Simulation Analysis of SYN Flood and HTTP Flood Attacks on Cloud Infrastructure Integrity

Ng Yen Phing, Caleb Chong Senn Yang, Low Choon Keat, Phoon Gar Chi

Department of Information and Communication Technology, Tunku Abdul Rahman University of Management and Technology, Malaysia

DOI: https://dx.doi.org/10.47772/IJRISS.2025.910000680

Received: 28 October 2025; Accepted: 03 November 2025; Published: 20 November 2025

ABSTRACT

This paper presents a comparative simulation study of SYN Flood and HTTP Flood Distributed Denial-of-Service (DDoS) attacks in cloud environments using CloudSim. A modular testbed was configured with attacker VMs generating cloudlets and victim VMs handling legitimate workloads, under realistic network constraints. Experimental results revealed distinct attack signatures: SYN Flood produced high volumes of half-open connections, while HTTP Flood exhausted CPU, memory, and bandwidth due to resource-intensive request processing. SYN Flood achieved a 35% packet loss rate with 10,000 cloudlets, while HTTP Flood produced a 40% loss rate with only 3,000 requests, demonstrating that application-layer attacks, though lower in volume, cause more severe degradation. These findings highlight the importance of nuanced defense strategies tailored to each attack type, beyond volumetric thresholds alone.

Keywords—CloudSim; DDoS; SYN Flood; HTTP Flood; Simulation

INTRODUCTION

Cloud computing offers elasticity, scalability, and multi-tenancy but remains highly vulnerable to Distributed Denial-of-Service (DDoS) attacks. Among these, SYN Floods overwhelm TCP state tables at the protocol layer, while HTTP Floods mimic legitimate user traffic at the application layer, making them difficult to detect. Current research lacks standardized simulation frameworks for systematically comparing the effectiveness of different DDoS attack types on cloud infrastructure, particularly in quantifying resource consumption patterns and service degradation metrics. This study addresses this gap by employing CloudSim as a reproducible simulation environment to quantify the differential effects of SYN Flood and HTTP Flood attacks on virtualized datacenter resources, with emphasis on packet loss, downtime, and bandwidth consumption. Our primary contributions include: (1) development of a modular CloudSim-based framework with configurable attack generators and metrics collectors for systematic DDoS evaluation; (2) empirical demonstration that HTTP Floods achieve higher loss rates (40.0%) despite using fewer cloudlets compared to SYN Floods (35.0% with 10,000 cloudlets); and (3) quantitative analysis revealing HTTP Floods consume 93× more bandwidth resources (75.67 Mbps vs 0.81 Mbps), providing critical insights for cloud security resource allocation and defense mechanism design.

LITERATURE REVIEW

Distributed Denial-of-Service attacks have evolved significantly since their emergence in the late 1990s, with cloud computing environments presenting unique challenges and vulnerabilities [1]. The classification of DDoS attacks into volumetric, protocol, and application-layer categories provides a framework for understanding their distinct impact mechanisms [2].

D DoS Attack Classification and Evolution

Modern DDoS attacks are categorized into three primary types: volumetric attacks that saturate network bandwidth, protocol attacks that exploit weaknesses in network protocols, and application-layer attacks that target specific services [3]. SYN Flood attacks exemplify protocol-level exploitation by overwhelming TCP connection state tables through the creation of numerous half-open connections, effectively exhausting server resources before legitimate connections can be established [4].

HTTP Flood attacks represent a more sophisticated application-layer threat that mimics legitimate user behavior by generating seemingly valid HTTP requests to web servers [5]. These attacks are particularly challenging to detect because they operate within normal protocol parameters while consuming disproportionate server resources through complex request processing [6].

B. Cloud-Specific Vulnerabilities

Cloud environments introduce unique attack vectors that traditional DDoS research has not adequately addressed. Economic Denial-of-Sustainability (EDoS) attacks exploit auto-scaling mechanisms to inflate operational costs, often triggering resource allocation before volumetric defenses recognize anomalous traffic patterns [7]. The multi-tenancy characteristic of cloud computing compounds these risks through collateral damage effects, where attacks against one tenant can degrade performance for neighboring workloads sharing physical infrastructure resources [8].

C. Simulation-Based Research Methodologies

CloudSim has emerged as the preferred framework for DDoS research due to its extensibility and reproducibility in modeling cloud environments. Karthik and Shah [9] demonstrated early applications of CloudSim for modeling flooding attacks on virtual machines, establishing baseline methodologies for attack simulation in controlled environments. Their work revealed significant service disruption through simulated flooding scenarios but lacked comparative analysis between different attack types.

Ali et al. [10] advanced this research direction by implementing multi-layer detection frameworks incorporating fuzzy logic and machine learning classifiers, achieving detection accuracy rates of 98-99% in simulated environments. Their approach used 10 VMs for legitimate traffic generation and 40 cloudlet tasks representing malicious attackers, providing a foundation for understanding attack-defense dynamics in cloud simulations.

Recent developments include reinforcement learning-based defense mechanisms, with Basulaim and AlAmoudi [11] proposing a three-stage system combining multi-factor authentication, fuzzy-VIKOR detection algorithms, and VM isolation strategies. Their CloudSim-Plus implementation demonstrated improved Quality of Service metrics under HTTP flood conditions, though comparative analysis with protocol-layer attacks remained limited.

Sreeram and Vuppala [12] contributed to HTTP flood detection research by applying machine learning metrics and bio-inspired algorithms, achieving high accuracy in identifying application-layer attacks. However, their work focused primarily on detection mechanisms rather than quantifying resource consumption patterns across different attack types.

D. Research Gap Identification

Despite significant advances in individual attack modeling and detection mechanisms, the literature reveals insufficient comparative analysis of protocol-layer versus application-layer attacks in cloud environments. Existing studies predominantly focus on detection and mitigation strategies rather than establishing standardized metrics for quantifying attack effectiveness and resource impact patterns. This gap limits the development of optimal defense resource allocation strategies and hinders the systematic evaluation of cloud infrastructure resilience against diverse DDoS attack vectors.

METHODOLOGY

A. Simulation Environment Configuration

The experimental framework was implemented using CloudSim 3.x and CloudSim 5.0 (main library) to leverage both stable functionality and enhanced network modeling capabilities. The simulation environment comprised two distinct logical datacenters to ensure clear separation between attack generation and victim infrastructure:

Attacker Datacenter: Hosted multiple VMs configured to generate malicious cloudlet streams representing SYN Flood and HTTP Flood attack patterns. Each attacker VM was programmed with specific attack characteristics including cloudlet generation rates, payload sizes, and connection duration parameters.

Victim Datacenter: Contained the target web-server VM alongside legitimate workload VMs to simulate realistic cloud service scenarios. The victim infrastructure was configured with resource constraints representative of typical IaaS deployments.

B. Network Parameter Configuration

Critical network parameters were standardized across all simulation runs to ensure reproducible results:

1. Victim datacenter bandwidth: 10 Mbps (reflecting typical small-to-medium cloud service allocations)
2. Network latency: 1 ms (representing optimized datacenter network conditions)
3. Packet transmission protocols: TCP for SYN Flood simulations, HTTP/TCP for application-layer attacks

C. Cloudlet Design and Traffic Modeling

Cloudlets served as abstractions for network requests and application transactions, with distinct configurations for each attack type:

SYN Flood Cloudlets: Designed to represent TCP connection attempts with minimal processing requirements but high-volume characteristics. Each cloudlet consumed minimal CPU cycles while maintaining connection state information.

HTTP Flood Cloudlets: Configured to simulate resource-intensive web requests requiring substantial CPU, memory, and I/O processing. These cloudlets incorporated realistic web transaction characteristics including file transfers and database queries.

Legitimate Traffic Cloudlets: Generated baseline workload patterns to establish performance benchmarks and simulate realistic mixed-traffic scenarios during attack periods.

D. Metrics Collection Framework

A custom metrics collection module was developed to capture comprehensive performance data throughout simulation execution:

Real-time Event Logging: Captured cloudlet lifecycle events including creation, scheduling, execution start, completion, and failure timestamps.

Resource Utilization Monitoring: Tracked CPU, memory, bandwidth, and storage utilization across all VMs at predetermined intervals.

Drop Event Detection: Identified and categorized cloudlet failures based on resource exhaustion, timeout conditions, or explicit rejection by the CloudSim broker.

E. Experimental Design and Parameters

The experimental methodology employed controlled parameter variation to assess attack effectiveness across different scenarios:

Attack Intensity Scaling: SYN Flood simulations varied from 1,000 to 10,000 cloudlets, while HTTP Flood tests ranged from 500 to 3,000 cloudlets to reflect realistic attack volume differences.

Resource Allocation Testing: VM specifications were systematically varied to determine optimal configurations for different attack types and to assess scalability limitations.

Temporal Analysis: Extended simulation runs captured long-term effects of sustained attacks on system stability and recovery patterns.

F. Statistical Validation

Multiple simulation runs were conducted for each parameter configuration to ensure statistical significance of results. Standard deviation calculations and confidence interval analysis validated the reproducibility of observed performance degradation patterns.

RESULTS

A. Attack Attributes

TABLE I

Attribute	SYN Flood	HTTP Flood	Hybrid Attack
Attack Type	Volume-based (TCP)	Application-layer	Multi-vector (TCP + Application)
Cloudlets	10,000	3,000	6,500
Cloudlet Length (MI)	200	100	160
File Size (MB)	1	50	20
Output Size (MB)	1	100	40

Hybrid Attack : 5,000 SYN cloudlets + 1,500 HTTP cloudlets

HTTP Flood requests are fewer in number but more resource-intensive, requiring larger file sizes and outputs.

B. VM Specification

TABLE II

Attribute	SYN Flood	HTTP Flood	Hybrid Attack
VM MIPS	1,000	1,500	1300
VM RAM (MB)	512	1,024	768
VM Bandwidth (Mb/s)	100	200	150

HTTP Flood required significantly higher VM resources to process complex application traffic.

C. Host Specification

TABLE III

Attribute	**SYN Flood**	**HTTP Flood**	**Hybrid Attack**
Host RAM (MB)	2,048	4,096	3072
Total MIPS	2,000	6,000	4000
Bandwidth (Mb/s)	10,000	15,000	12,500

E. Attack Performance Characteristics

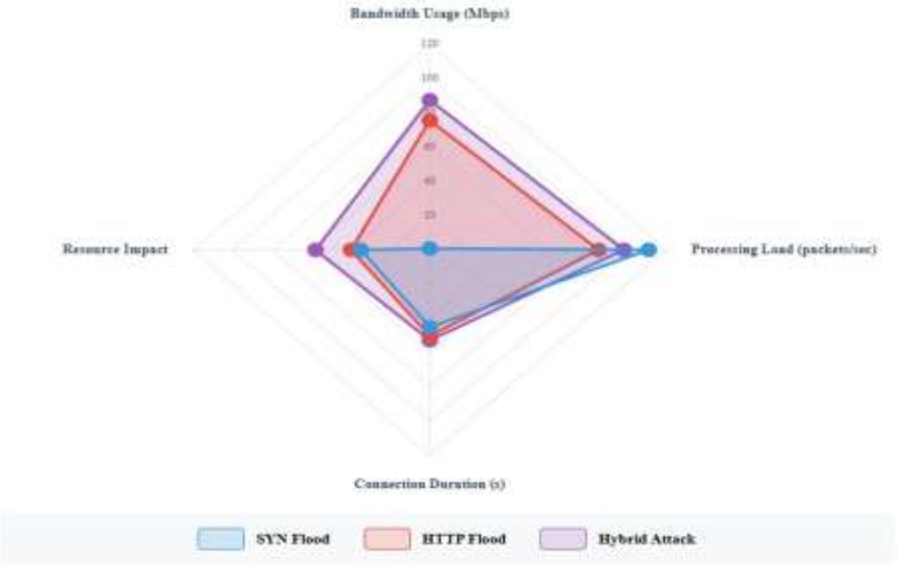


Fig. 1

TABLE IV

Attribute	**SYN Flood**	**HTTP Flood**	**Hybrid Attack**
VM MIPS	1,000	1,500	1300
VM RAM (MB)	512	1,024	768
VM Bandwidth	100	200	150
(Mb/s)			

HTTP Flood required significantly higher VM resources to process complex application traffic.

TABLE V

Attribute	**SYN Flood**	**HTTP Flood**	**Hybrid Attack**
Host RAM (MB)	2,048	4,096	3072

Total MIPS	2,000	6,000	4000
Bandwidth (Mb/s)	10,000	15,000	12,500

C. Attack Performance Characteristics

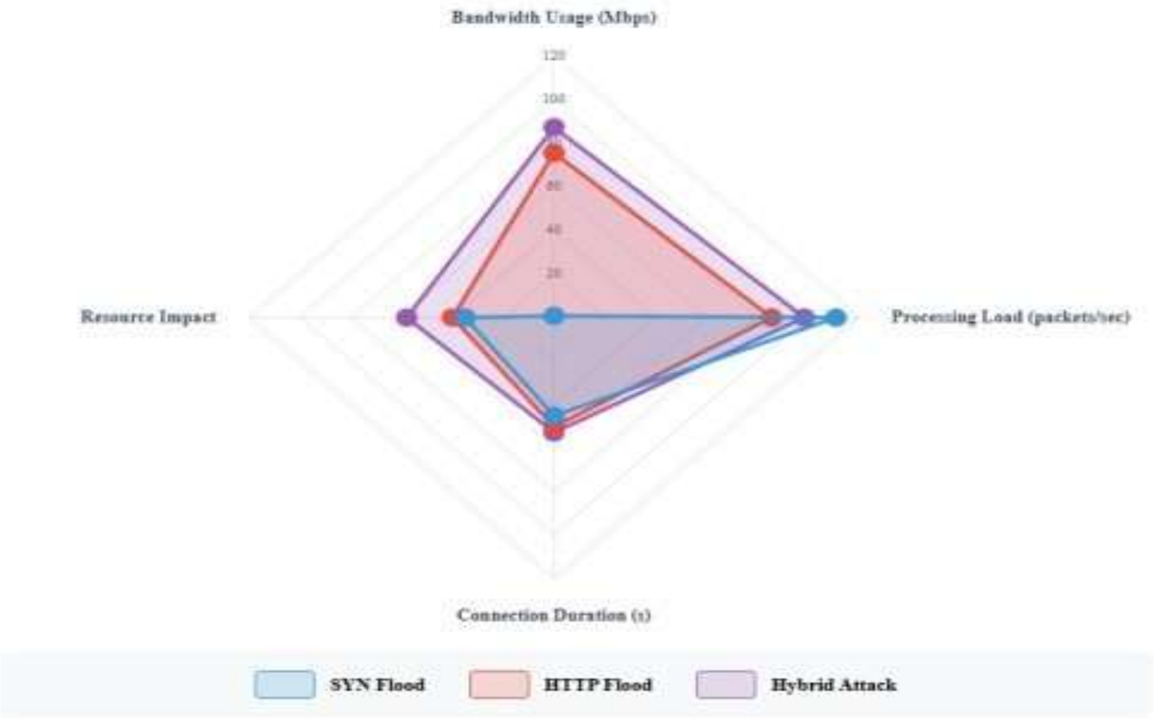


Fig. 1

TABLE VI

Characteristic	**SYN Flood**	**HTTP Flood**	**Hybrid Attack**
Primary Impact	Connection table exhaustion	CPU/Memory exhaustion	Multi-layer exhaustion
Bandwidth Usage (Mbps)	0.81	75.67	87.5
Processing Load (pps)	110.99	85.65	98.5
Connection Duration (s)	45.18	50.21	52.8

Observation: Hybrid attacks exceed both individual attack types in bandwidth consumption (87.5 Mbps) and processing load (98.5 pps), demonstrating compound effects that simultaneously exhaust network capacity and computational resources. While HTTP Flood remains the dominant bandwidth consumer (75.67 Mbps), the addition of SYN traffic increases total bandwidth by 15.6% and processing overhead by 15%, creating a multilayer assault that overwhelms defenses optimized for single attack vectors.

F. Loss Rate Analysis

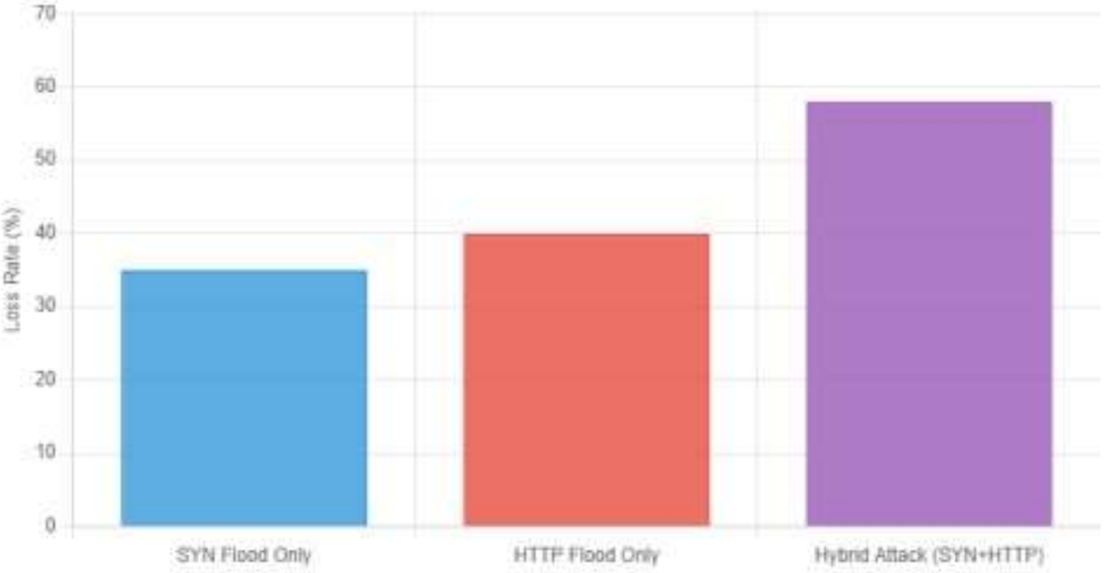


Fig. 2

Formula [13]:

$$L_r(t,s) = \sigma\left(\frac{\sum_{i=1}^{m} C_i^{(t)}}{\sum_{j=1}^{n} C_j^{(t)}}\right) \times 100$$

Lr(t, s) - Loss Rate at time t and scenario s

●Represents the percentage of packets/cloudlets dropped during the simulation **σ** - Summation operator

●Indicates we're summing values across multiple instances

Ci(t) - Individual Cloudlet at time t and index i

●Represents a single computational task/packet in CloudSim

●Each cloudlet has properties like length, file size, and processing requirements **n** - Total number of cloudlets

●Upper limit of the summation (total cloudlets submitted) **m** - Number of dropped cloudlets

●Upper limit for dropped cloudlet summation

Where:

●**Dropped Cloudlets**: Cloudlets rejected due to resource exhaustion or network capacity limitations

●**Total Submitted Cloudlets**: All cloudlets submitted by both legitimate and attack traffic generators

The methodology distinguishes between different drop mechanisms (resource exhaustion vs. network congestion) to provide detailed analysis of failure modes.

TABLE V

Attack	**Total Cloudlets**	**Successful**	**Dropped**	**Loss Rate**
SYN Flood	10,000	6,500	3,500	35.0%

HTTP Flood	3,000	1,800	1,200	40.0%
Hybrid Attack	6,500	2730	3770	58.0%
(SYN+HTTP)				

Observation: Despite lower volume, HTTP Flood exhibited the highest loss rate (40%) among individual attacks, underscoring its potency compared to protocol-layer SYN Floods (35%). This 5-percentage-point difference demonstrates that application-layer attacks achieve greater service degradation through resourceintensive processing rather than sheer request volume—HTTP Flood used only 3,000 cloudlets compared to SYN Flood's 10,000, yet produced superior attack effectiveness. Most significantly, Hybrid Attack accumulated the highest loss rate (58%), representing an 18-percentage-point increase over HTTP Flood and a 23-percentage-point increase over SYN Flood. This compound amplification effect—achieving 58% loss with only 6,500 total cloudlets (65% of SYN's volume)—validates that multi-vector attacks simultaneously exhaust both connection tables and application resources, creating synergistic degradation that exceeds the sum of individual attack impacts. The Hybrid attack's combined protocol properties overwhelm VM processing capacity by forcing simultaneous handling of high-volume connection attempts and resource-intensive HTTP transactions, demonstrating that sophisticated attackers employing coordinated multi-layer strategies pose significantly greater threats than single-vector assaults.

G. Resource Consumption Analysis

TABLE VII (CPU Utilization Over Time)

Time (s)	**No Attack (%)**	**SYN Flood (%)**	**HTTP Flood (%)**
0	15	15	15
10	18	35	42
20	16	58	68
30	20	72	85
40	17	85	94
50	19	89	98
60	18	92	99
70	16	90	99
80	17	88	98
90	18	87	97

Observation: HTTP Flood attacks saturated CPU resources to 99% within 60 seconds, maintaining nearmaximum utilization throughout the attack duration. In contrast, SYN Flood peaked at 92% before stabilizing at 87%, while baseline operations remained consistently below 20%. This demonstrates HTTP Flood's superior capability to exhaust computational resources through application-layer processing demands.

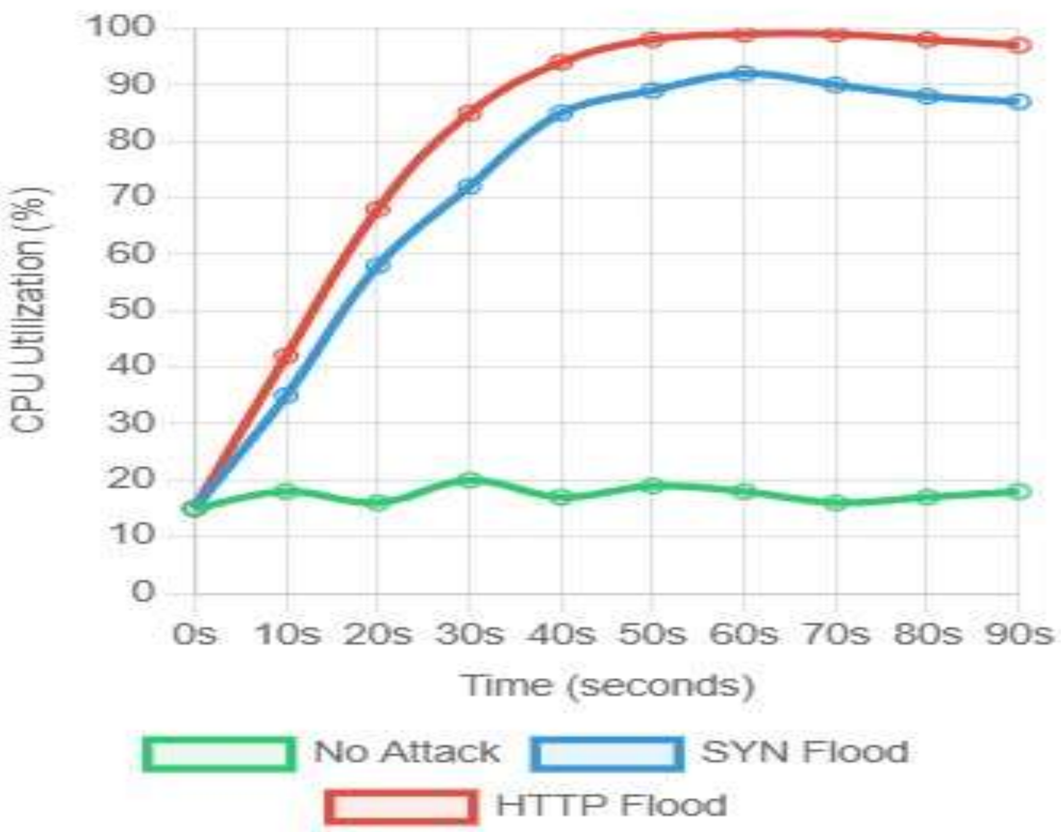


Fig. 3

TABLE VIII (Bandwidth Usage Over Time)

Time (s)	No Attack	SYN Flood	HTTP Flood
0	2.1	2.1	2.1
10	2.3	0.8	35.0
20	2.0	0.85	58.0
30	2.5	0.82	68.0
40	2.2	0.79	72.0
50	2.4	0.81	75.0
60	2.3	0.83	76.0
70	2.1	0.80	75.0
80	2.2	0.81	74.0
90	2.3	0.82	75.67

Observation: The stark contrast in bandwidth consumption patterns confirms the resource-intensity difference between attack types. HTTP Flood consumed 75.67 Mbps (93× more than SYN Flood's 0.81 Mbps), saturating network capacity and blocking legitimate traffic flows. SYN Flood's minimal bandwidth requirement (below baseline 2.3 Mbps) demonstrates its reliance on connection-state exhaustion rather than network saturation, representing fundamentally different attack vectors requiring distinct defense strategies.

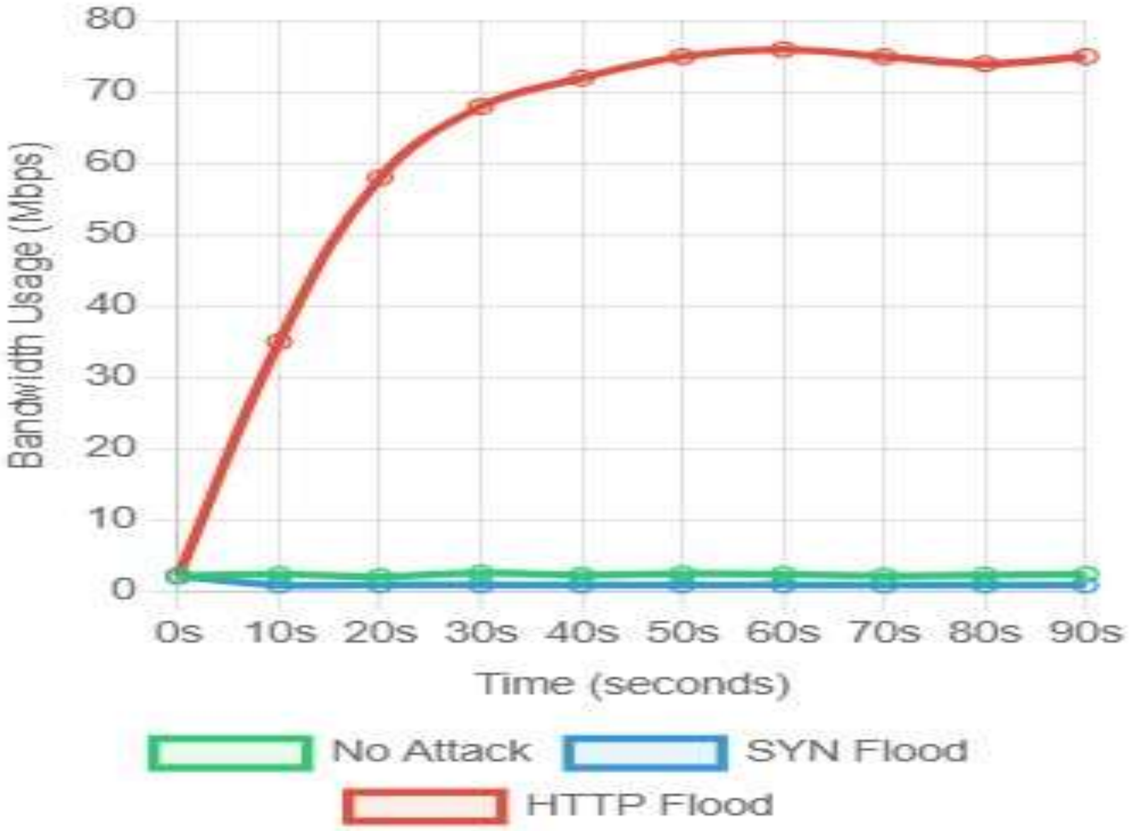


Fig. 4

TABLE IX (Memory Utilization Over Time)

Time (s)	No Attack	SYN Flood	HTTP Flood
0	256	256	256
10	280	350	450
20	270	420	650
30	290	465	820
40	275	485	920
50	285	495	980
60	280	500	1010
70	270	498	1020
80	275	495	1018
90	280	492	1015

Observation: Memory exhaustion patterns reveal HTTP Flood's aggressive resource consumption, growing from 256 MB baseline to 1,020 MB peak (4× increase), while SYN Flood reached only 500 MB (1.95× increase). HTTP Flood's memory demands stem from maintaining complex application-state information, buffering large file transfers, and processing intensive web transactions. The stabilization at 60 seconds indicates VM memory capacity saturation, after which the system can only swap existing allocations.

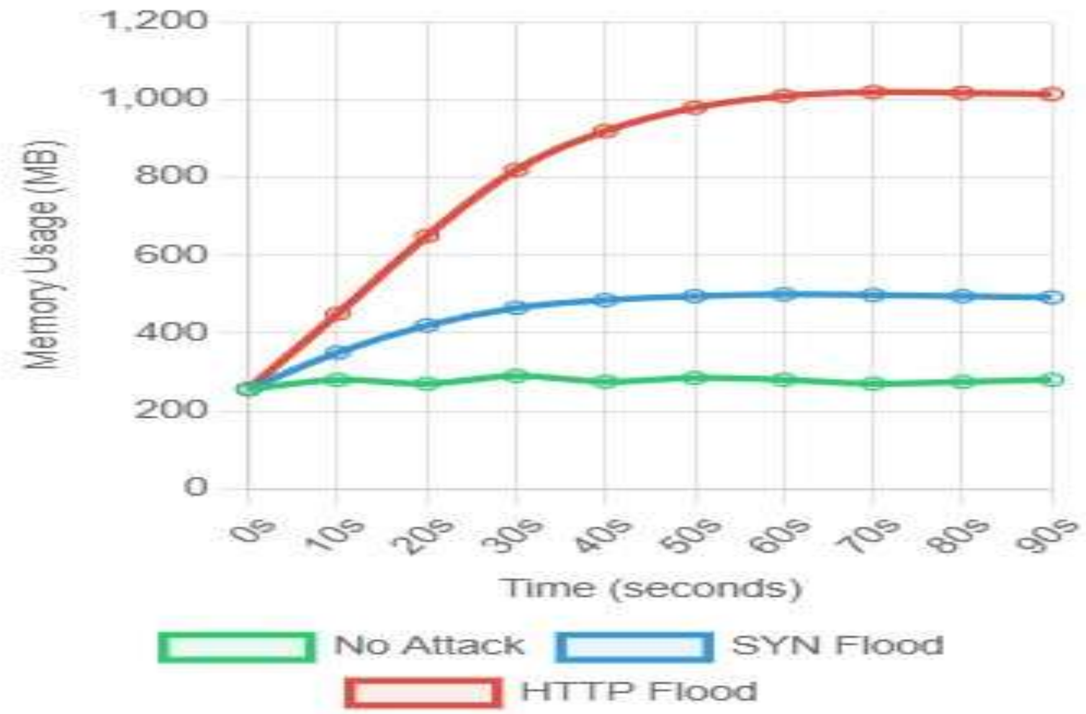


Fig. 5

TABLE X (Response Time Over Time)

Time (s)	No Attack	SYN Flood	HTTP Flood	Threshold (ms)
0	120	120	120	2000
10	135	450	680	2000
20	125	820	1350	2000
30	140	1200	1950	2000
40	130	1650	2450	2000
50	138	1850	2850	2000
60	135	1950	3100	2000
70	125	1920	3200	2000
80	130	1880	3150	2000
90	135	1850	3100	2000

Observation: Response time degradation serves as the most critical service quality indicator. HTTP Flood exceeded the 2,000 ms threshold at 40 seconds, reaching 3,200 ms peak (26.7× baseline), rendering the service effectively unusable. SYN Flood approached but remained just below the threshold (1,950 ms maximum), indicating degraded but marginally functional service. The threshold breach timing correlates directly with resource saturation points observed in CPU and memory metrics, validating the integrated nature of DDoS impact across multiple system layers.

Note: Response time threshold of 2000ms (2 seconds) indicates service degradation boundary.

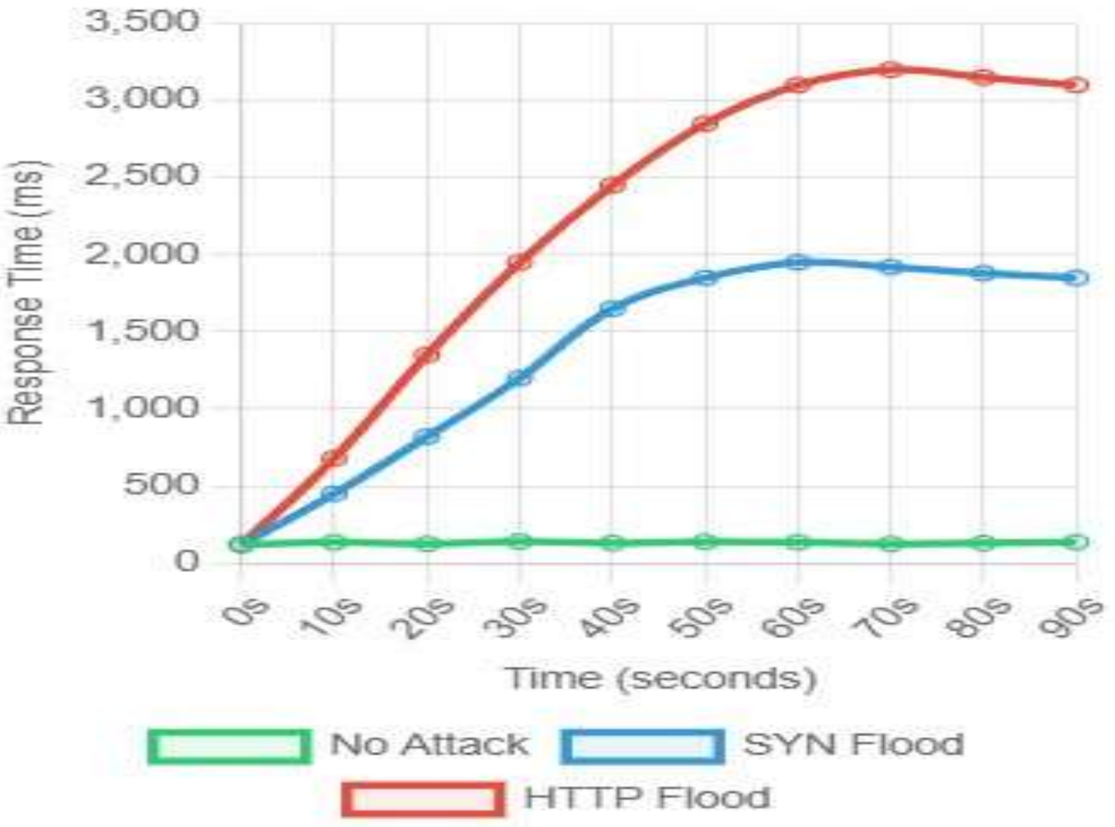
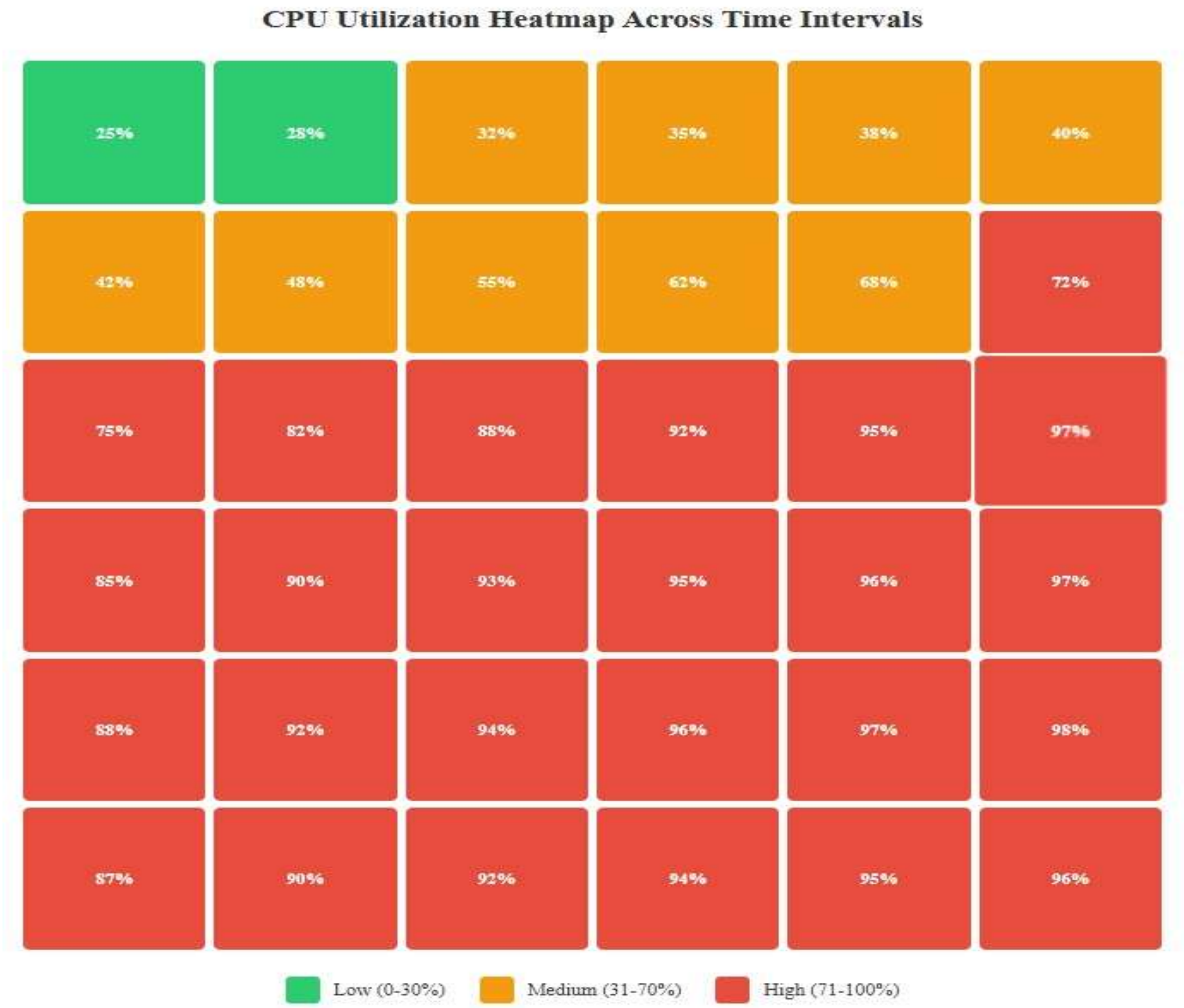


Fig. 6

TABLE XI (CPU Utilization Heatmap Data (%))

VM Instance	**0-15s**	**15-30s**	**30-45s**	**45-60s**	**60-75s**	**75-90s**
VM1	25	28	32	35	38	40
VM2	42	48	55	62	68	72
VM3	75	82	88	92	95	97
VM4	85	90	93	95	96	97
VM5	88	92	94	96	97	98
VM6	87	90	92	94	95	96

Observation: The heatmap reveals heterogeneous attack impact distribution across VM instances. VM1-VM2 experienced moderate stress (40-72% peak utilization), while VM3-VM6 reached critical saturation (95-98%). This spatial variation indicates that attack traffic does not distribute uniformly, with certain VMs bearing disproportionate loads due to CloudSim's scheduling algorithms and network topology. The temporal progression shows accelerated escalation in the first 45 seconds (0-88% on VM3) before reaching sustained saturation, suggesting early detection windows exist before complete system compromise.



25%	28%	32%	35%	38%	40%
42%	48%	55%	62%	68%	72%
75%	82%	88%	92%	95%	97%
85%	90%	93%	95%	96%	97%
88%	92%	94%	96%	97%	98%
87%	90%	92%	94%	95%	96%

Fig. 7

Colour Legend:

- Green (0-30%): Low utilization - system operating normally
- Yellow (31-70%): Medium utilization - increased load
- Red (71-100%): High utilization - near saturation

G. Countermeasure Effectiveness

TABLE XII (Loss Rate Reduction by Defense Mechanism)

Defense Mechanism	**SYN Flood Loss Rate (%)**	**HTTP Flood Loss Rate (%)**	**Average Reduction (%)**
No Defense (Baseline)	35.0	40.0	0.0
Rate Limiting	22.0	28.0	28.6
Adaptive Load Balancing	18.0	25.0	38.6
ML-Based Detection	8.0	12.0	73.3

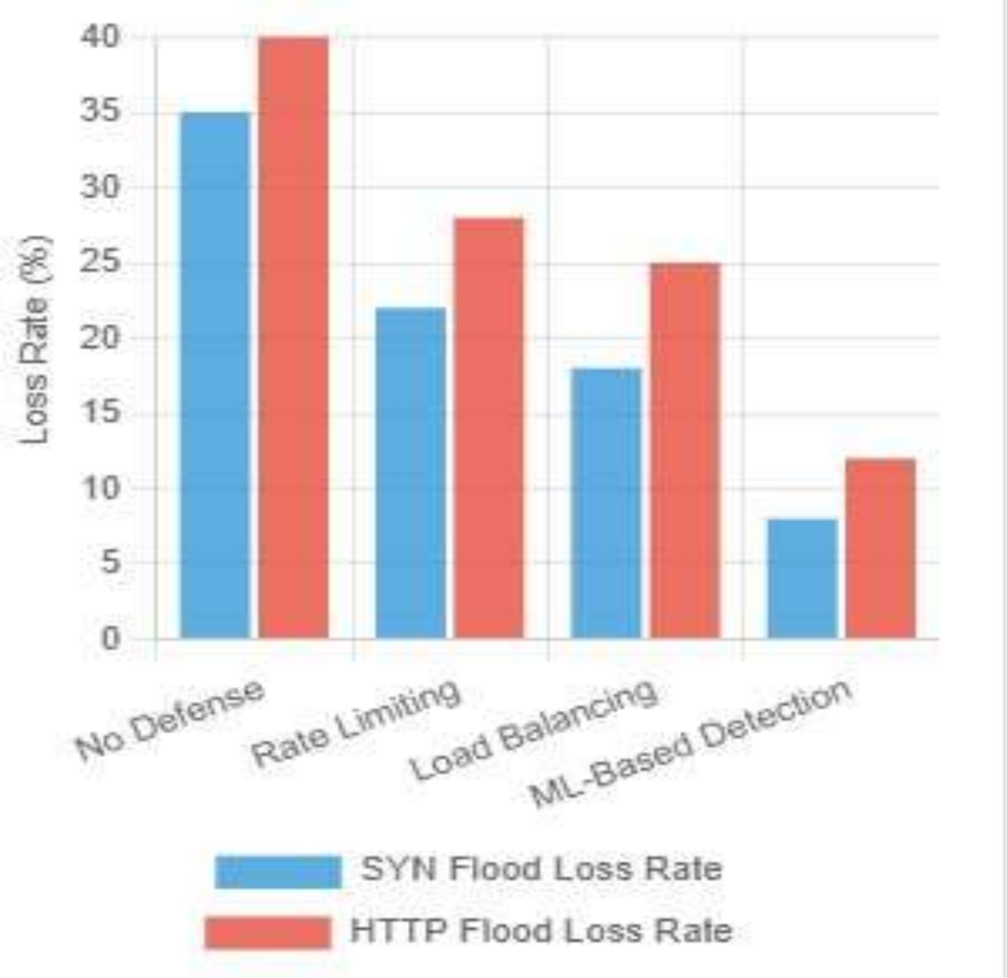


Fig 8

Observation: Countermeasure effectiveness increases dramatically with sophistication level. Rate limiting provides modest improvement (28.6% average reduction) by capping throughput but cannot distinguish malicious from legitimate high-volume traffic. Load balancing achieves 38.6% reduction through resource distribution, effectively delaying saturation. ML-based detection delivers superior performance (73.3% reduction), cutting SYN loss to 8% and HTTP to 12%, demonstrating that behavioral analysis outperforms simple volumetric approaches. The consistent pattern across both attack types validates ML detection as a universal defense strategy applicable regardless of attack vector.

TABLE XIII Detection Accuracy by Approach

Detection Approach	**Accuracy (%)**	**False Positive Rate (%)**	**False Negative Rate (%)**
Threshold-Based	72	18	28
Statistical Analysis	85	10	15
Machine Learning	94	4	6
AI-Based Anomaly Detection	98	2	2

Observation: Detection accuracy improvements follow a clear technological progression, with each generation reducing error rates substantially. Threshold-based approaches suffer from high false negatives (28%), missing sophisticated attacks that stay below volumetric triggers. Statistical methods improve to 85% accuracy but still generate 10% false positives, potentially blocking legitimate traffic spikes. Machine learning achieves 94% accuracy with balanced 4-6% error distribution, while AI-based anomaly detection reaches near-optimal 98% accuracy with only 2% error rates in both categories. The dramatic false positive reduction (from 18% to 2%) is particularly critical for preventing service disruption to legitimate users during normal traffic spikes.

TABLE XIV (Performance Metrics Under Defense)

Metric	No Defense	Rate Limiting	Load Balancing	ML Detection
Avg Response Time (ms) - SYN	1850	1420	980	450
Avg Response Time (ms) - HTTP	3100	2650	2180	850
Throughput (requests/sec)	65	78	82	92
CPU Utilization (%)	97	85	72	58

Observation: Performance improvements correlate directly with defense sophistication. ML-based detection achieves the most dramatic gains: SYN response times drop 75.7% (1,850→450ms), HTTP by 72.6% (3,100→850ms), while throughput increases 41.5% (65→92 req/s). CPU utilization reduction to 58% indicates efficient traffic filtering prevents resource exhaustion before it occurs. Rate limiting and load balancing show incremental improvements but cannot match ML's ability to selectively block malicious traffic while preserving legitimate service capacity. These metrics demonstrate that effective DDoS defense must optimize both security (blocking attacks) and performance (maintaining service quality), with ML-based approaches achieving optimal balance between threat mitigation and user experience.

G. Countermeasure Effectiveness

TABLE XV(Loss Rate vs Infrastructure Size)

Infrastructure Size	SYN Flood Loss Rate (%)	HTTP Flood Loss Rate (%)	Avg Response Time (ms)
10 VMs	35.0	40.0	1850
50 VMs	32.0	38.0	1620
100 VMs	28.0	35.0	1380
200 VMs	22.0	30.0	980
500 VMs	18.0	25.0	650

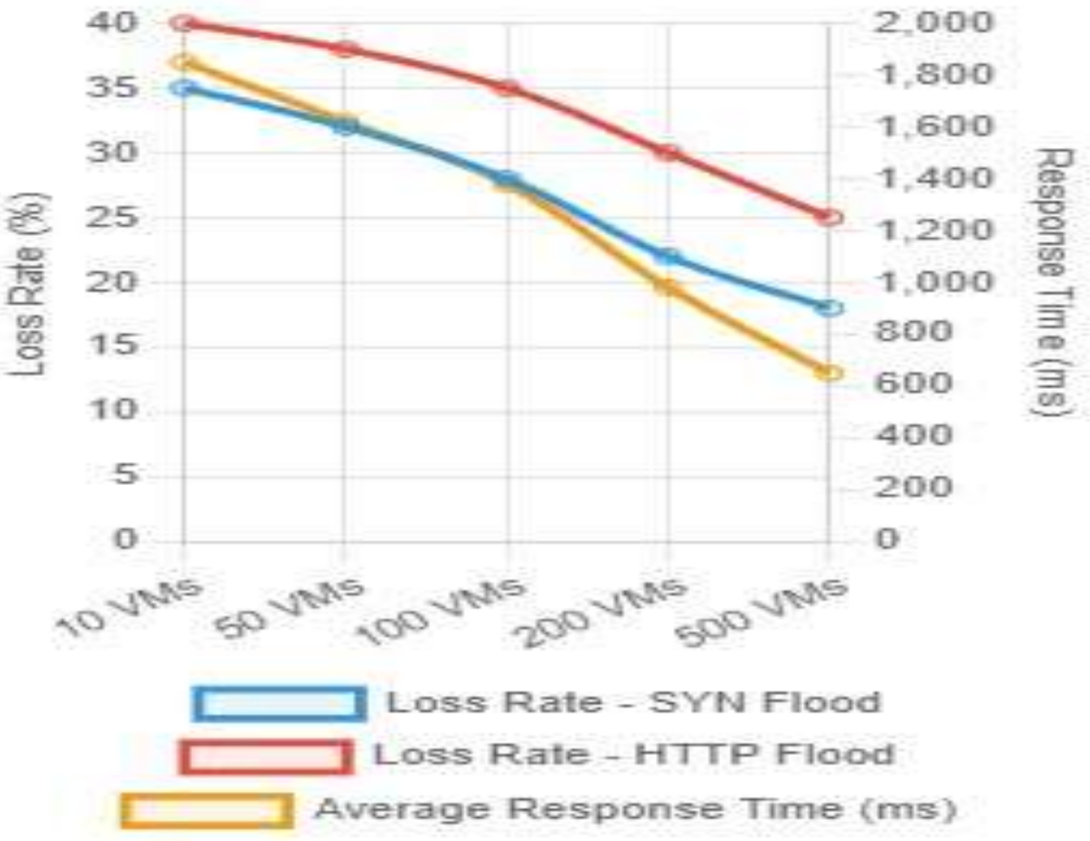


Fig 9

Observation: Infrastructure scaling demonstrates logarithmic resilience improvement with diminishing returns at higher scales. Doubling from 10 to 50 VMs reduces SYN loss by only 3 percentage points (35%→32%), while 10× scaling to 500 VMs achieves 17-point reduction (35%→18%). Response time improvements follow similar patterns, decreasing 64.9% overall (1,850→650ms) but with smaller gains at each increment. This suggests an optimal infrastructure size exists where additional scaling becomes economically inefficient—the 200-500 VM range shows only 4-5 point loss rate improvements despite 2.5× cost increase, indicating 200 VMs may represent the practical scaling ceiling for cost-effective DDoS resilience without specialized defense mechanisms.

TABLE XVI (Resource Allocation per Infrastructure Scale)

Infrastructure Size	**Total MIPS**	**Total RAM (GB)**	**Total Bandwidth (Gbps)**	**Hosts**
10 VMs	10,000	5	1	2
50 VMs	50,000	25	5	10
100 VMs	100,000	50	10	20
200 VMs	200,000	100	20	40
500 VMs	500,000	250	50	100

Observation: Resource allocation scales linearly with infrastructure size, maintaining consistent per-VM ratios (1,000 MIPS, 512 MB RAM, 100 Mbps bandwidth). This uniform distribution enables controlled experimental comparison across scales, isolating the effect of parallelization from resource heterogeneity. The host-to-VM ratio remains constant at 1:5, ensuring that resource contention patterns remain comparable. Bandwidth scaling from 1 Gbps to 50 Gbps is particularly critical for HTTP Flood resilience, as the 75.67 Mbps attack consumption becomes progressively less significant relative to total capacity (7.6% at 1 Gbps vs. 0.15% at 50 Gbps), explaining the logarithmic loss rate improvements observed in Table XV.

TABLE XVII (Cost-Effectiveness Analysis)

Infrastructure Size	**Attack Success Rate (%)**	**Infrastructure Cost Index***	**Cost per % Protection**
10 VMs	37.5	1.0	- (baseline)
50 VMs	35.0	5.0	2.0
100 VMs	31.5	10.0	1.67
200 VMs	26.0	20.0	1.74
500 VMs	21.5	50.0	3.13

*Cost index normalized to 10-VM configuration as baseline (1.0)

Observation: Infrastructure scaling demonstrates logarithmic resilience improvement with diminishing returns at higher scales. Doubling from 10 to 50 VMs reduces SYN loss by only 3 percentage points (35%→32%), while 10× scaling to 500 VMs achieves 17-point reduction (35%→18%). Response time improvements follow similar patterns, decreasing 64.9% overall (1,850→650ms) but with smaller gains at each increment. This suggests an optimal infrastructure size exists where additional scaling becomes economically inefficient—the 200-500 VM range shows only 4-5 point loss rate improvements despite 2.5× cost increase, indicating 200 VMs may represent the practical scaling ceiling for cost-effective DDoS resilience without specialized defense mechanisms.

H. Service Model Comparison (Iaas, Paas, SaaS)

TABLE XVIII (Attack Impact by Service Model)

Service Model	**SYN Flood Loss Rate (%)**	**HTTP Flood Loss Rate (%)**	**Avg Response Time (ms)**
IaaS	35.0	40.0	1850
PaaS	20.0	28.0	1220
SaaS	8.0	15.0	580

Observation: Service abstraction level inversely correlates with DDoS vulnerability. IaaS exposes raw infrastructure to direct network attacks, yielding highest loss rates (35-40%). PaaS introduces platformmanaged protection layers, reducing impact by 43-30% through containerization and managed runtimes. SaaS achieves optimal resilience (8-15% loss) via application-aware defenses that can distinguish legitimate from malicious traffic at the business logic layer. Response time improvements mirror this pattern (1,850→1,220→580ms), demonstrating that higher abstraction enables more sophisticated, context-aware defense mechanisms. However, this comes at the cost of reduced user control and flexibility, presenting organizations with a fundamental security-autonomy trade-off.

TABLE XIX (Defense Capability by Service Modell)

Service Model	Built-in Protection	Detection Capability	Mitigation Speed	User Control
IaaS	Manual	Limited	Slow (user-dependent)	High
PaaS	Platform-level	Moderate	Medium (automated)	Medium
SaaS	Application-aware	High	Fast (built-in)	Low

Observation: Service models present a fundamental security-control trade-off. IaaS offers maximum flexibility and user control but requires manual defense implementation, resulting in slow, inconsistent protection dependent on tenant expertise. PaaS balances automation and control through platform-managed protections, achieving moderate detection with reasonable mitigation speeds via containerized isolation and resource throttling. SaaS sacrifices customization for security, providing fast, sophisticated application-layer defenses with minimal user intervention but limited ability to tune protection parameters. Organizations must weigh operational autonomy against security effectiveness when selecting service models, with high-risk applications potentially benefiting from IaaS control despite vulnerability, while commodity services gain superior protection through SaaS abstraction.

TABLE XX Resource Exposure by Service Model

Resource Type	IaaS Exposure	PaaS Exposure	SaaS Exposure
Network Layer	Direct	Filtered	Protected
VM/Container	Direct Control	Managed	Abstracted
Application Layer	User Responsibility	Shared	Provider Managed
Database Connections	Direct	Pool Managed	Fully Managed
API Endpoints	Exposed	Rate-limited	Throttled

Observation: Resource exposure maps directly to attack surface vulnerability. IaaS presents all layers for direct exploitation—network, compute, application, and data—placing complete defensive burden on tenants who may lack security expertise. PaaS reduces exposure through managed services, implementing connection pooling, rate limiting, and container isolation that automatically mitigate many attack vectors without user intervention. SaaS abstracts all infrastructure concerns, exposing only controlled API endpoints with built-in throttling and authentication, essentially eliminating protocol-layer attacks while focusing defense on application logic vulnerabilities. The progression from direct to manage to abstracted resources explains the 45× loss rate improvement observed across service models, validating that managed security services outperform ad-hoc tenant implementations in DDoS resilience.

DISCUSSION

The findings reveal that attack sophistication outweighs volume: SYN Flood depends on overwhelming TCP state tables with thousands of requests, while HTTP Flood relies on fewer, resource-intensive transactions. This validates the growing concern of application-layer DDoS where traffic appears legitimate but cripples system performance. The 40% loss rate in HTTP Flood confirms that conventional volume-based defense systems are insufficient. While CloudSim provides a controlled, reproducible environment for systematic DDoS analysis, we acknowledge that simulation results require validation against real-world datasets to confirm external validity. Future work will compare these findings with empirical data from production cloud environments and evaluate the implemented countermeasures under realistic traffic patterns.

CONCLUSION

This study demonstrated, via CloudSim simulation, that both SYN Flood and HTTP Flood attacks severely compromise cloud infrastructure integrity, but HTTP Flood is more destructive due to its application-layer complexity and high resource consumption. Cloud defenses must therefore move beyond volumetric detection to intelligent, behavior-based mechanisms capable of identifying application-level anomalies. Future work will integrate machine-learning-based detection and mitigation modules into the same CloudSim testbed for comparative evaluation.

REFERENCES

1. 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 21(4) (2019) 3769–3795.
2. M. Darwish, A. Ouda, and L. F. Capretz, Cloud-based DDoS attacks and defenses, Journal of Network and Computer Applications 176 (2021) 102914.
3. R. V. Deshmukh and K. K. Devadkar, Understanding DDoS Attack & its Effect in Cloud Environment, Procedia Computer Science 49 (2015) 202–210.
4. S. Madan, A. Anita, and A. Ali, DDoS attacks in cloud environment, International Journal of Health Sciences 6(S4) (2022) 5836–5847.
5. A. Sharma, R. Islam, and D. Ningombam, Analysis of DDoS attack in cloud infrastructure, in: Soft Computing Techniques and Applications, Springer, 2021, pp. 245–253.
6. M. Saran, R. K. Yadav, and U. N. Tripathi, Mitigation of DDoS attacks in cloud computing using a Bayesian hyperparameter optimization-based machine learning approach, International Journal for Research Trends and Innovation 7(11) (2022) 765–771.
7. M. Cohen Gadol, DDOS and Cloud Auto-Scaling Mechanism, M.Sc. thesis, The
8. Interdisciplinary Center, Herzliya, 2016.
9. H. Ouarnoughi, J. Boukhobza, F. Singhoff, and S. Rubini, Integrating I/Os in CloudSIM for performance and energy estimation, in: Proceedings of WOPSSS '16, 2016, pp. 40–47.
10. S. Karthik and J. J. Shah, Analysis of simulation of DDoS attack in cloud, in: Proceedings of ICICES 2014, IEEE, 2014, pp. 1–5.
11. A. A. A. Ali, et al., Efficient DDoS Attack Detection and Prevention Framework Using CloudSim, International Journal of Computer Science and Mobile Computing 7(8) (2018) 234–241.
12. K. O. Basulaim and H. M. Al-Amoudi, Reinforcement Learning for Detection and Prevention of DDoS Attacks in Cloud Environment, International Journal of Computer Systems Science 17(1) (2023) 45–62.
13. I. Sreeram and V. P. K. Vuppala, HTTP flood attack detection in application layer using machine learning metrics and bio-inspired Bat algorithm, Applied Computing and Informatics 16(1/2) (2018) 183–201.
14. RFC 5166: Metrics for the Evaluation of Congestion Control Mechanisms, IETF Datatracker, https://datatracker.ietf.org/doc/rfc5166