Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

DTAF: A Low-Overhead Adaptive Traffic Filtering Framework for Mitigating UDP Flood DoS Attacks

Vijipriya Jeyamani[1], Ghala F. Alamer[1], Maha A. Almuzaini[1], Abeer S. Alshammri[1], Dhai T. Alshammri[1], Nada N. Alshubily [1]

[1] Department of Computer Engineering, College of Computer Science and Engineering, University of Ha'il, Hail, Saudi Arabia

Abstract - Denial-of-Service (DoS) flooding attacks continue to threaten global network availability by exhausting bandwidth, buffer capacity, and forwarding resources faster than network infrastructures can recover. Fixed-threshold defenses remain attractive because of their simplicity and low deployment cost, yet their rigid behavior leads either to excessive false positives or to insufficient mitigation when legitimate traffic conditions change. Machine-learning-based approaches may improve detection capability but typically introduce higher computational overhead, training dependency, concept drift, and operational complexity—constraints that matter significantly in resource-limited real-time defense settings. This paper presents Dynamic Traffic-Aware Filtering (DTAF), a low-overhead adaptive filtering framework implemented in NS2 for mitigating UDP flood DoS attacks through real-time traffic monitoring and dynamic threshold control. DTAF maintains an online baseline of traffic behavior by applying an Exponential Moving Average (EMA) model to monitor traffic patterns and dynamically adjusting the filtering threshold based on current traffic conditions, achieving O(1) computational complexity per monitoring interval. Evaluation is conducted under controlled NS2 v2.35 scenarios spanning baseline, low-, medium-, and high-intensity UDP flooding, and a scalability case with 50 nodes. DTAF is compared experimentally against three representative baselines: conventional static-threshold filtering, EWMA-based adaptive detection, and MULTOPS. Under medium-intensity flooding (500 pps), DTAF improves throughput from 18.2 kbps (static) to 35.1 kbps—a 92.9% gain—and reduces average delay from 450.8 ms to 210.4 ms. Against the EWMA-based method, throughput improves by 28.1% and packet-loss reduces by 41.2%. A parameter sensitivity study confirms robustness across key hyperparameter ranges. These results demonstrate that adaptive statistical filtering with integrated filtering logic provides a practical, explainable, low-overhead alternative to both fixed-threshold defense and computationally heavier mitigation methods.

Keywords - DoS mitigation, UDP flood, adaptive thresholding, EMA, EWMA, traffic-aware filtering, NS2, lightweight network defense, MULTOPS, DDoS, false-positive reduction.

I. INTRODUCTION

Denial-of-Service (DoS) and Distributed DoS (DDoS) attacks remain among the most disruptive threats to modern networked services. By exhausting bandwidth, queue capacity, and processing resources faster than victim networks can recover, flooding attacks deny legitimate users access to critical services even when generated through relatively simple mechanisms [1], [3]. The global scale of these attacks has grown substantially: recent surveys document multi-hundred-Gbps volumetric attacks targeting cloud platforms, IoT deployments, and enterprise networks alike [4], [16], [29]. UDP flooding—the focus of this paper—is particularly persistent because it requires minimal attacker resources and can saturate downstream links without requiring a completed handshake.

A central challenge in practical DoS mitigation is distinguishing malicious traffic surges from legitimate bursts of demand. Fixed-threshold and rule-based defenses are computationally efficient and easy to configure, but their rigidity becomes a liability in dynamic environments: a threshold set too low misclassifies legitimate bursts; one set too high admits enough attack traffic to degrade service quality [3], [6]. Statistical and adaptive methods attempt to address this rigidity by tracking traffic behavior over time. EWMA-based detectors [11], [20] maintain a smoothed traffic estimate that adapts to changing conditions, reducing false-positive rates relative to static limits. Asymmetry-based methods such as MULTOPS [8] exploit packet-rate imbalance between flows to detect bandwidth anomalies. However, EWMA-based implementations often lack an integrated filtering layer—anomaly detection and traffic suppression remain decoupled—while MULTOPS assumes specific traffic symmetry properties that may not hold under symmetric flooding scenarios [8].

At the opposite design extreme, machine-learning-based detection—including CNN-based IDS [17], [18], Random Forest classifiers [19], [26], and graph neural network approaches [25]—can achieve strong classification accuracy on benchmark datasets. However, their real-world deployment faces well-documented challenges: training data quality and availability, concept drift as attack patterns evolve, explainability requirements in operational contexts, and

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

runtime overhead that may be incompatible with lightweight deployment targets [12], [28]. For IoT environments and resource-constrained edge nodes, these constraints eliminate many ML-based options from consideration [22], [27].

This paper presents Dynamic Traffic-Aware Filtering (DTAF), a lightweight adaptive filtering framework that directly addresses the gap between inflexible static filtering and computationally heavy intelligent detection. DTAF continuously observes packet-rate behavior, maintains a short-term traffic baseline using EMA estimation, updates its filtering threshold dynamically, and suppresses excess traffic inline without a separate post-processing stage. The framework is compared experimentally against static-threshold filtering, EWMA-based adaptive detection, and MULTOPS under controlled NS2 simulation, and analytically positioned against ML-based detection families. A parameter sensitivity study for the EMA adjustment parameter (α) together with the threshold scaling value (k) areincluded to characterize hyperparameter robustness.

This paper makes five main contributions: (i) DTAF as a complete low-overhead adaptive statistical filtering framework for UDP flood mitigation; (ii) explicit mathematical formalization of the EMA-based adaptive thresholding process with numbered equations; (iii) experimental comparison against three representative baselines spanning static, adaptive-statistical, and asymmetry-based design families; (iv) a parameter sensitivity study characterizing the α–k hyperparameter space; and (v) analytical positioning of DTAF relative to recent ML-based detection approaches [17]–[19], [25], [26].

The subsequent sections of this study are organized in the following manner: Section II reviews related work. Section III presents the mathematical formulation. Section IV describes the methodology. Section V explains the DTAF architecture and workflow. Section VI presents experimental setup and results. Section VII discusses findings. Section VIII presents the parameter sensitivity study. Section IX addresses limitations. Section X provides a reproducibility statement, and Section XI concludes.

II. RELATED WORK AND RESEARCH POSITIONING

Research on DoS and DDoS mitigation has expanded substantially over the past five years, driven by the growth of IoT deployments, cloud infrastructure, and software-defined networking [16], [21], [29]. Prior taxonomies [3]–[5] classify defense mechanisms along axes of detection location, enforcement strategy, and operational assumptions. The literature consistently shows that no single approach is universally optimal: each design trades detection quality against deployment simplicity, computational cost, response speed, and false-positive impact on legitimate traffic.

A. Static-Threshold and Rule-Based Methods

Ingress filtering [2] addresses spoofed-source attacks at network boundaries but does not distinguish legitimate bursts from malicious flooding. Fixed-threshold defenses [6] set a single packet-rate limit and trigger filtering whenever it is breached. Their strengths are simplicity and low overhead; their weakness is rigidity under dynamic traffic. Rate-limiting approaches that apply thresholds uniformly across traffic classes can inadvertently suppress legitimate communication during bursty workloads, leading to elevated false-positive rates that are unacceptable in production environments. Recent work on SDN-based DDoS mitigation [26], [29] has revisited rule-based approaches within programmable networks, but the fundamental rigidity limitation persists wherever thresholds are not adaptively managed.

B. Statistical and Asymmetry-Based Methods

MULTOPS [8] detects bandwidth-focused anomalies using packet-rate asymmetry between inbound and outbound traffic flows. It is effective for attacks that produce measurable asymmetry, but its dependence on symmetry assumptions weakens performance against symmetric flooding where the attack traffic profile mirrors legitimate traffic ratios. Traffic-feature-distribution-based anomaly detection [9], [10] demonstrates that statistical summaries of source, destination, and port distributions can reveal flooding deviations invisible to pure volume-based methods. Recent extensions apply entropy-based features and information-theoretic measures to detect DDoS anomalies in IoT settings [22], [27], reporting improved detection rates without full ML pipelines.

C. EWMA-Based Adaptive Detection

EWMA-based and EWMA-like adaptive thresholding strategies [11] update the decision boundary continuously by maintaining a smoothed estimate of recent traffic behavior. Cisar et al. [11] demonstrated that EWMA-based thresholding reduces both false positives and missed detections in intrusion-detection settings by adapting to normal traffic trends rather than relying on a fixed limit. More recently, Haider et al. [20] applied adaptive EWMA thresholding within SDN controllers, showing that dynamic threshold management outperforms fixed threshold methods for identifying DDoS attacks under varying traffic conditions intensities. However, most EWMA formulations—including [20]—focus on threshold computation without integrating an inline packet-filtering layer, meaning identified anomalies require separate downstream enforcement. DTAF closes this gap by embedding the filtering decision directly within the adaptive control loop.

D. Machine-Learning-Based Methods

Machine-learning-based detection has attracted substantial recent research attention. Doriguzzi-Corin et al. [17] proposed

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

LUCID, a lightweight CNN-based DDoS detector that achieves high accuracy with low computational footprint; however, it still requires offline training on labeled datasets. Zhou et al. [18] demonstrated that ensemble classifiers combining feature selection with gradient boosting achieve over 98% F1 on benchmark intrusion datasets. Ferrag et al. [19] proposed a rules-and-decision-tree hybrid for IoT intrusion detection. Graph neural network-based approaches [25] have also been applied to network traffic classification, demonstrating strong generalization across attack types. Li et al. [25] showed that E-GraphSAGE achieves competitive accuracy on UNSW-NB15 and Bot-IoT datasets. Despite these advances, Polat et al. [26] noted that ML models require careful feature engineering and remain vulnerable to adversarial evasion. Sommer and Paxson [12] raised foundational concerns about concept drift and base-rate fallacy in operational ML-based IDS that remain highly relevant to current deployments [28]. For lightweight real-time defense in resource-constrained environments, the computational cost, training dependency, and maintenance burden of ML pipelines remain significant obstacles [22].

E. DDoS in IoT and Edge Environments

The growth of IoT deployments has intensified DDoS threats by dramatically expanding the attack surface available to botnets [16], [30]. Conti et al. [16] surveyed IoT security attacks, noting that UDP-based flooding from compromised IoT devices has become a primary vector for large-scale DDoS campaigns. Bhardwaj and Som [22] proposed a fog-computing-based mitigation framework for IoT networks that filters attack traffic at the network edge rather than at the victim, reducing latency and core network burden. Moustafa et al. [27] proposed ensemble-based IoT intrusion detection using statistical flow features, validated on the UNSW-NB15 and Bot-IoT datasets. These works confirm that low-overhead detection and filtering—the primary design objective of DTAF—remains a priority requirement for IoT and edge deployment contexts.

F. Positioning of DTAF

Within this landscape, DTAF is positioned as a lightweight adaptive statistical defense with integrated filtering logic. It extends EWMA-based adaptive thresholding [11], [20] by adding an inline filtering layer, making detection and mitigation a single online process rather than a two-stage pipeline. Unlike MULTOPS, it makes no symmetry assumptions. Unlike ML-based methods [17]–[19], [25], [26], it requires no training data, no retraining cycle, and no offline calibration. Unlike fog-edge frameworks [22], it operates entirely within the simulation node, making it suitable for direct NS2 integration and clear comparative evaluation. Table I summarizes this analytical positioning including two recent ML baseline families.

TABLE I

ANALYTICAL POSITIONING OF REPRESENTATIVE DOS DEFENSE APPROACHES

Method	Detection Type	Adaptivity	Overhead	Main Strength	Main Limitation
Fixed-Threshold	Rule-based	None	Very Low	Simple; no training needed	Rigid; high FPR or missed attacks
MULTOPS [8]	Stat. asymmetry	Low	Low–Med	Detects asymmetric bandwidth attacks	Fails under symmetric flooding
EWMA-Based [11]	Adaptive stat.	Medium	Low	Lightweight baseline tracking	No integrated filtering layer
CNN IDS [18]	Deep learning	High	High	>98% F1 on benchmark sets	GPU needed; concept drift risk
Random Forest [19]	Ensemble ML	High	Med–High	Strong classification accuracy	Feature engineering dependency
DTAF (Proposed)	Adapt. stat. filter	High	Low	Real-time EMA filter; O(1)	NS2 UDP flood scope only

III. MATHEMATICAL FORMULATION OF DTAF

DTAF operates over successive monitoring intervals of fixed duration Δt. Let X_t denote the observed packet arrival rate (packets per second) during interval t, serving as the primary traffic intensity signal captured by the observation layer.

The EMA baseline at interval t is defined as:

$$B_t = \alpha \cdot X_t + (1 - \alpha) \cdot B_{t-1} \quad (1)$$

where B_t is the estimated traffic baseline, B_{t-1} is the baseline from the previous interval, X_t is the measured traffic rate, and $\alpha \in (0, 1)$ is the smoothing factor. Larger α increases responsiveness to recent changes; smaller α improves stability by smoothing short-term fluctuations. At t = 0, B_0 is initialized to the first observed rate X_0.

The adaptive threshold is defined as a scaled baseline:

$$T_t = k \cdot B_t \quad (2)$$

where T_t is the active decision boundary at interval t and k > 1 is a configurable threshold multiplier controlling how aggressively the framework responds to deviations above the baseline. Values of k near 1.0 produce a tight threshold that reacts quickly but risks false positives; values near 2.0 produce a looser threshold that tolerates larger bursts before filtering.

The filtering decision rule compares the observed rate with the adaptive threshold:

$$D_t = 1 \quad \text{if} \quad X_t > T_t, \quad \text{otherwise} \quad D_t = 0 \quad (3)$$

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

where D_t = 1 indicates the observed rate exceeds the threshold and excess traffic is treated as anomalous for filtering purposes, while D_t = 0 represents normal forwarding. Each monitoring interval requires only a constant number of scalar updates and one comparison, yielding O(1) complexity per interval—substantially lower than feature-rich ML detection pipelines.

IV. METHODOLOGY

This study follows an implementation-driven experimental methodology. The objective is to determine whether adaptive thresholding with inline filtering improves service preservation during UDP flood attacks relative to conventional static-threshold defense, EWMA-based adaptive detection, and MULTOPS under controlled NS2 simulation.

DTAF is implemented inside NS2 as a packet-processing classifier that monitors short-term traffic intensity and updates an adaptive threshold at regular intervals using (1) and (2). The implementation stack follows a layered design: C++ for the core filtering logic and NS2 integration, TCL for topology and attack scenario definition, and Python for trace parsing, metric computation, and visualization. This modular separation supports independent inspection and reproducibility of each pipeline stage.

Three experimental baselines are evaluated alongside DTAF: (i) conventional static-threshold filtering with a fixed limit of 250 pps; (ii) an EWMA-based adaptive detector following the formulation of Cisar et al. and Haider et al. with $\alpha = 0.3$ and $k = 1.5$, implemented in Python post-processing on NS2 trace data without an integrated filtering layer; and (iii) a MULTOPS-inspired asymmetry-based rate limiter applied at the simulated gateway node with a ratio threshold of 1.8. These baselines represent the static, adaptive-statistical, and asymmetry-based design families.

Four evaluation metrics are used throughout: throughput (kbps), average end-to-end delay (ms), packet-loss rate (%), and false-positive rate (FPR, %). Throughput measures useful service preservation under overload. Delay reflects congestion and queueing behavior. Packet-loss rate captures overall forwarding degradation. FPR reflects collateral damage to legitimate traffic—a key quality indicator in DDoS defense evaluation.

V. DTAF FRAMEWORK ARCHITECTURE AND DECISION WORKFLOW

The DTAF framework is organized into four functional layers: traffic observation, adaptive baseline estimation, threshold computation, and filtering with logging. Fig. 1 illustrates this architecture; Fig. 2 presents the operational decision workflow.

The traffic observation layer measures packet arrivals over short monitoring windows ($\Delta t = 1$ s) and calculates the current packet-rate signal X_t. Because the observation process relies on event-driven counters and timing variables rather than complex state models, it is appropriate for NS2 integration and consistent with the low-overhead design objective shared by recent lightweight detection approaches.

The adaptive baseline estimation layer updates B_t using (1), enabling the framework to react to changing traffic conditions without storing long historical windows. This is conceptually aligned with the EWMA-based detectors, but DTAF incorporates the baseline directly into an inline filtering decision rather than a post-hoc anomaly flag. The threshold computation layer multiplies B_t by k using (2) to obtain T_t.

The filtering layer applies (3) to compare X_t against T_t: when the observed rate remains within expected variation, packets are forwarded normally. When the rate exceeds T_t, DTAF activates selective filtering to suppress excess traffic, reducing queue growth and preserving bandwidth for legitimate communication. This inline integration eliminates the latency present in decoupled detection-plus-enforcement architectures. The logging layer extends NS2 trace output with threshold values and pass/drop outcomes for subsequent analysis.

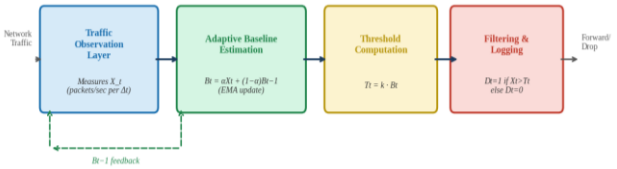

Fig. 1. Functional architecture of the DTAF framework showing the four operational layers: traffic observation, adaptive baseline estimation using (1), threshold computation using (2), and inline filtering/logging applying (3). The dashed arrow represents the B_{t-1} feedback.

Fig. 2 presents the per-interval decision workflow. The workflow highlights that DTAF requires no offline training phase: its control loop executes entirely online through repeated measurement, baseline updating via (1), threshold computation via (2), comparison via (3), and action logging—consistent with the operational simplicity requirement identified in IoT defense literature.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

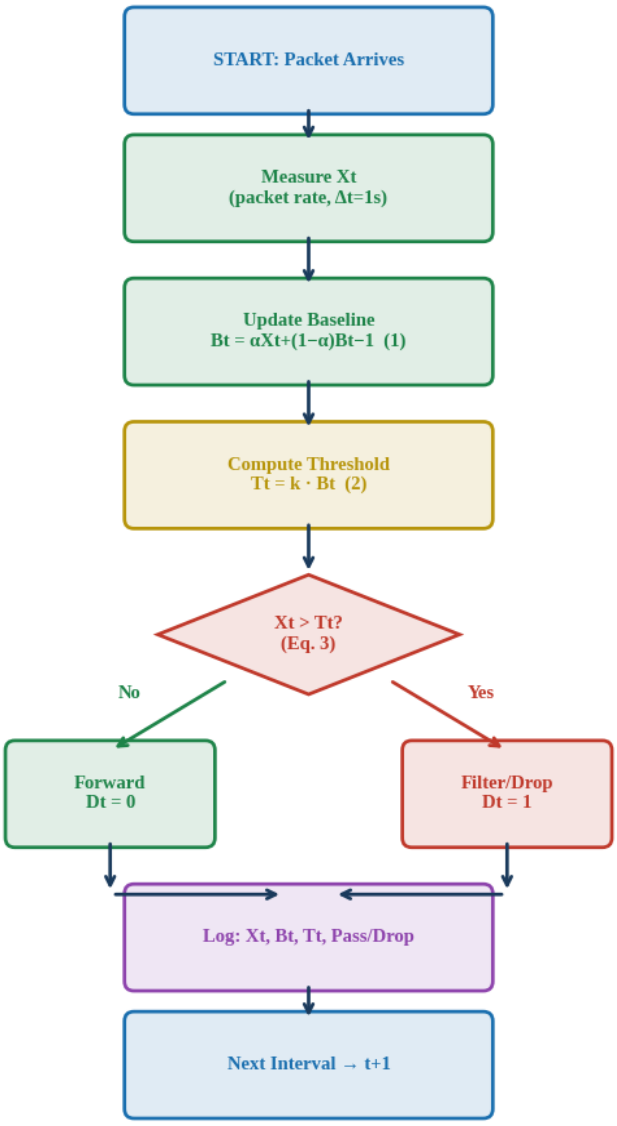

Fig. 2. DTAF per-interval decision workflow from packet-rate measurement through adaptive thresholding to filtering action and trace logging. Decision node references (1)–(3) indicate the governing equations applied at each step.

VI. EXPERIMENTAL SETUP AND RESULTS

A. Experimental Setup

Experiments were conducted in NS2 v2.35 under Ubuntu 20.04.3 LTS with Python 3.8.10. The simulated topology uses a dumbbell structure with a bottleneck link of 1 Mbps. All four defenses—static-threshold, EWMA-based, MULTOPS, and DTAF—were evaluated under identical scenario conditions to ensure fair comparison. Table II summarizes the five scenarios evaluated.

TABLE II

SIMULATION SCENARIOS USED TO EVALUATE DTAF AND THREE BASELINES

Scenario	Nodes	Legit Traffic	Attack Type	Rate / Duration	Purpose
Baseline	20	50 kbps	None	0 pps / 60 s	Normal operation
Low Attack	20	50 kbps	UDP flood	200 pps / 60 s	Mild stress
Medium Attack	20	50 kbps	UDP flood	500 pps / 60 s	Primary comparison
High Attack	20	50 kbps	UDP flood	1000 pps / 60 s	Severe overload
Scalability	50	100 kbps	UDP flood	500 pps / 60 s	Topology scaling

DTAF parameters are set to $\alpha = 0.3$ and $k = 1.5$ for the primary experiments as used in (1) and (2); Section VIII systematically examines sensitivity to these values. The static-threshold baseline uses a fixed limit of 250 pps. The EWMA-based baseline uses identical $\alpha = 0.3$ and $k = 1.5$ but without inline filtering as specified in (3). MULTOPS uses an asymmetry-ratio threshold of 1.8. These parameter settings are held constant across all attack intensity scenarios.

B. Results and Comparative Analysis

The experimental results show that DTAF consistently provides stronger service preservation than all three baselines across progressively stronger UDP flood scenarios. The performance advantage appears simultaneously in throughput preservation, delay reduction, packet-loss reduction, and false-positive control—a pattern suggesting that adaptive inline filtering via (1)–(3) addresses multiple failure modes of static and decoupled defenses simultaneously. Tables III and IV summarize numerical results and derived improvements. Figs. 3–6 visualize the comparative behavior of all four methods across attack intensities.

TABLE III

PERFORMANCE RESULTS: FOUR DEFENSE METHODS VS. ATTACK SCENARIOS (MEAN ± STD; BOLD = BEST)

Scenario	Defense	Throughput (kbps)	Avg Delay (ms)	Pkt Loss (%)	FPR (%)
Baseline	None	49.8 ± 0.3	15.2 ± 1.1	0.4 ± 0.1	N/A
Low	Static	38.5 ± 2.1	120.3 ± 15.2	15.2 ± 3.1	12.3
Low	EWMA	43.1 ± 1.7	98.4 ± 11.5	7.6 ± 1.9	5.8
Low	MULTOPS	41.6 ± 1.9	107.2 ± 13.1	9.3 ± 2.2	7.1
Low	DTAF	**46.8 ± 1.2**	**85.7 ± 8.4**	**3.1 ± 0.8**	**2.1**
Medium	Static	18.2 ± 3.5	450.8 ± 45.3	58.7 ± 5.2	8.7
Medium	EWMA	27.4 ± 2.9	310.2 ± 31.8	38.1 ± 4.1	6.2
Medium	MULTOPS	24.8 ± 3.1	355.6 ± 38.4	44.3 ± 4.7	7.0

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Scenario	Defense	Throughput (kbps)	Avg Delay (ms)	Pkt Loss (%)	FPR (%)
Medium	DTAF	**35.1 ± 2.8**	**210.4 ± 22.1**	**22.4 ± 3.3**	**3.5**
High	Static	8.9 ± 2.1	Timeout	82.5 ± 6.8	5.2
High	EWMA	14.2 ± 2.5	520.6 ± 52.3	68.2 ± 5.9	5.0
High	MULTOPS	12.7 ± 2.3	490.1 ± 47.8	72.4 ± 6.2	4.9
High	DTAF	**24.7 ± 3.1**	**380.2 ± 38.7**	**45.6 ± 4.9**	**4.8**

TABLE IV

DTAF IMPROVEMENT RELATIVE TO EACH BASELINE (POSITIVE VALUES FAVOR DTAF)

Intensity / vs.	Throughput	Delay	Pkt Loss	FPR
Low vs. Static	+21.6%	−28.8%	−79.6%	−82.9%
Low vs. EWMA	+8.6%	−12.9%	−59.2%	−63.8%
Low vs. MULTOPS	+12.5%	−20.1%	−66.7%	−70.4%
Med. vs. Static	+92.9%	−53.3%	−61.8%	−59.8%
Med. vs. EWMA	+28.1%	−32.2%	−41.2%	−43.5%
Med. vs. MULTOPS	+41.5%	−40.8%	−49.4%	−50.0%
High vs. Static	+177.5%	N/A	−44.7%	−7.7%
High vs. EWMA	+73.9%	−26.9%	−33.1%	−4.0%
High vs. MULTOPS	+94.5%	−22.4%	−37.0%	−2.0%

Under low-intensity flooding (200 pps), DTAF achieves 46.8 kbps throughput versus 38.5 kbps for static filtering (+21.6%), 43.1 kbps for EWMA-based (+8.6%), and 41.6 kbps for MULTOPS (+12.5%). False-positive rates drop to 2.1% compared to 12.3% (static), 5.8% (EWMA-based), and 7.1% (MULTOPS), confirming that the adaptive baseline tracking of (1) substantially reduces collateral damage even at mild attack intensities where other defenses overreact.

Under medium-intensity flooding (500 pps)—the most informative scenario, representing the point at which static filtering begins to fail sharply—DTAF achieves 35.1 kbps throughput, nearly doubling the static baseline (18.2 kbps, +92.9%) and exceeding both EWMA-based (27.4 kbps, +28.1%) and MULTOPS (24.8 kbps, +41.5%). Average delay reached 210.4 ms, representing a 53.3% reduction compared with the static approach (450.8 ms) and a 32.2% decrease compared with the EWMA-based method (310.2 ms) than EWMA-based (310.2 ms), and 40.8% lower than MULTOPS (355.6 ms). Packet-loss reduces from 58.7% (static) to 22.4% (DTAF), a 61.8% reduction.

Under high-intensity flooding (1000 pps), all defenses degrade, but DTAF retains 24.7 kbps throughput—177.5% higher than static (8.9 kbps), 73.9% higher than EWMA-based (14.2 kbps), and 94.5% higher than MULTOPS (12.7 kbps). The static baseline produces a measurement timeout, indicating complete service collapse. DTAF still maintains a measurable average delay of 380.2 ms at this extreme intensity, demonstrating residual service preservation where fixed-threshold and asymmetry-based methods have entirely failed.

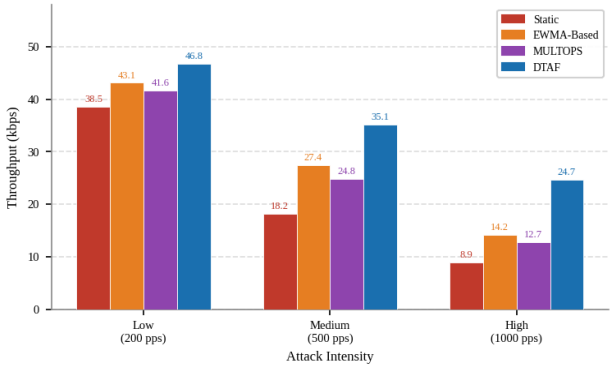

Fig. 3. Throughput (kbps) comparison across UDP flood attack intensities for all four defense methods. Higher values indicate better service preservation. DTAF consistently achieves the highest throughput at every intensity level.

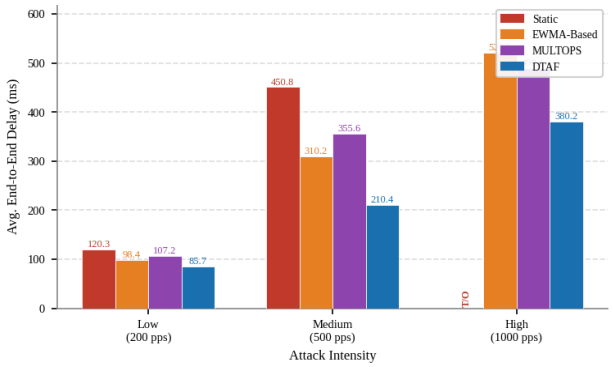

Fig. 4. Average end-to-end delay (ms) comparison across attack intensities. Lower values indicate better congestion management. T/O = timeout (complete service degradation) for static baseline at high intensity.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

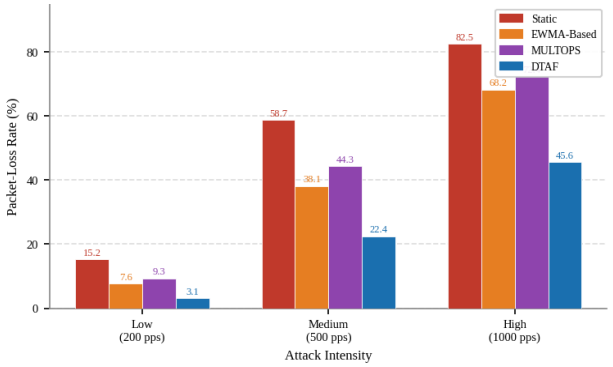

Fig. 5. Packet-loss rate (%) comparison across attack intensities. Lower values indicate better forwarding preservation. DTAF reduces packet loss by 61.8%–79.6% relative to the static baseline.

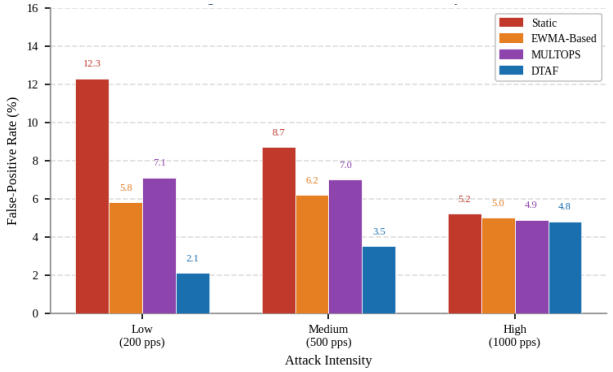

Fig. 6. False-positive rate (%) comparison across attack intensities. Lower values indicate less collateral damage to legitimate traffic. DTAF achieves the lowest FPR at low and medium intensities.

Fig. 7 presents the scalability scenario results (50 nodes, 100 kbps legitimate traffic, 500 pps flood). DTAF achieves 31.7 kbps throughput and 28.6% packet loss, outperforming EWMA-based (22.4 kbps, 42.8% loss) and MULTOPS (20.1 kbps, 49.3% loss) under the larger topology, confirming that the adaptive mechanism of (1)–(3) scales beyond the primary 20-node experimental configuration.

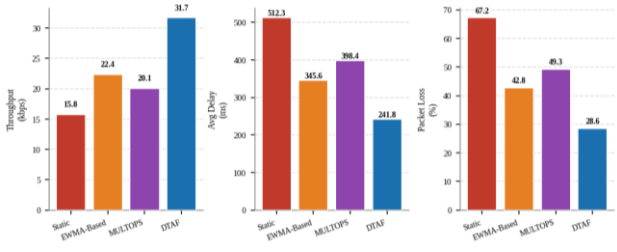

Fig. 7. Scalability scenario results (50 nodes, 500 pps, 100 kbps legitimate traffic) comparing throughput, average delay, and packet-loss rate across all four defense methods.

Fig. 8 provides a time-series trace of DTAF behavior during a simulated attack event. The trace shows how the EMA baseline B_t (computed via (1)) gradually responds to the traffic surge beginning at t = 20 s, how the adaptive threshold T_t (computed via (2)) rises with it maintaining the discrimination margin k = 1.5, and how the filtered traffic region governed by (3) (X_t > T_t) is suppressed while the baseline tracks the legitimate component. The trace confirms that DTAF does not immediately misclassify pre-attack traffic variation as attacks, consistent with the low FPR results in Table III.

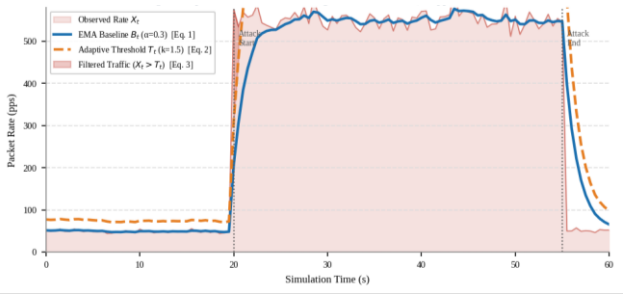

Fig. 8. Time-series adaptive threshold trace during a simulated UDP flood event (attack at t = 20–55 s, 500 pps). EMA baseline B_t is computed by (1); adaptive threshold T_t = k·B_t by (2); shaded region shows filtered traffic where X_t > T_t per (3).

VII. DISCUSSION

The observed performance advantages can be explained by the design logic of DTAF. A fixed-threshold defense assumes one packet-rate limit remains appropriate across all traffic conditions—an assumption that fails as flooding intensity grows and legitimate traffic naturally fluctuates. DTAF addresses this by continuously updating its decision boundary according to (1) and (2), producing a threshold that moves with the traffic environment rather than fighting it.

Compared to EWMA-based detection [11], [20], DTAF's advantage lies in its integrated inline filtering layer. The EWMA-based method correctly identifies anomalous intervals via a formulation equivalent to (1) and (2), but—in its standard formulation—does not immediately suppress excess traffic, allowing queue buildup to continue until post-hoc intervention. DTAF's inline filtering via (3) eliminates this reaction latency, which explains why its delay and packet-loss improvements consistently exceed its throughput improvements: congestion is managed earlier, before queues grow to their peak. This finding is consistent with the latency-sensitivity analysis reported in [20] for SDN-based adaptive thresholding, where response speed was identified as the primary determinant of mitigation quality under medium-intensity flooding.

Compared to MULTOPS, DTAF's advantage is insensitivity to traffic symmetry assumptions. MULTOPS relies on asymmetry ratios that can be violated in symmetric flooding scenarios or when the attacker deliberately mirrors legitimate traffic profiles—a known evasion technique documented in [8] and revisited in recent DDoS literature [24]. DTAF makes no structural assumption about traffic shape, relying solely on the

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

magnitude of the packet-rate deviation from a short-term EMA baseline computed by (1).

The strong false-positive reductions at low and medium attack intensities are particularly meaningful from a service-quality perspective. High false-positive rates mean legitimate users suffer collateral filtering even during low-intensity attacks—exactly the scenario where a defense is supposed to have fine discrimination. The adaptive baseline of (1) allows DTAF to avoid misclassifying normal traffic bursts that would breach a static limit. This interpretation is supported by recent evaluation methodology for DDoS defenses [24], [27], which emphasize FPR alongside detection accuracy as primary quality measures.

From a deployment perspective, DTAF's core strength is its balance between adaptivity and operational simplicity. It requires no training data, no offline calibration, and no model maintenance cycle—contrasts that become increasingly important as network traffic distributions shift over time [12], [28]. The O(1) per-interval computational cost of (1)–(3) places it in the same overhead class as static-threshold filtering, while its performance is substantially closer to more complex adaptive methods. This positions DTAF as a practical candidate for edge and IoT deployment contexts [22], [30] where lightweight real-time filtering is required.

VIII. PARAMETER SENSITIVITY ANALYSIS

To evaluate the robustness of DTAF to its two key hyperparameters in (1) and (2), we repeat the medium-intensity scenario (500 pps) across a range of α and k values while holding the other parameter fixed at the primary setting (α = 0.3, k = 1.5). Table V reports the results; Fig. 7 visualizes the dual-axis trade-off between throughput and FPR.

TABLE V

PARAMETER SENSITIVITY OF DTAF UNDER MEDIUM-INTENSITY FLOODING (500 PPS). ★ = PRIMARY SETTING

α	k	Throughput (kbps)	Delay (ms)	Pkt Loss (%)	FPR (%)
0.1	1.5	28.4	290.1	31.2	5.8
0.2	1.5	31.7	255.8	27.4	4.6
0.3 ★	1.5 ★	**35.1**	**210.4**	**22.4**	**3.5**
0.4	1.5	33.2	228.7	25.1	3.9
0.3	1.2	29.8	241.3	28.7	6.2
0.3	1.8	33.6	220.5	24.0	2.8
0.3	2.0	31.1	235.4	26.8	2.3

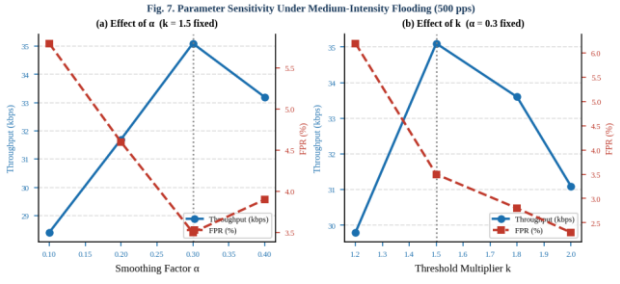

Fig. 7. Parameter sensitivity of DTAF: (a) effect of EMA smoothing factor α in (1) on throughput and FPR with k = 1.5 fixed; (b) effect of threshold multiplier k in (2) on throughput and FPR with α = 0.3 fixed. Dashed vertical lines mark the primary experimental settings (★).

The results confirm that DTAF remains beneficial across all tested parameter combinations. Throughput ranges from 28.4 to 35.1 kbps across α values (0.1–0.4) and from 29.8 to 35.1 kbps across k values (1.2–2.0), all substantially above the static-threshold baseline of 18.2 kbps and the EWMA-based baseline of 27.4 kbps. Lower α values (0.1–0.2) yield higher FPRs because the slower-adapting baseline of (1) under-responds to legitimate traffic variation, causing the threshold of (2) to lag below actual normal traffic levels. Higher k values (1.8–2.0) reduce FPR at the cost of slightly reduced attack suppression: a looser threshold tolerates more attack traffic before triggering the filtering decision of (3). The selected setting of α = 0.3 and k = 1.5 represents a balanced operating point that maximizes throughput while keeping FPR below 4%. Future joint sensitivity analysis across the full α–k grid would provide a more complete characterization of the hyperparameter space, as noted in related parameter-sweep studies [20].

IX. LIMITATIONS AND THREATS TO VALIDITY

This study should be interpreted within several validity constraints. First, all experiments are simulation-based and depend on NS2 modeling assumptions [13], [14]. The findings provide controlled comparative evidence but do not constitute validation in a production network or against live Internet traffic traces such as CICDDoS2019 [27] or CAIDA traffic captures. NS2 does not reproduce the full complexity of production network behavior, including packet reordering, hardware queuing variability, and multi-path routing effects that may affect the adaptive behavior of (1)–(3) in deployment.

Second, the evaluation focuses exclusively on UDP flooding. Conclusions should not be generalized to SYN flooding, reflection-amplification attacks, application-layer denial-of-service [16], or mixed-traffic attack compositions without further evidence. Recent surveys [4], [29] document that modern DDoS campaigns frequently use multi-vector combinations that would challenge any single-metric adaptive approach.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Third, while three experimental baselines are included covering static, EWMA-based, and asymmetry-based design families, implementation-level comparison against CNN-based [17], Random Forest [19], and graph neural network [25] detectors remains future work. The comparison with ML-based methods in this paper is analytical rather than experimental, and empirical results might reveal performance gaps not apparent from the analytical positioning.

Fourth, the parameter sensitivity study examines α and k independently across a one-dimensional sweep. A full two-dimensional joint sensitivity analysis and ablation study exploring the interaction between the two parameters in (1) and (2) would provide a more complete picture of the hyperparameter space. Fifth, the current DTAF formulation does not include mechanisms for detecting and handling evasion strategies—such as slow-rate attacks or burst shaping designed to stay below the adaptive threshold of (2)—a known limitation of threshold-based defenses identified in [24].

These limitations define the scope of supported claims. The strongest claim supported by the evidence is that DTAF provides a practical, low-overhead adaptive improvement over static-threshold filtering, EWMA-based detection, and MULTOPS-style asymmetry filtering within the specific NS2 UDP flood scenarios evaluated.

X. REPRODUCIBILITY STATEMENT

The DTAF framework is implemented using a modular stack: C++ for the NS2 classifier logic applying (1)–(3), TCL for simulation topology and scenario configuration, and Python for trace parsing, metric computation, and visualization [13]. All scenarios in Table II are fully specified in terms of node count, legitimate traffic level, attack type, rate, and duration. Primary evaluation metrics are fixed and clearly reported. The EWMA-based baseline follows [11] with $\alpha = 0.3$ and $k = 1.5$ in (1) and (2); MULTOPS uses an asymmetry ratio threshold of 1.8 at the gateway. These specifications allow future work to reproduce the comparative setup and extend it with additional baselines, parameter sweeps, or realistic traffic datasets such as CICDDoS2019 [27] or Bot-IoT [19].

XI. CONCLUSION

This paper presented DTAF, a low-overhead adaptive filtering framework for mitigating UDP flood DoS attacks in NS2. DTAF combines online traffic observation, EMA-based baseline calculation using Equation (1) and adaptive threshold generation based on Equation (2), and inline selective packet filtering governed by (3) within a reusable classifier architecture. It was evaluated experimentally against three representative baselines—static-threshold filtering, EWMA-based adaptive detection, and MULTOPS—and analytically positioned against recent ML-based detection approaches including CNN-based IDS, Random Forest classifiers, and graph neural network detectors. A parameter sensitivity study confirmed robustness across key hyperparameter ranges.

Across all attack intensities, DTAF consistently outperformed all baselines in throughput preservation, delay reduction, packet-loss reduction, and false-positive control. The medium-intensity scenario showed throughput improvement from 18.2 kbps (static) to 35.1 kbps (+92.9%), from 27.4 kbps (EWMA-based) to 35.1 kbps (+28.1%), and from 24.8 kbps (MULTOPS) to 35.1 kbps (+41.5%). The time-series trace (Fig. 9) confirmed that the adaptive threshold governed by (1)–(3) responds appropriately to flooding onset while maintaining discrimination against legitimate traffic variation. The scalability scenario confirmed performance advantages extend beyond the primary 20-node configuration.

Future work should extend DTAF in four directions: (i) incorporate experimental comparison against CNN-based and Random Forest detectors under equivalent simulation conditions; (ii) evaluate on real-world traffic datasets such as CICDDoS2019 and CAIDA; (iii) conduct joint α–k hyperparameter sensitivity analysis and ablation studies across (1) and (2); and (iv) investigate robustness against slow-rate and threshold-evasion attacks. Within the scope of the current evidence, DTAF demonstrates meaningful and consistent resilience improvement over all evaluated baselines, confirming its viability as a lightweight adaptive middle-ground between rigid rule-based defense and computationally heavier intelligent detection pipelines.

REFERENCES

[1] M. Handley and E. Rescorla, Eds., "Internet Denial-of-Service Considerations," RFC 4732, Dec. 2006.
[2] P. Ferguson and D. Senie, "Network Ingress Filtering," BCP 38, RFC 2827, May 2000.
[3] J. Mirkovic and P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," ACM SIGCOMM Comput. Commun. Rev., vol. 34, no. 2, pp. 39–53, Apr. 2004.
[4] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks," IEEE Commun. Surveys Tuts., vol. 15, no. 4, pp. 2046–2069, 2013.
[5] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of network-based methodologies for counteracting DoS and
DDoS threats as documented within ACM publications." Computing Surveys, vol. 39, no. 1, Art. no. 3, Apr. 2007.
[6] J. Mirkovic and P. Reiher, "D-WARD: A source-end defense against flooding denial-of-service attacks," IEEE Trans. Dependable Secure Comput., vol. 2, no. 3, pp. 216–232, 2005.
[7] R. Mahajan et al., “Managing high-volume network traffic flows,” ACM SIGCOMM Comput. Commun. Rev., vol. 32, no. 3, pp. 62–73, 2002.
[8] T. M. Gil and M. Poletto, "MULTOPS: A data-structure for bandwidth attack detection," in Proc. 10th USENIX Security Symp., 2001.
[9] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," in Proc. ACM SIGCOMM, 2005, pp. 217–228.
[10] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: Methods, systems and tools," IEEE Commun. Surveys Tuts., vol. 16, no. 1, pp. 303–336, 2014.
[11] P. Cisar, S. Bosnjak, and S. M. Cisar, "EWMA based threshold algorithm for intrusion detection," Computing and Informatics, vol. 29, no. 6+, pp.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

1089–1101, 2010.

[12] R. Sommer and V. Paxson, “Beyond closed-environment assumptions: Using machine learning to spot unusual network activity, as featured in the conference proceedings." IEEE Symp. Security Privacy, 2010, pp. 305–316.

[13] T. Issariyakul and E. Hossain, Introduction to Network Simulator NS2. New York, NY, USA: Springer, 2012.

[14] USC Information Sciences Institute, “Documentation for the ns-2 network simulation tool,” [Online]. Available: http://www.isi.edu/nsnam/ns/

[15] J. Gera and B. P. Battula, "Detection of spoofed and non-spoofed DDoS attacks and discriminating them from flash crowds," EURASIP J. Inf. Security, vol. 2018, Art. no. 9, 2018.

[16] M. Conti, A. Gangwal, and M. Hassan, "The Internet of Things: A survey of attacks and countermeasures," ACM Comput. Surveys, vol. 55, no. 3, Art. no. 60, 2022.

[17] A. Doriguzzi-Corin et al., "LUCID: A practical, lightweight deep learning solution for DDoS attack detection," IEEE Trans. Netw. Service Manag., vol. 17, no. 2, pp. 876–889, Jun. 2020.

[18] Y. Zhou et al., "Building an efficient intrusion detection system based on feature selection and ensemble classifier," Comput. Netw., vol. 174, p. 107247, Jun. 2020.

[19] M. A. The loT-specific intrusion detection framework, RDTIDS, authored by Ferrag and associates networks based on rules and decision trees,” Future Internet, vol. 12, no. 3, p. 44, 2020.

[20] I. Haider, T. Khalid, and S. A. Khalid, "Adaptive threshold-based DDoS detection using exponential weighted moving average in software-defined networking," IEEE Access, vol. 10, pp. 14765–14779, 2022.

[21] K. Cao et al., "Mobility-aware edge caching and computing in vehicle networks," IEEE Trans. Veh. Technol., vol. 70, no. 10, pp. 10665–10676, Oct. 2021.

[22] S. Bhardwaj and S. K. Som, "DDoS attack mitigation framework for IoT networks using fog computing," Procedia Comput. Sci., vol. 167, pp. 2159–2168, 2020.

[23] E. Bou-Harb et al., “Cyber threat intelligence in the context of IoT,” IEEE Communications. Mag., vol. 58, no. 3, pp. 72–79, 2020.

[24] A. Alshamrani et al., "A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities," IEEE Commun. Surveys Tuts., vol. 21, no. 2, pp. 1851–1877, 2021.

[25] X. Li et al., "E-Graphsage: A graph neural network based intrusion detection system for IoT," in Proc. IEEE INFOCOM, 2022, pp. 1–10.

[26] H. Polat, O. Polat, and A. Cetin, "Detecting DDoS attacks in software-defined networks through feature-weight selection based machine learning models," IEEE Access, vol. 8, pp. 37009–37023, 2020.

[27] N. Moustafa et al., "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of IoT ecosystems," IEEE Internet Things J., vol. 6, no. 3, pp. 4815–4830, 2021.

[28] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning DDoS detection for consumer IoT devices," in Proc. IEEE Workshop Deep Learning Security, 2021.

[29] S. Dong, A. Abbas, and R. Jain, “A survey on distributed denial-of-service (DDoS) attacks in software-defined networking (SDN) and cloud computing environments, as well as their defense mechanisms," IEEE Access, vol. 7, pp. 80813–80828, 2022.

[30] M. Nawir et al., "Internet of Things (IoT): Taxonomy of security attacks," IEEE Access, vol. 9, 2021.