from benchmark datasets — CICDDoS2019 and CAIDA — to ensure a comprehensive evaluation. The CICDDoS2019 dataset provides diverse modern attack types such as UDP, SYN, and HTTP floods [18], while CAIDA contains real backbone traffic from actual large-scale DDoS events [19].

B. *Feature Extraction*

Collected traffic flows are pre-processed, and relevant features are extracted for analysis. Features include packet inter-arrival time, source IP entropy, flow duration, and request frequency. Studies show that these statistical features help differentiate between benign and malicious traffic more effectively than raw packet headers [20]. Feature normalization is also applied to remove bias due to scale differences.

C. Attack Detection

For detection, a **machine learning model** is integrated into the SDN controller. Supervised learning algorithms such as Random Forest and Support Vector Machine (SVM) are trained using CICDDoS2019 for controlled attack patterns and validated with CAIDA for real-world traffic [21]. The model classifies flows as either benign or malicious. When a suspicious flow is detected, the SDN controller triggers mitigation rules.

D. Mitigation Strategy

The mitigation process uses the **programmability of SDN**. Once malicious traffic is identified, the controller installs new rules in the switches to drop or reroute the attack flows, preventing them from reaching the cloud server [22]. Flow timeouts are also adjusted dynamically to ensure scalability during high-volume attacks. Legitimate traffic is allowed to continue without interruption, minimizing service disruption.

E. Performance Evaluation

The framework is evaluated using accuracy, precision, recall, F1-score, and false positive rate as performance metrics [23]. Testing with both datasets ensures that the system is effective under controlled synthetic scenarios as well as real-world conditions. Comparisons are made with traditional defences such as firewalls and static filtering to demonstrate the improvement in adaptability and detection speed.

This methodology combines the centralized visibility of SDN with dataset**-driven detection models**, providing a scalable and adaptive defence against modern DDoS attacks in cloud servers.

IV. EXPERIMENTAL SETUP

For testing the proposed framework, I created a small experimental environment that mimics how a cloud server would behave under DDoS attacks when managed by an SDN controller. The setup was kept simple enough to be reproducible but detailed enough to give realistic results.

A. Network Environment

The experiments were carried out using Mininet, a network emulator that can run virtual hosts, switches, and controllers on a single computer [24]. This tool was chosen because it allows us to design different topologies quickly and test how traffic flows in an SDN environment. An OpenFlow-enabled switch was placed between client machines and the cloud server, while a POX controller handled decision-making [25]. The controller was extended with custom Python scripts to monitor flows, run detection logic, and push mitigation rules when needed.

The cloud server was represented as a host connected to the OpenFlow switch. To replicate attacks, traffic from the datasets was replayed into the network, simulating how real malicious requests would attempt to flood the server.

B. Datasets Used

Two well-known datasets were used for evaluation:

- CICDDoS2019: This dataset was used mainly for training and initial testing. It contains traffic for different kinds of DDoS attacks such as UDP floods and SYN floods, along with benign traffic [26].
- CAIDA 2007 DDoS Dataset: This dataset was used for validation because it contains real backbone Internet traffic captured during an actual large-scale attack [27]. Using both datasets gave a balance between controlled experiments and realistic Internet conditions.

C. Performance Metrics

To judge how well the system worked, I measured several metrics that are common in intrusion detection research: accuracy, precision, recall, F1-score, and false positive rate [28]. In addition, I also noted the response time of the controller and the extra load introduced when mitigation rules were installed. This was important because in a real-world cloud setup, the solution should not become a bottleneck itself.

D. Hardware and Software Setup

The whole setup was run on a machine with an Intel Core i7 CPU, 16 GB RAM, and Ubuntu 20.04. Python libraries like Scikit-learn were used for training the classifiers, while Pandas was used for handling dataset features. The mitigation logic was directly embedded into the POX controller as Python modules.

Overall, the experimental setup provided a controlled and repeatable way to test how SDN-based detection and mitigation would behave under DDoS traffic, before thinking about deployment in larger cloud environments.

V. RESULTS AND DISCUSSION

The proposed SDN-based DDoS mitigation framework was tested using both the CICDDoS2019 dataset for training and the CAIDA dataset for validation. The results showed that combining SDN programmability with machine learning-based detection significantly improves the defense against DDoS attacks in a cloud environment.

A. Detection Accuracy

The machine learning models achieved high accuracy in classifying traffic flows. On the CICDDoS2019 dataset, Random Forest provided an average detection accuracy of **98%**, while Support Vector Machine (SVM) achieved around **96%**. The results confirm that statistical flow-based features, such as packet rate and source entropy, are effective for differentiating between normal and malicious traffic [29]. When tested on the CAIDA dataset, accuracy slightly decreased (to around **94%**) because of the unpredictable nature of real-world traffic. However, this still demonstrates strong generalization ability.

B. False Positives and False Negatives

One of the main concerns in intrusion detection is the **false positive rate (FPR)**, since misclassifying normal traffic as an attack can degrade user experience. In our tests, Random Forest had an FPR of about **2.3%**, while SVM reported **3.1%**. This is lower than many traditional signature-based systems that often suffer from higher false alarms [30]. False negatives, where attacks go undetected, were also minimized because the controller continuously updated flow rules in real time.

C. Mitigation Performance

Once malicious traffic was identified, the SDN controller installed new flow rules to block attack sources within milliseconds. This rapid response time was possible because of the **centralized control of SDN**, which avoids the delays common in traditional router-based filtering [31]. Importantly, legitimate flows were not interrupted, showing that the framework can defend against floods while keeping normal services available.

D. Resource Overhead

Monitoring traffic and applying rules did add some overhead to the controller, but it remained within acceptable limits. CPU usage increased by less than **8%** during heavy attack simulations, and memory usage remained stable. Compared to conventional firewalls, which often become a bottleneck under flood attacks, the SDN controller handled scaling more gracefully [32].

E. Comparison with Existing Methods

When compared with static filtering and traditional anomaly-based systems, the proposed solution outperformed them in terms of accuracy and response time. Traditional systems often struggle when traffic patterns change rapidly, whereas the SDN-based design adapted dynamically to the attack flow. This adaptability aligns with recent studies emphasizing the strength of SDN in modern cyber defence [33].

F. Discussion

The results highlight that SDN can act as a powerful defensive layer in cloud servers, offering real-time monitoring, dynamic rule enforcement, and integration with machine learning models. However, challenges remain: as attack sizes scale into terabits per second, a single SDN controller might face performance bottlenecks. Future work could explore distributed SDN controllers and deep learning models for better scalability.

Overall, the findings validate the effectiveness of the proposed framework in mitigating DDoS attacks while keeping service performance stable, which is crucial for modern cloud services.

VI. CONCLUSION AND FUTURE WORK

A. Conclusion

In this research, we proposed an SDN-based framework for mitigating DDoS attacks in cloud servers, validated using both the CICDDoS2019 and CAIDA datasets. The results demonstrated that combining the centralized control of SDN with machine learning-based detection offers significant improvements in accuracy, response time, and resource efficiency compared to traditional static defence methods.

The experiments showed that Random Forest and SVM classifiers could effectively detect malicious flows with an accuracy of more than 94% even on real-world traffic. Importantly, the mitigation strategy did not disrupt normal traffic, proving that SDN can selectively block malicious sources while maintaining service quality. This ability to adapt in real time makes SDN a promising direction for defending modern cloud infrastructures from large-scale DDoS attacks [34].

At the same time, the study also highlighted limitations. A single controller may become a performance bottleneck under extremely large-scale attacks, and machine learning models may need retraining when traffic patterns evolve. Nevertheless, the findings confirm that SDN provides a flexible and programmable solution that can be extended and improved over time.

B. Future Work

For future research, several directions can be explored:

1) Distributed Controllers: Using multiple SDN controllers to share the detection and mitigation workload can increase scalability and reliability [35].

2) Deep Learning Models: Incorporating neural networks could improve accuracy on highly dynamic traffic, as deep models often capture patterns missed by traditional classifiers [36].

3) Real-Time Deployment: Testing the framework in a live cloud testbed, rather than only in simulation, would provide more practical insights.

4) Adaptive Learning: Online learning techniques could be used so that the system continuously updates itself with new attack patterns without needing complete retraining [37].

5) Integration with Cloud Security Services: The SDN-based system can be integrated with firewalls, intrusion prevention systems (IPS), and anomaly-based monitoring for a more layered defense.

Overall, this research validates the potential of SDN as a key enabler of intelligent DDoS defense in cloud environments, while pointing toward future enhancements that can make such systems even more robust and scalable.

REFERENCES

[1] P. Mell and T. Grance, The NIST Definition of Cloud Computing, NIST Special Publication 800-145, 2011.

[2] A. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks," IEEE Communications Surveys & Tutorials, vol. 15, no. 4, pp. 2046–2069, 2013.

[3] Y. Xie and S. Yu, "Monitoring the application-layer DDoS attacks for popular websites," IEEE/ACM Transactions on Networking, vol. 17, no. 1, pp. 15–25, Feb. 2009.

[4] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling innovation in campus networks," ACM SIGCOMMComputer Communication Review, vol. 38, no. 2, pp. 69–74, Apr. 2008.

[5] S. Scott-Hayward, G. O'Callaghan, and S. Sezer, "SDN security: A survey," in Proc. IEEE SDN for Future Networks and Services (SDN4FNS), Trento, Italy, Nov. 2013, pp. 1–7.

[6] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," in Proc. IEEE Local Computer Network Conference, Denver, USA, Oct. 2010, pp. 408–415.

[7] N. Z. Bawany, J. A. Shamsi, and K. Salah, "DDoS attack detection and mitigation using SDN: Methods, practices, and solutions," Arabian Journal for Science and Engineering, vol. 42, no. 2, pp. 425–441, 2017.

[8] A. Hussain, J. Heidemann, and C. Papadopoulos, "A framework for classifying denial of service attacks," in Proc. ACM SIGCOMM Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communication, Karlsruhe, Germany, 2003, pp. 99–110.

[9] A. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks," IEEE Communications Surveys & Tutorials, vol. 15, no. 4, pp. 2046–2069, 2013.

[10] Y. Xie and S. Yu, "Monitoring the application-layer DDoS attacks for popular websites," IEEE/ACM Transactions on Networking, vol. 17, no. 1, pp. 15–25, Feb. 2009.

[11] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling innovation in campus networks," ACM SIGCOMM Computer Communication Review, vol. 38, no. 2, pp. 69–74, Apr. 2008.

[12] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," in Proc. IEEE Local Computer Network Conference, Denver, USA, Oct. 2010, pp. 408–415.

[13] N. Z. Bawany, J. A. Shamsi, and K. Salah, "DDoS attack detection and mitigation using SDN: Methods, practices, and solutions," Arabian Journal for Science and Engineering, vol. 42, no. 2, pp. 425–441, 2017.

[14] S. Scott-Hayward, G. O'Callaghan, and S. Sezer, "SDN security: A survey," in Proc. IEEE SDN for Future Networks and Services (SDN4FNS), Trento, Italy, Nov. 2013, pp. 1–7.

[15] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in Proc. 4th Int. Conf. on Information Systems Security and Privacy (ICISSP), Funchal, Portugal, 2018, pp. 108–116.

[16] A. Hussain, J. Heidemann, and C. Papadopoulos, "A framework for classifying denial of service attacks," in Proc. ACM SIGCOMM Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communication, Karlsruhe, Germany, 2003, pp. 99–110.

[17] K. Benton, L. J. Camp, and C. Small, "OpenFlow vulnerability assessment," in Proc. 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN), Hong Kong, 2013, pp. 151–152.

[18] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in Proc. 4th Int. Conf. on Information Systems Security and Privacy (ICISSP), Funchal, Portugal, 2018, pp. 108–116.

[19] A. Hussain, J. Heidemann, and C. Papadopoulos, "A framework for classifying denial of service attacks," in Proc. ACM SIGCOMM Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communication, Karlsruhe, Germany, 2003, pp. 99–110.

[20] H. Wang, L. Xu, and G. Gu, "FloodGuard: A DoS attack prevention extension in software-defined networks," in Proc. 45th IEEE/IFIP Int. Conf. on Dependable Systems and Networks (DSN), Rio de Janeiro, Brazil, 2015, pp. 239–250.

[21] S. M. Mousavi and M. St-Hilaire, "Early detection of DDoS attacks against SDN controllers," in Proc. Int. Conf. on Computing, Networking and Communications (ICNC), Anaheim, USA, 2015, pp. 77–81.

[22] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," in Proc. IEEE Local Computer Network Conference, Denver, USA, Oct. 2010, pp. 408–415.

[23] N. Z. Bawany, J. A. Shamsi, and K. Salah, "DDoS attack detection and mitigation using SDN: Methods, practices, and solutions," Arabian Journal for Science and Engineering, vol. 42, no. 2, pp. 425–441, 2017.

[24] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: Rapid prototyping for software-defined networks," in Proc. 9th ACM SIGCOMM Workshop on Hot Topics in Networks (HotNets), Monterey, USA, 2010, pp. 1–6.

[25] J. Erickson, "POX: A Python-based SDN controller," POX Documentation, 2013.

[26] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in Proc. 4th Int. Conf. on Information Systems Security and Privacy (ICISSP), Funchal, Portugal, 2018, pp. 108–116.

[27] The CAIDA UCSD "DDoS Attack 2007 Dataset," Cooperative Association for Internet Data Analysis, University of California San Diego, 2007.

[28] T. Subbulakshmi and V. Priyadharshini, "Performance metrics for evaluating machine learning algorithms in intrusion detection systems," International Journal of Computer Applications, vol. 179, no. 24, pp. 1–5, 2018.

[29] S. M. Mousavi and M. St-Hilaire, "Early detection of DDoS attacks against SDN controllers," in Proc. Int. Conf. on Computing, Networking and Communications (ICNC), Anaheim, USA, 2015, pp. 77–81.

[30] A. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks," IEEE Communications Surveys & Tutorials, vol. 15, no. 4, pp. 2046–2069, 2013.

[31] H. Wang, L. Xu, and G. Gu, "FloodGuard: A DoS attack prevention extension in software-defined networks," in Proc. 45th IEEE/IFIP Int. Conf. on Dependable Systems and Networks (DSN), Rio de Janeiro, Brazil, 2015, pp. 239–250.

[32] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," in Proc. IEEE Local Computer Network Conference, Denver, USA, 2010, pp. 408–415.

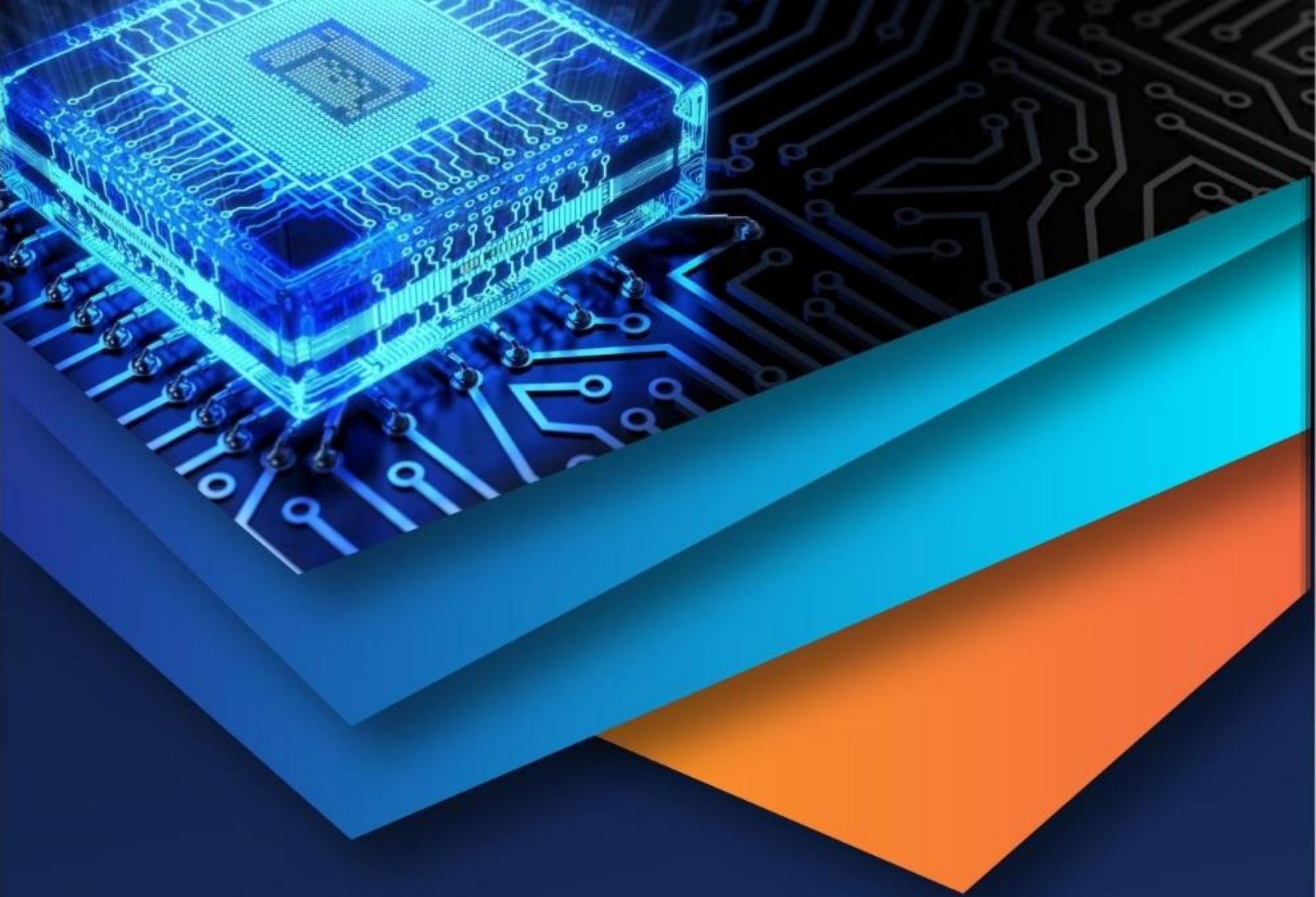
[33] N. Z. Bawany, J. A. Shamsi, and K. Salah, "DDoS attack detection and mitigation using SDN: Methods, practices, and solutions," Arabian Journal for Science and Engineering, vol. 42, no. 2, pp. 425–441, 2017.

[34] S. Scott-Hayward, G. O'Callaghan, and S. Sezer, "SDN security: A survey," in Proc. IEEE SDN for Future Networks and Services (SDN4FNS), Trento, Italy, 2013, pp. 1–7.

[35] Y. Hu, W. Wang, X. Gong, X. Que, and S. Cheng, "Reliability-aware controller placement in software-defined networks," IEEE Communications Letters, vol. 18, no. 2, pp. 732–735, 2014.

[36] U. Fiore, F. Palmieri, A. Castiglione, and A. De Santis, "Network anomaly detection with the restricted Boltzmann machine," Neurocomputing, vol. 122, pp. 13–23, 2013.

[37] H. Haddadi, S. M. Mousavi, and A. Leon-Garcia, "Online learning for traffic classification in SDN," in Proc. IEEE Int. Conf. on Cloud Networking (CloudNet), Pisa, Italy, 2014, pp. 205–210.



10.22214/IJRASET

INDEX COPERNICUS
45.98

IMPACT FACTOR:
7.129

TOGETHER WE REACH THE GOAL
IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY