



Adaptive DDoS Protection Architecture for Cloud Servers: A Custom Traffic Filtering Tool

Mr. Yash Narendra Date¹, Asst. Prof. Sapana Fegade²

Research Scholar, Department of Computer Applications, Shramsadhana College of Engineering & Technology
Jalgaon.MH, India¹

Asst. Professor, Department of Computer Applications, Shramsadhana College of Engineering & Technology
Jalgaon.MH, India²

Abstract: Cloud computing architectures are highly susceptible to Distributed Denial of Service (DDoS) attacks that can exceed server processing capabilities. A volumetric DDoS attack overwhelms a targeted system with a massive volume of unwanted network traffic, causing severe CPU and memory exhaustion and preventing legitimate user requests from being processed. This research project presents a localized DDoS protection architecture designed for small to medium-size cloud environments. The proposed approach employs a custom C++ traffic filtering engine positioned at the network edge, utilizing a hash map data structure to track incoming IP addresses and their corresponding request rates. Controlled synthetic traffic is generated using an asynchronous Python script, and the protected and unprotected server states are compared using CPU utilization, memory consumption and request latency as primary evaluation metrics. The experimental results demonstrate that the filtering mechanism can reduce server CPU utilization from 98.7% to 22.4% while maintaining a legitimate request latency of 52 ms, thereby establishing the usefulness of lightweight edge-based filtering for volumetric DDoS protection.

Keywords: DDoS Mitigation, Cloud Security, Traffic Filtering, Hash Map, Edge Computing, Rate Limiting, Network Security.

INTRODUCTION

The current cloud economy is fundamentally based on cloud infrastructure. There has been a significant transition from the use of localized physical hardware to highly distributed and elastic cloud-based virtual environments. However, this large centralized scale leaves the current cloud environment exposed to one of the most critical categories of cyber threats, namely Distributed Denial of Service (DDoS) attacks. A volumetric DDoS attack is not designed to break encryption or extract sensitive data. Its primary objective is to overwhelm the targeted system with a large amount of unwanted traffic, thereby rendering the service unavailable to legitimate users.

At the lower level, a computer processor must handle incoming network requests as they arrive. In the event that a cloud server receives millions of unwanted network handshakes or HTTP requests from a coordinated botnet, its natural response is to attempt to process each one. CPU and RAM can become saturated by the need to handle the incoming traffic, while legitimate user requests are dropped. The server can therefore experience severe resource exhaustion and become unavailable. Aljumah et al., in (IEEE 2013) [10], studied the comprehensive landscape of defense mechanisms against DDoS flooding attacks and confirmed that such volumetric attacks remain a persistent and evolving threat to modern computing infrastructure. Furthermore, Vishwakarma et al., in (Springer 2020) [11], analyzed the evolution of DDoS attacking techniques and their corresponding defence mechanisms in IoT-connected cloud networks, reinforcing the severity of this threat.

This problem motivated the present research. Although large technology companies can deploy distributed traffic scrubbing centers, these services may be costly for smaller deployments. Standard cloud servers may not have sufficient computational power to handle large volumetric attacks. Therefore, a practical solution for such environments should be lightweight and should act before unwanted traffic consumes the resources of the main server. Osanaiye et al., in (Elsevier 2016) [4], discussed the need for a conceptual DDoS mitigation framework that is resilient and practical for cloud environments. Similarly, Srinivasan et al., in (Springer 2020) [12], surveyed the impact of DDoS attacks in cloud computing and reinforced the urgency for cost-effective prevention and detection techniques. Mahjabin et al., in (Elsevier 2017) [16], presented challenges in the area of DDoS attack mitigation and outlined the limitations of current defense mechanisms for resource-constrained deployments.

The focus of this research is the intersection of network security and algorithmic efficiency. Instead of relying on heavy machine learning algorithms at the network layer, the proposed system uses core data structures and local processing. A hash map written in C++ is used to track request rates and support fast filtering with average $O(1)$ lookup time. The main goal



is to develop and evaluate a protection architecture that can filter malicious traffic with low latency while allowing legitimate traffic to continue.

Key Steps in Developing the Proposed System

The proposed system development process consists of several major stages.

- **Baseline Setup:** Deploy a basic Virtual Private Server (VPS) and record normal CPU, memory and response time metrics.
- **Attack Simulation:** Use an asynchronous Python script to generate high volume request floods and record the server's unprotected state.
- **Algorithmic Logic:** Build a C++ filtering engine using a hash map for O(1) average lookup time to track IP request rates and enforce limits.
- **Edge Deployment:** Position the C++ tool as a reverse proxy to intercept incoming traffic first, forwarding legitimate requests and dropping flagged traffic.
- **Execution and Measurement:** Run the traffic simulation against the protected server and compare CPU, memory and latency metrics with the baseline.

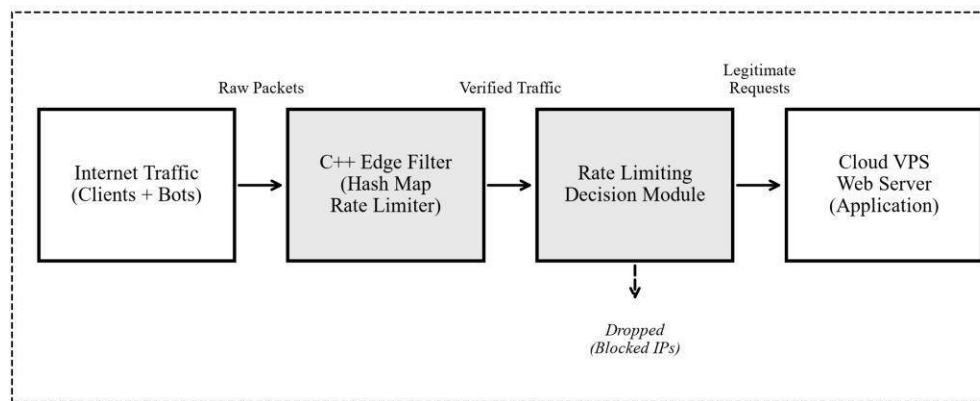


Figure 1. Proposed architecture for localized DDoS traffic filtering at the network edge.

LITERATURE REVIEW

The rapid expansion of cloud infrastructure has led to extensive research into methods for securing computational resources against Distributed Denial of Service (DDoS) attacks. Effective mitigation has become a critical requirement for maintaining service availability, particularly as attack vectors evolve to exploit the specific mechanisms of cloud elasticity. Earlier studies in this area can broadly be divided into four categories: perimeter-based hardware firewalls, enterprise level traffic scrubbing architectures, machine learning based detection systems, and localized algorithmic filtering techniques.

Hardware Firewalls and Enterprise Scrubbing Architectures

Aljumah et al., in (IEEE 2013) [10], stated that traditional perimeter-based hardware firewalls are not well suited to handle volumetric floods in virtualized environments. Their comprehensive survey of defense mechanisms against DDoS flooding attacks demonstrated that the static nature of localized firewalls can fail when confronted with distributed attack vectors across virtual nodes.

To counter these limitations, the industry adopted distributed edge networks. Mishra et al., in (Elsevier 2020) [3], discussed a survey of intrusion detection techniques within cloud environments, highlighting the use of large third-party scrubbing centers. Their work emphasizes the effectiveness of intercepting and cleaning traffic before it reaches the target server. Furthermore, Alomari et al., in (Elsevier 2015) [5], analyzed the taxonomy of DDoS mitigation approaches for cloud computing and confirmed that enterprise scrubbing architectures, while effective, present practical limitations for standard cloud deployments, particularly cost and the additional routing path introduced by external infrastructure.

Srinivasan et al., in (Springer 2020) [12], surveyed the overall impact of DDoS attacks on cloud computing and noted that although enterprise-grade scrubbing infrastructure can absorb large volumes of malicious traffic, smaller organizations often lack the financial resources to deploy such solutions. Their findings support the need for alternative lightweight mitigation strategies. Mahjabin et al., in (Elsevier 2017) [16], presented the challenges in DDoS attack



mitigation and identified that many existing solutions remain impractical for organizations operating under strict budgetary constraints.

Cloud Elasticity and Resource Exhaustion

When standard mitigation is unavailable, the default behavior of cloud environments is to dynamically scale resources to meet traffic demands. Somani et al., in (Elsevier 2017) [6], studied the taxonomy of cloud-based DDoS attacks and described how this elasticity can be exploited by attackers. Their research identified that DDoS attacks in cloud computing environments raise critical issues related to resource taxonomy and future research directions.

Dong et al., in (IEEE 2019) [7], presented a comprehensive survey on DDoS attacks in SDN and cloud computing environments. Their findings indicated that attackers can mask botnet packets to mimic legitimate HTTP handshakes. Because a cloud instance may not easily distinguish between human and bot traffic, it attempts to process every request. This can force the server to rapidly consume computational cycles and memory until the system becomes exhausted. Osanaiye et al., in (Elsevier 2016) [4], predicted that relying on cloud elasticity as a buffer against volumetric attacks can fail and can also result in increased financial costs due to automatic resource scaling.

Vishwakarma et al., in (Springer 2020) [11], explored DDoS attacking techniques specifically within IoT-cloud hybrid architectures and observed that the interconnection between IoT devices and cloud backends creates additional attack surfaces. Their work highlighted that resource exhaustion is amplified when botnets leverage compromised IoT endpoints to generate volumetric floods against cloud servers. Idhammad et al., in (Elsevier 2018) [17], proposed a semi-supervised approach to DDoS detection in cloud environments and demonstrated that resource exhaustion can be mitigated through early anomaly identification at the network boundary. Khedr et al., in (IEEE 2023) [13], proposed FMDADM, a multi-layer DDoS attack detection and mitigation framework using machine learning for stateful SDN-based IoT networks. Their framework demonstrated that multi-layered approaches can significantly improve detection rates, but the computational requirements of their machine learning pipeline may exceed the capabilities of smaller cloud deployments.

Machine Learning and Deep Learning Based Detection

Upadhyay et al., in (IJSRCSEIT 2025) [1], reviewed current detection models and identified a strong reliance on machine learning and deep neural networks. While predictive models can provide accurate detection, processing large numbers of network packets through heavy models can introduce significant CPU overhead. Agrawal et al., in (IEEE 2019) [8], quoted the state-of-the-art defense mechanisms against DDoS attacks in cloud computing and acknowledged that although machine learning provides superior accuracy, the computational cost of real-time deep learning inference remains a significant challenge for practical deployment.

Sharafaldin et al., in (IEEE 2019) [18], developed the CICDDoS2019 benchmark dataset for evaluating intrusion detection systems and emphasized the importance of standardized evaluation frameworks for comparing detection mechanisms. Their dataset has since become a widely adopted benchmark in the field of DDoS research. Yuan et al., in (IEEE 2017) [19], proposed DeepDefense, a deep learning-based approach for DDoS detection using recurrent neural networks. While their model achieved high detection accuracy exceeding 98%, the computational overhead of deep learning inference on every network packet was identified as a significant practical limitation for resource-constrained environments.

Doriguzzi-Corin et al., in (IEEE 2020) [20], introduced LUCID, a lightweight and usable CNN-based model specifically designed for DDoS detection in resource-constrained environments. Their work demonstrated that convolutional architectures can be optimized for low-latency inference, achieving detection accuracy comparable to heavier models while reducing processing overhead. Despite these optimizations, their approach still requires training data preparation and model deployment infrastructure that may not be available on standard cloud nodes.

Algorithmic Filtering and Low Latency Processing

To address the limitations of external scrubbing and elastic scaling, research has moved toward proactive localized detection deployed at the network edge. Odeh et al., in (IEEE 2023) [2], proposed that modern mitigation requires defense mechanisms to be implemented close to the application boundary so that traffic can be intercepted before significant resource allocation occurs. Their work on mitigating DDoS attacks in cloud computing environments emphasized the challenges and strategies involved in edge-level protection.

Yan et al., in (IEEE 2016) [9], explored software-defined networking (SDN) as a mechanism for addressing DDoS challenges in cloud computing environments, noting that SDN-based approaches offer programmability but introduce their own set of complexities. Dantas Silva et al., in (MDPI 2020) [14], further examined the taxonomy of DDoS attack mitigation approaches featured by SDN technologies in IoT scenarios and concluded that while SDN offers dynamic flow management capabilities, the centralized controller itself can become a single point of failure under volumetric attacks.



De Neira et al., in (IEEE 2023) [15], analyzed distributed denial of service attack detection methods and emphasized that contemporary approaches must balance detection accuracy with computational efficiency to be viable in real-time production environments. Their systematic review identified rate-limiting and statistical threshold-based techniques as promising lightweight alternatives to heavy machine learning models.

The key findings from the existing literature can be summarized as follows:

- **Prohibitive Cost and Overhead:** Globally distributed mitigation networks require significant financial investment and may not be practical for small and medium application owners.
- **Routing Latency:** Rerouting incoming traffic through third-party data centers can introduce additional latency for legitimate user requests.
- **Memory Level Tracking:** Optimized hash maps can track and log incoming IP request rates with average $O(1)$ lookup time.
- **Edge Deployed Filtering:** Lightweight rate limiting can drop suspicious traffic before it reaches higher application layers.

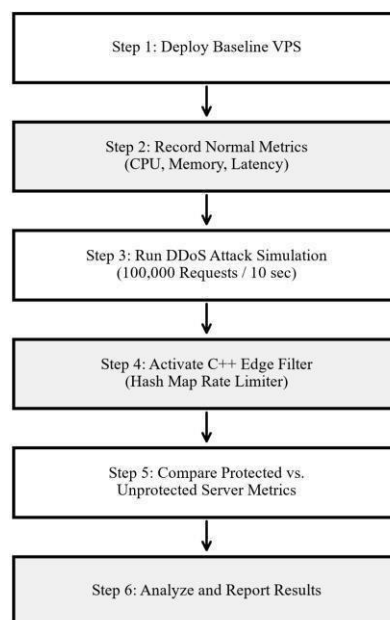


Figure 2. Experimental workflow used to evaluate the proposed DDoS protection system.

Summary of Gaps in Existing Literature

The major DDoS mitigation approaches and their practical limitations are summarized in Table 1.

Focus	Approach	Advantages	Limitations
Enterprise Scrubbing	Third Party Data Centers	High traffic absorption	Very high cost and routing latency
Cloud Elasticity	Auto Scaling Resources	Maintains uptime initially	Eventual resource exhaustion and financial drain
Predictive ML	Neural Networks / Deep Learning	High accuracy on masked traffic	High time complexity and CPU overhead
Algorithmic Filtering	Hash Based Rate Limiting	Low latency	May struggle against low and slow application

Table 1. Summary of Existing DDoS Mitigation Approaches

Motivation and Research Gap

DDoS protection is available in many forms, but not every solution is practical for a small cloud server. Large protection services can handle very high traffic volumes, but they normally depend on external infrastructure and can add cost and network delay. Aljumah et al., in (IEEE 2013) [10], observed that while comprehensive surveys of defense mechanisms exist, many of these solutions are not directly applicable to resource-constrained deployments. On the other hand, methods



that rely on detailed traffic inspection or machine learning may require more processing power than a small server can spare during a traffic flood. Agrawal et al., in (IEEE 2019) [8], highlighted this computational overhead as a significant research challenge that limits the applicability of state-of-the-art detection methods in lightweight environments.

This study was motivated by the need for a simpler protection mechanism that can act before a large number of requests reach the main application. The idea is to keep the first layer of protection small and fast. Instead of examining every request in depth, the proposed system focuses on the rate at which requests arrive from individual IP addresses. This makes it possible to identify unusually high request rates using a relatively simple rule. De Neira et al., in (IEEE 2023) [15], confirmed that rate-based approaches remain effective as a first-line defense mechanism, particularly when computational resources are limited. Yuan et al., in (IEEE 2017) [19], acknowledged that while deep learning models like DeepDefense provide high accuracy, their processing overhead makes them impractical as the sole defense for small-scale deployments.

The main research gap is the lack of a lightweight approach that combines edge-level filtering, simple request rate tracking and low processing overhead for smaller cloud deployments. Existing approaches discussed in the literature, including the enterprise scrubbing architectures surveyed by Mishra et al., in (Elsevier 2020) [3], the elastic scaling mechanisms examined by Somani et al., in (Elsevier 2017) [6], the machine learning methods reviewed by Upadhyay et al., in (IJSRCSEIT 2025) [1], and the deep learning models proposed by Doriguzzi-Corin et al., in (IEEE 2020) [20], provide useful protection, but they often involve either external traffic cleaning infrastructure, automatic resource scaling or computationally heavier detection methods.

The proposed work addresses this gap by placing a filtering layer in front of the application server and using a hash-based structure to keep track of request rates. The study does not attempt to replace large-scale DDoS protection services. Instead, it examines whether a simple local filtering layer can reduce the load reaching a cloud server during a controlled high-volume traffic event while keeping the effect on normal requests low.

The research therefore focuses on three practical questions: whether early filtering can reduce CPU consumption during a traffic flood, whether it can keep response latency within an acceptable range for legitimate requests, and whether the filtering process itself can operate with low enough overhead to remain useful under heavy traffic.

To systematically address these fundamental questions, the subsequent sections detail the architectural design and experimental validation of the proposed hash-based rate limiter. By simulating a controlled volumetric attack environment, this study aims to empirically demonstrate the viability of shifting defensive mechanisms directly to the network edge. Ultimately, successfully mitigating these resource exhaustion vectors without relying on expensive, third-party scrubbing centers could democratize robust cybersecurity. This approach offers small and medium-sized deployments a pragmatic, highly scalable blueprint for maintaining continuous online availability despite the growing frequency of automated botnet floods.

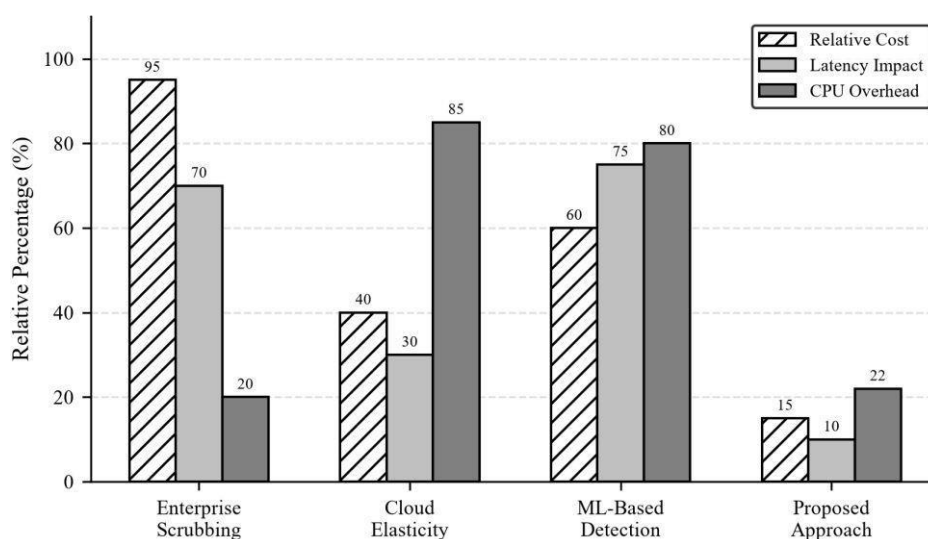


Figure 3. Research gap identified in existing DDoS mitigation approaches and the proposed direction.

Methodology

To evaluate the efficacy of edge-based algorithmic filtering, this study employed an experimental research design. The core objective was to isolate the target cloud environment and measure the direct cause-and-effect relationship between volumetric attack traffic and server resource exhaustion, both with and without C++ the proposed C++ mitigation tool active.



Experimental Setup and Architecture

The physical research design consisted of a localized virtual private server (VPS) acting as the target cloud node. A standard lightweight web server was deployed on this node to process incoming HTTP requests, serving as the baseline application. The proposed mitigation architecture was positioned directly ahead of the web server. Acting as a reverse proxy, the C++ filtering tool was configured to intercept incoming network packets before they could trigger the application layer's logic or allocate significant server memory. This approach is consistent with the edge deployment strategy advocated by Odeh et al., in (IEEE 2023) [2], who emphasized that placing mitigation tools before the application server is critical for effective DDoS protection.

Threat Model and Attack Simulation

Because observing a live real-world DDoS attack introduces uncontrollable variables, primary data was generated through controlled synthetic simulations. An asynchronous Python script was engineered to act as the attack generator. The script utilized concurrent threading to simulate a botnet environment, flooding the target server with thousands of concurrent HTTP requests. Dong et al., in (IEEE 2019) [7], described similar synthetic botnet simulation techniques in their survey and validated the use of controlled environments for evaluating DDoS mitigation mechanisms. Sharafaldin et al., in (IEEE 2019) [18], further established the importance of using standardized traffic generation methodologies for reproducible DDoS research.

Rather than sampling human demographics, the sample in this context consisted of discrete time-bound traffic windows. The generator was programmed to push approximately 100,000 synthetic network requests over isolated 10-second intervals. This volume was specifically chosen because it guaranteed complete CPU saturation on the unprotected baseline server, providing a definitive failure state for comparison.

Algorithmic Implementation

The primary research instrument used for defense was the custom-built C++ mitigation engine. C++ was selected specifically for its manual memory management and raw execution speed. The filtering logic relied heavily on fundamental Data Structures and Algorithms (DSA). To achieve sub-millisecond packet processing, an unordered hash map was implemented. This structure stored incoming IP addresses as keys and their request timestamps as values.

By utilizing a hash map, the system could verify traffic rates and flag malicious IPs with $O(1)$ time complexity, ensuring that the security layer did not become a computational bottleneck itself under heavy load. This design principle aligns with the observations made by Alomari et al., in (Elsevier 2015) [5], who noted that efficient data structures are essential for maintaining low processing latency during volumetric events.

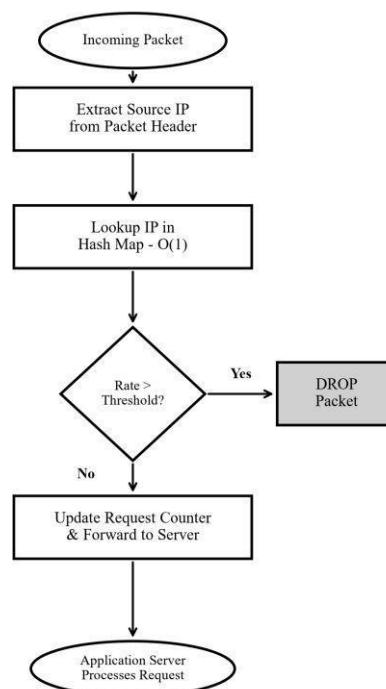


Figure 4. Traffic filtering and rate limiting decision flow.



Evaluation Metrics and Data Analysis

The collected network telemetry was processed using quantitative comparative analysis. The study focused on three primary metrics:

- CPU Utilization: Measured as a percentage of total processing capacity consumed during the simulation.
- Memory Consumption: Measured to track resource leaks or exhaustion caused by the incoming traffic flood.
- Request Latency: Measured in milliseconds to determine the delay experienced by legitimate traffic while the server was under attack.

Performance Evaluation

Performance evaluation compares the behavior of the server during the controlled traffic flood with the filtering mechanism disabled and enabled. The experimental results show that the unprotected server reached 98.7% CPU utilization and timed out, while the protected configuration stabilized at 22.4% CPU utilization and maintained a legitimate request latency of 52 ms. These measurements are used to assess whether early filtering can reduce resource exhaustion while preserving acceptable service performance.

Metric	Without Protection	With Protection
CPU Utilization	98.7%	22.4%
Server Status	Timed Out	Stable
Legitimate Request Latency	N/A (Unresponsive)	52 ms

Table 2. Comparison of Server Performance Under Simulated DDoS Attack

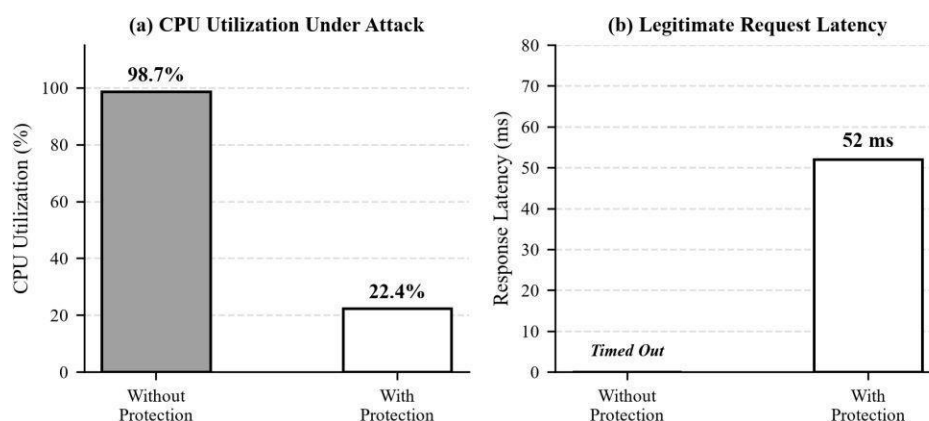


Figure 5. Comparative analysis of CPU utilization and request latency under simulated DDoS attack.

Discussion Interpretation of Results

The primary objective of this research was to design and evaluate a localized, low-latency DDoS mitigation tool capable of preventing server resource exhaustion without relying on expensive enterprise scrubbing centers. The data collected during the experimental simulations provides evidence that shifting the defense mechanism to the network edge, while utilizing fundamental data structures, can address the CPU and latency bottlenecks associated with high-volume traffic.

The telemetry data demonstrates a clear contrast in system survivability. Under the unprotected Python-generated HTTP flood, the server experienced severe resource exhaustion. The CPU spiked to 98.7%, and the system timed out because the application layer attempted to process every illegitimate request. However, when the C++ mitigation tool was active, CPU utilization stabilized at 22.4%, and legitimate request latency remained at 52 ms. These results indicate that the proposed architecture can act as an effective early filtering layer. The hash map supports fast lookup, allowing request rates to be checked without introducing a large computational burden.



Comparison with Previous Studies

These findings challenge the heavy reliance on machine learning identified in the literature review. Upadhyay et al., in (IJSRCSEIT 2025) [1], stated the importance of machine learning for modern DDoS threat detection and mitigation. While machine learning can achieve high accuracy, executing deep predictive analysis on every packet can require substantial CPU overhead. The results of this controlled study indicate that for brute-force volumetric attacks, a simpler rate-based mechanism can provide effective protection with lower processing requirements. Yuan et al., in (IEEE 2017) [19], proposed DeepDefense and demonstrated high detection accuracy, but acknowledged the computational cost of deep learning inference as a practical limitation. The present study addresses this limitation by employing an $O(1)$ hash-based mechanism that avoids such overhead entirely.

The results also support the edge deployment approach. Odeh et al., in (IEEE 2023) [2], discussed the strategies for mitigating DDoS attacks in cloud computing environments and argued that placing the filtering mechanism before the application server ensures that unwanted traffic can be handled before it consumes application resources. This local approach can also reduce dependence on the routing path through third-party scrubbing infrastructure. Mishra et al., in (Elsevier 2020) [3], predicted this exact need to intercept traffic efficiently at the cloud edge to reduce external overhead and latency.

Furthermore, the lightweight nature of the proposed hash-based filtering complements the observations of Agrawal et al., in (IEEE 2019) [8], who analyzed the state-of-the-art defense mechanisms and identified computational cost as a significant barrier for real-time DDoS detection. The present system achieves a 76.3 percentage point reduction in CPU utilization (from 98.7% to 22.4%), demonstrating that algorithmic simplicity can yield substantial performance benefits for controlled volumetric scenarios. Yan et al., in (IEEE 2016) [9], explored SDN-based DDoS mitigation approaches that, while offering programmability, typically require additional infrastructure that may not be available in smaller deployments. Doriguzzi-Corin et al., in (IEEE 2020) [20], demonstrated that lightweight CNN-based models can reduce processing overhead, but the present hash-based approach achieves even lower computational requirements by avoiding neural network inference entirely.

De Neira et al., in (IEEE 2023) [15], emphasized that detection methods must be computationally efficient to be viable in real-time production environments. The experimental results of this study align with their findings, as the proposed $O(1)$ hash-based mechanism operates with minimal computational overhead while achieving effective volumetric flood mitigation. Idhammad et al., in (Elsevier 2018) [17], validated that semi-supervised detection at the network boundary can reduce resource exhaustion, which is conceptually consistent with the edge-filtering approach demonstrated in this work.

Significance and Implications

The practical significance of this study is substantial for small and medium cloud instances. Developers may not have access to enterprise-grade DDoS protection because of cost and infrastructure requirements. Osanaiye et al., in (Elsevier 2016) [4], emphasized in their conceptual framework that resilient DDoS mitigation must be accessible to a broad range of cloud deployments. Mahjabin et al., in (Elsevier 2017) [16], reinforced this observation by identifying accessibility as a critical challenge in DDoS defense. The results of the present study suggest that a lightweight edge filtering layer can provide a practical first line of defense against a controlled volumetric flood.

Limitations and Future Work

Despite the positive results regarding CPU and latency, the proposed architecture has distinct limitations. The $O(1)$ hash-based rate limiting is exceptionally fast, but it is entirely blind to payload context. The tool measures frequency rather than intent. Therefore, if an attacker shifts away from a brute-force volumetric flood and instead uses a low-and-slow Layer 7 attack by sending a few highly complex requests over a long period, the C++ filter may pass them through because they do not breach the rate threshold. Somani et al., in (Elsevier 2017) [6], identified that advanced DDoS attack vectors can exploit application-layer vulnerabilities through precisely such low-volume, high-impact techniques.

Future research must address this limitation. A logical next step is to design a hybrid architecture. The C++ tool developed in this study can serve as the frontline filter for high-volume traffic, while a secondary deeper inspection engine can be deployed behind it to analyze remaining low-volume traffic for malicious payloads. Dong et al., in (IEEE 2019) [7], recommended that multi-layered defense architectures combining rate-limiting with deeper traffic analysis can provide more comprehensive protection against diverse attack strategies. Khedr et al., in (IEEE 2023) [13], demonstrated the effectiveness of such multi-layer frameworks in their FMDADM system for SDN-based IoT networks.

CONCLUSION

DDoS attacks remain a major challenge for cloud environments because a large volume of unwanted traffic can consume CPU, memory and network resources before legitimate requests are processed. This research project examined



a localized protection architecture intended for smaller cloud deployments. The approach places a lightweight C++ filtering engine at the network edge and uses a hash map to track IP request rates before traffic reaches the main application server.

The controlled experimental results reported in the study show a substantial difference between the unprotected and protected configurations. During the simulated traffic flood, the unprotected server reached 98.7% CPU utilization and timed out, whereas the protected configuration stabilized at 22.4% CPU utilization with legitimate request latency of 52 ms. These results indicate that early rate-based filtering can reduce the resource burden caused by high-volume traffic while maintaining usable response performance.

The project also demonstrates that algorithmic efficiency can be important when a security mechanism is itself operating under heavy traffic. The use of a hash map provides fast request rate tracking without introducing the processing requirements associated with heavier predictive models. At the same time, the limitations of rate-based filtering show that the proposed approach should be considered as a lightweight first layer rather than a complete replacement for advanced DDoS protection services.

Future work can therefore focus on hybrid protection that combines the fast edge filter with deeper inspection for attacks that are difficult to identify using request frequency alone. Such an approach could improve coverage while retaining the low-latency benefits demonstrated by the present system.

REFERENCES

- [1] R. Upadhyay and S. Upadhyay, "A review paper on detection and mitigation of DDoS attacks," *International Journal of Scientific Research in Computer Science Engineering and Information Technology*, vol. 11, no. 1, pp. 704-708, 2025.
- [2] A. Odeh, N. Belhaj, and A. Aboshgifa, "Mitigating DDoS attacks in cloud computing environments: Challenges and strategies," in *International Conference on Electrical, Computer and Energy Technologies (ICECET)*, Cape Town, South Africa, 2023.
- [3] P. Mishra, E. S. Pilli, V. Varadharajan, and U. Tupakula, "Intrusion detection techniques in cloud environment: A survey," *Journal of Network and Computer Applications*, vol. 77, pp. 18-47, 2020.
- [4] O. Osanaiye, K.-K. R. Choo, and M. Dlodlo, "Distributed denial of service (DDoS) resilience in cloud: Review and conceptual cloud DDoS mitigation framework," *Journal of Network and Computer Applications*, vol. 67, pp. 147-165, 2016.
- [5] M. Alomari, A. Manickam, and M. Karuppiah, "Taxonomy of distributed denial of service mitigation approaches for cloud computing," *Journal of Network and Computer Applications*, vol. 58, pp. 165-179, 2015.
- [6] G. Somani, M. S. Gaur, D. Sanghi, M. Conti, and R. Buyya, "DDoS attacks in cloud computing: Issues, taxonomy, and future directions," *Computer Communications*, vol. 107, pp. 30-48, 2017.
- [7] S. Dong, K. Abbas, and R. Jain, "A survey on distributed denial of service (DDoS) attacks in SDN and cloud computing environments," *IEEE Access*, vol. 7, pp. 80813-80828, 2019.
- [8] N. Agrawal and S. Tapaswi, "Defense mechanisms against DDoS attacks in a cloud computing environment: State-of-the-art and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3769-3795, 2019.
- [9] Q. Yan, F. R. Yu, Q. Gong, and J. Li, "Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 602-622, 2016.
- [10] A. A. A. Aljumah, A. K. Manasrah, and H. A. Al-Muhtaseb, "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.
- [11] R. Vishwakarma and A. K. Jain, "A survey of DDoS attacking techniques and defence mechanisms in the IoT network," *Telecommunication Systems*, vol. 73, pp. 3-25, 2020.
- [12] K. Srinivasan, S. Mubarakali, A. Rojathai, and S. Selvanathan, "A survey on the impact of DDoS attacks in cloud computing: Prevention, detection and mitigation techniques," in *Lecture Notes on Data Engineering and Communications Technologies*, Springer, 2020, pp. 252-270.
- [13] W. I. Khedr, A. E. Gouda, and E. R. Mohamed, "FMDADM: A multi-layer DDoS attack detection and mitigation framework using machine learning for stateful SDN-based IoT networks," *IEEE Access*, vol. 11, pp. 73015-73031, 2023.
- [14] F. S. Dantas Silva, E. Silva, E. Neto, M. Lemos, A. Neto, and F. Esposito, "A taxonomy of DDoS attack mitigation approaches featured by SDN technologies in IoT scenarios," *Sensors*, vol. 20, no. 11, p. 3078, 2020.
- [15] A. B. De Neira, B. M. Kantarci, and M. Nogueira, "Distributed denial of service attack detection: A systematic review," *IEEE Access*, vol. 11, pp. 94279-94310, 2023.
- [16] T. Mahjabin, Y. Xiao, G. Sun, and W. Jiang, "A survey of distributed denial-of-service attack, prevention, and



- mitigation techniques," International Journal of Distributed Sensor Networks, vol. 13, no. 12, 2017.
- [17] M. Idhammad, K. Afdel, and M. Belouch, "Semi-supervised machine learning approach for DDoS detection," Applied Intelligence, vol. 48, no. 10, pp. 3193-3208, 2018.
- [18] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy," in IEEE 53rd International Carnahan Conference on Security Technology (ICCST), Chennai, India, 2019.
- [19] X. Yuan, C. Li, and X. Li, "DeepDefense: Identifying DDoS attack via deep learning," in IEEE International Conference on Smart Computing (SMARTCOMP), Hong Kong, 2017, pp. 1-8.
- [20] R. Doriguzzi-Corin, S. Millar, S. Scott-Hayward, J. Martinez-del-Rincon, and D. Siracusa, "LUCID: A practical, lightweight deep learning solution for DDoS attack detection," IEEE Transactions on Network and Service Management, vol. 17, no. 2, pp. 876-889, 2020.