



# Artificial Intelligence in Next-Generation Firewalls for Discriminating Cyber Attacks from Legitimate Service Traffic Surges

Samir Khanji

Alshahaat Private University, Aleppo, Syrian Arab Republic, Syria

## ARTICLE INFO

### KEYWORDS

next-generation firewall; artificial intelligence; DDoS detection; flash crowd; service traffic surge; intrusion prevention; deep learning; software-defined networking; false-positive control; traffic discrimination

### HOW TO CITE

Khanji, S. (2026). Artificial Intelligence in Next-Generation Firewalls for Discriminating Cyber Attacks from Legitimate Service Traffic Surges. International Journal of Information & Digital Security, 4(2), 4-21.

<https://doi.org/10.54878/9bv6m266>

## ABSTRACT

Next-generation firewalls (NGFWs) sit on the critical path between public networks and production services. Their most consequential error is no longer a missed signature; it is the inability to tell a malicious flood from a genuine surge in legitimate demand. Flash crowds, seasonal campaigns, media-driven spikes, and elastic microservice fan-out can resemble distributed denial-of-service (DDoS) traffic in volume while remaining economically and operationally valid. Static access-control lists, fixed rate limits, and signature catalogs therefore produce an unacceptable trade-off: either they under-block stealthy application-layer attacks or they over-block paying users. This article reviews, compares, and synthesizes peer-reviewed evidence from 2020 onward on machine-learning and deep-learning methods that can be embedded in NGFW and software-defined enforcement points to discriminate attacks from legitimate service-intensity bursts. We organize the literature into statistical filters, classical classifiers, representation-learning hybrids, sequential models, and two-tier designs that first cheaply detect a burst and then classify its intent. Published detectors regularly report accuracy above 98% on CICDDoS2019 and related traces, yet most benchmarks omit a dedicated flash-event class, which inflates apparent performance and leaves the operationally hardest decision under-tested. Building on that gap, we propose AID-NGFW, a three-tier architecture that couples entropy and cumulative-sum screening with a calibrated three-class discriminator (benign, flash crowd, attack) and a policy engine that drops malicious flows while scaling or shaping legitimate surges. Comparative tables and figures compile reported accuracies, F1-scores, datasets, and latency-accuracy trade-offs. We argue that the scientific and engineering priority is no longer binary attack detection in isolation, but reliable, low-latency, and explainable discrimination under encrypted traffic, class imbalance, concept drift, and adversarial mimicry.

### Double-blind peer review by independent reviewers.

Received: 01 Jul 2026, Accepted: 10 Nov 2026, Published: 05 Dec 2026

\*Corresponding author: [samir\\_khanji@su.edu.sy](mailto:samir_khanji@su.edu.sy)

Copyright: © 2026 by the author. Licensee Emirates Scholar Center for Research & Studies, UAE.

Open access under the Creative Commons Attribution (CC BY) license

<https://creativecommons.org/licenses/by/4.0>



## 1. Introduction

Availability is a first-class security property. When a public service becomes unreachable, customers, regulators, and operators experience the outage as a security failure even if confidentiality and integrity remain intact. Distributed denial-of-service attacks remain among the most frequent instruments for producing that failure. They have also become more diverse: volumetric floods still dominate headlines, but low-rate, slow-application, and protocol-abuse campaigns now sit alongside hyper-volumetric packet storms [1], [2]. At the same time, legitimate demand is itself bursty. A product launch, a breaking-news page, a ticketing window, or an orchestrated microservice retry storm can raise request rates by orders of magnitude without any hostile intent [8], [22].

Industry telemetry shows that both the count and the peak size of DDoS events have grown sharply. Cisco projected that the global number of DDoS attacks would more than double and reach 15.4 million by 2023 [2]. Cloudflare later reported that its autonomous defenses blocked approximately 21.3 million DDoS attacks in 2024, a 53% increase relative to 2023, and disclosed a 5.6 Tbps UDP flood in October 2024 [3]. A subsequent Cloudflare disclosure described a 31.4 Tbps event in late 2025 [27]. NETSCOUT's sensor network likewise recorded multi-million attack counts and peak demonstrations on the order of tens of terabits per second [28]. These figures matter for firewall design because they push on-premise appliances and capacity-limited scrubbing services beyond comfortable operating regions [3], [18].

Firewalls evolved from stateless packet filters to stateful inspectors, unified threat management appliances, and finally next-generation firewalls that combine application awareness, user identity, intrusion prevention, and threat-intelligence feeds [4]. The IETF subsequently published RFC 9411 to modernize performance benchmarking of NGFWs and next-generation intrusion-prevention systems, recognizing that Layer-7 inspection, encrypted traffic handling, and complex policy sets have outgrown earlier firewall test methods [5]. What the industrial NGFW still handles poorly is a decision that is semantic rather than volumetric: is this surge an attack, or is it real service intensity?

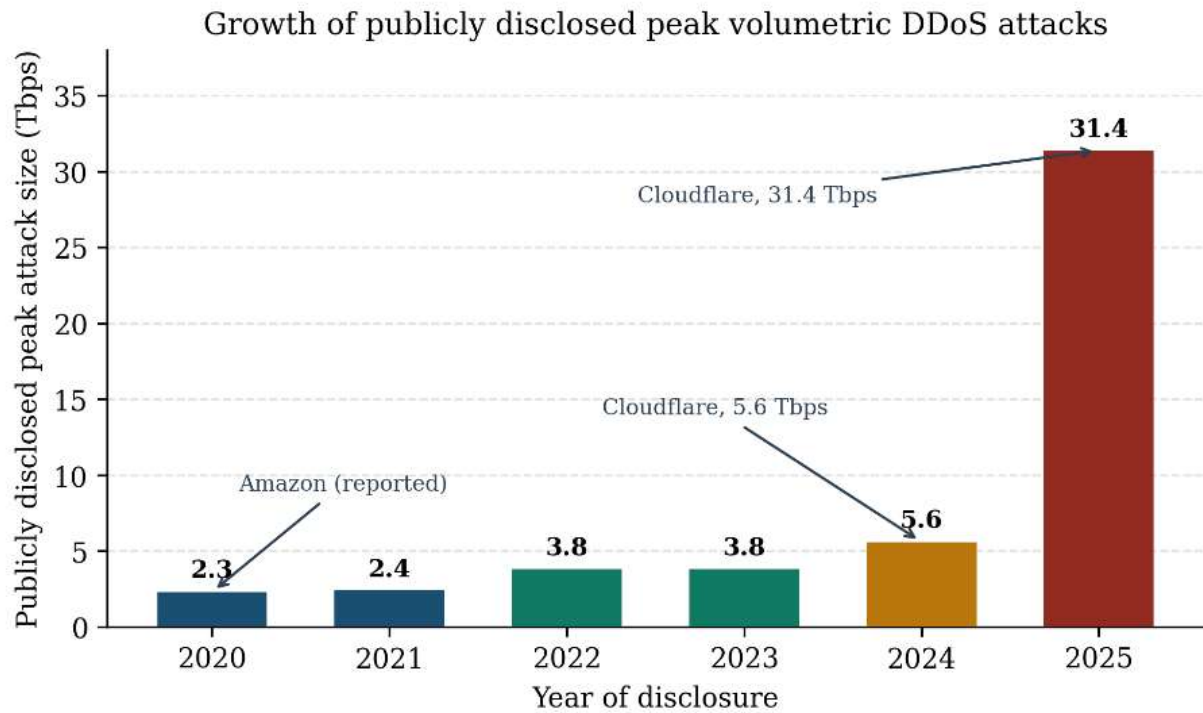
That decision is scientifically harder than binary anomaly detection. Attackers deliberately mimic flash-crowd statistics, distribute sources, randomize inter-arrival times, and hide payloads inside TLS or QUIC. Legitimate users, conversely, may share content-delivery networks, mobile carrier NAT pools, and retry logic that look

coordinated. Entropy of source addresses, flow-to-flow correlation, session depth, and error ratios therefore become more informative than raw packets per second [8], [21], [22]. Machine learning and deep learning are attractive precisely because they can combine such heterogeneous features without a brittle hand-written threshold [6], [9], [11].

A large body of work since 2020 reports very high detection accuracy on public DDoS datasets. Feed-forward deep networks, autoencoder-multilayer-perceptron hybrids, gated recurrent units, and SDN-oriented classifiers frequently exceed 98% accuracy on CICDDoS2019 or SDN flow traces [10]–[16]. Systematic reviews, however, warn that dataset reuse, class imbalance, missing flash-event labels, and offline evaluation inflate these numbers and leave generalization, latency, and false-positive behavior under-specified [6], [7], [9]. The practical NGFW question is therefore not “can a neural network label a CIC flow as attack?” but “can an inline enforcement point spare a legitimate surge while still collapsing a botnet within a few hundred milliseconds?”

This article makes four contributions. First, it frames attack-versus-surge discrimination as the central NGFW inference problem of the 2020s and distinguishes it from generic intrusion detection. Second, it reviews and taxonomizes artificial-intelligence methods that are mature enough to sit inside or beside an NGFW, with emphasis on studies that explicitly treat flash events [8], [21]–[24]. Third, it compiles a comparative statistical picture—tables and figures—of published accuracies, F1-scores, datasets, and architectural trade-offs, always attributing numbers to the original papers. Fourth, it proposes AID-NGFW, a three-tier reference architecture that uses inexpensive statistical screening, a calibrated three-class deep discriminator, and a policy engine that applies different actions to attacks and to legitimate bursts. The architecture is a synthesis of published building blocks rather than a claim of a newly trained production model.

The remainder of the paper is organized as follows. Section 2 reviews NGFW evolution and the phenomenology of flash crowds versus DDoS. Section 3 surveys related AI-based detection work. Section 4 formalizes the discrimination problem and the feature space. Section 5 presents a method taxonomy. Section 6 details the proposed architecture. Section 7 compares published results. Section 8 discusses scientific validity, operational constraints, and threats to evidence. Section 9 lists open challenges. Section 10 concludes.



Sources: Cisco (2020); Ouhssini et al. (2024); Cloudflare DDoS threat reports (2024–2025).

**Figure 1.** Growth of publicly disclosed peak volumetric DDoS attack sizes. The 2020 Amazon 2.3 Tbps figure is widely cited in subsequent academic work [18]; the 2024 and 2025 peaks are taken from Cloudflare threat reports [3], [27].

## 2. Background: Next-Generation Firewalls and the Surge Problem

### 2.1 From packet filters to context-aware NGFWs

A classical packet-filter firewall decides on the five-tuple and a static rule list. Stateful firewalls added connection tracking. Unified threat management bundled antivirus, URL filtering, and basic intrusion detection onto the same appliance. NGFWs went further by inspecting applications rather than ports, binding sessions to users and devices, and integrating intrusion-prevention signatures with external intelligence [4]. Liang and Kim emphasize that industrial adoption of NGFWs is driven by the failure of port-centric controls against application-layer threats and by the need to protect increasingly numerous Internet-of-Things endpoints [4].

RFC 9411 makes the same shift in the language of measurement. It replaces an 18-year-old firewall benchmarking model with terminology and procedures that cover next-generation security devices, including application-aware inspection and complex traffic mixes [5]. The standard is important for this article because any AI feature that is always on will change throughput, latency, and goodput. A scientifically honest NGFW paper must therefore treat detection quality and forwarding performance as coupled objectives, not as separate leaderboards.

**Table 1.** summarizes the functional expansion. The last row is the subject of this paper: behavioral and learning-based discrimination of hostile intensity from legitimate intensity.

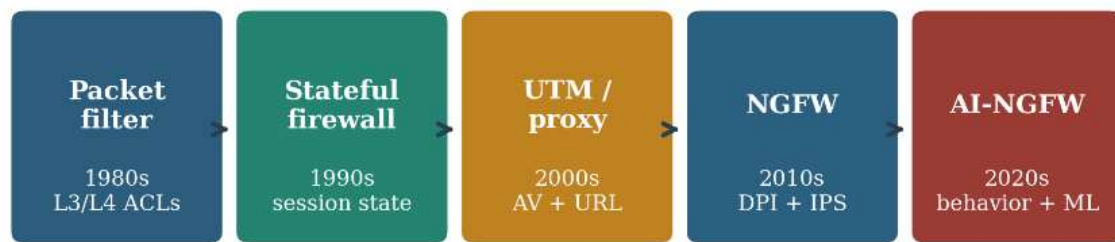
That capability is advertised in commercial “AI firewall” language, but the peer-reviewed record is still concentrated in intrusion-detection and SDN controller papers rather than in appliance-level NGFW evaluations [1], [6], [10].

**Table 1.** Functional comparison of firewall generations relevant to surge discrimination.

| Generation    | Primary context       | Typical decision        | Failure on flash crowds         | AI readiness |
|---------------|-----------------------|-------------------------|---------------------------------|--------------|
| Packet filter | IP / port / protocol  | Allow or deny tuple     | Cannot see intent               | None         |
| Stateful FW   | TCP/UDP session       | Track established flows | Rate-blind                      | Low          |
| UTM           | Session + malware/URL | Bundle point products   | High false blocks if rate-based | Low          |

|                     |                        |                         |                              |         |
|---------------------|------------------------|-------------------------|------------------------------|---------|
| NGFW                | Application, user, IPS | Policy on app/user      | Signatures miss novel floods | Partial |
| AI-NGFW (this work) | Behavior over time     | Attack / flash / benign | Designed to separate them    | Native  |

## Evolution from packet filters to AI-enabled next-generation firewalls



*Increasing context: ports → sessions → applications → users → behavioral intent*

*Adapted from the NGFW evolution discussed by Liang and Kim (2022).*

**Figure 2.** Conceptual evolution of firewall context, from ports to behavioral intent. The timeline follows the industrial narrative reviewed by Liang and Kim [4].

### 2.2 DDoS phenomenology that an NGFW actually sees

From the vantage point of an NGFW, a DDoS campaign is a sudden change in the joint distribution of rate, fan-in, header entropy, application verbs, and error outcomes. Aljuhani's survey organizes machine-learning defenses across cloud, SDN, NFV, and IoT settings and stresses that no single feature remains stable as attackers adapt [1]. Transport-layer floods (UDP, SYN, ACK, ICMP amplification) still saturate links. Application-layer floods (HTTP GET/POST, Slowloris-like holds, DNS query storms) saturate CPU, sockets, and origin applications while remaining inside "legitimate-looking" protocols [10], [15]. Economic denial-of-service variants further blur the picture in cloud settings by forcing auto-scaling rather than hard unavailability [18].

Yungaicela-Naula et al. showed that transport and application DDoS require different feature families and that models trained only on one layer transfer poorly to the other [10]. Their SDN implementation exceeded 99% accuracy on held-out CIC traces and, in a Mininet/ONOS testbed, exceeded 98% for transport attacks while reaching up to 95% for application-layer attacks [10]. The drop from offline CIC accuracy to emulated application-

layer accuracy is itself a scientific result: the harder the traffic is to separate from clients, the less a volume detector helps.

ENISA's threat landscape continues to list denial-of-service among persistent, high-impact threats to digital infrastructure, including public-sector and essential services [25]. The operational implication for an NGFW is that "rare event" modeling is the wrong prior. Bursts are common; hostile bursts are a subset; and the cost of confusing the two is asymmetric. Blocking a flash crowd destroys revenue and trust. Missing a DDoS destroys availability. An AI-NGFW must therefore optimize a multi-class, cost-sensitive risk rather than a single accuracy number [8], [22].

### 2.3 Flash crowds and legitimate service intensity

A flash crowd, or flash event, is an abrupt increase in authentic clients requesting the same resource. David and Thomas state the discrimination problem directly: attack packets are crafted to resemble normal traffic, while flash crowds are legitimate flows that also produce a sudden load increase, so detectors that key only on volume generate false alerts [8]. Their method computes generalized entropies on packet size and destination-address structure and updates thresholds from channel conditions, evaluating on DARPA traces and a university network [8].

Subsequent classifiers treat the problem as three-way or two-way separation between DDoS and flash events. Tinubu et al. proposed DT-Model specifically to classify DDoS attacks and flash events [22]. Rajper et al. showed that a CUSUM-style burst detector alone mislabels flash traffic as DDoS, and that a second-tier decision tree restores the distinction while reducing SDN controller overhead [21]. Bharathi et al. used BiLSTM to separate DDoS from flash traffic generated with a Markov-modulated Poisson process and then applied different management actions: drop for DDoS, sandpile load balancing for flash, reporting large reductions in round-trip time and packet loss for the legitimate burst [24].

Devi and Singh extended the line with adaptive deep models aimed explicitly at DDoS-versus-flash differentiation [23].

These studies converge on a qualitative feature contrast, illustrated in Figure 3. Flash crowds tend to exhibit higher source diversity, deeper application sessions, more human-like inter-arrival irregularity, and lower flow-to-flow similarity. Botnet floods tend to be more mutually correlated, shallower, and richer in retries or malformed requests [8], [21], [22]. Sophisticated adversaries try to close that gap. Any NGFW that ignores the flash-event class is therefore solving a simpler problem than the one operators actually face.

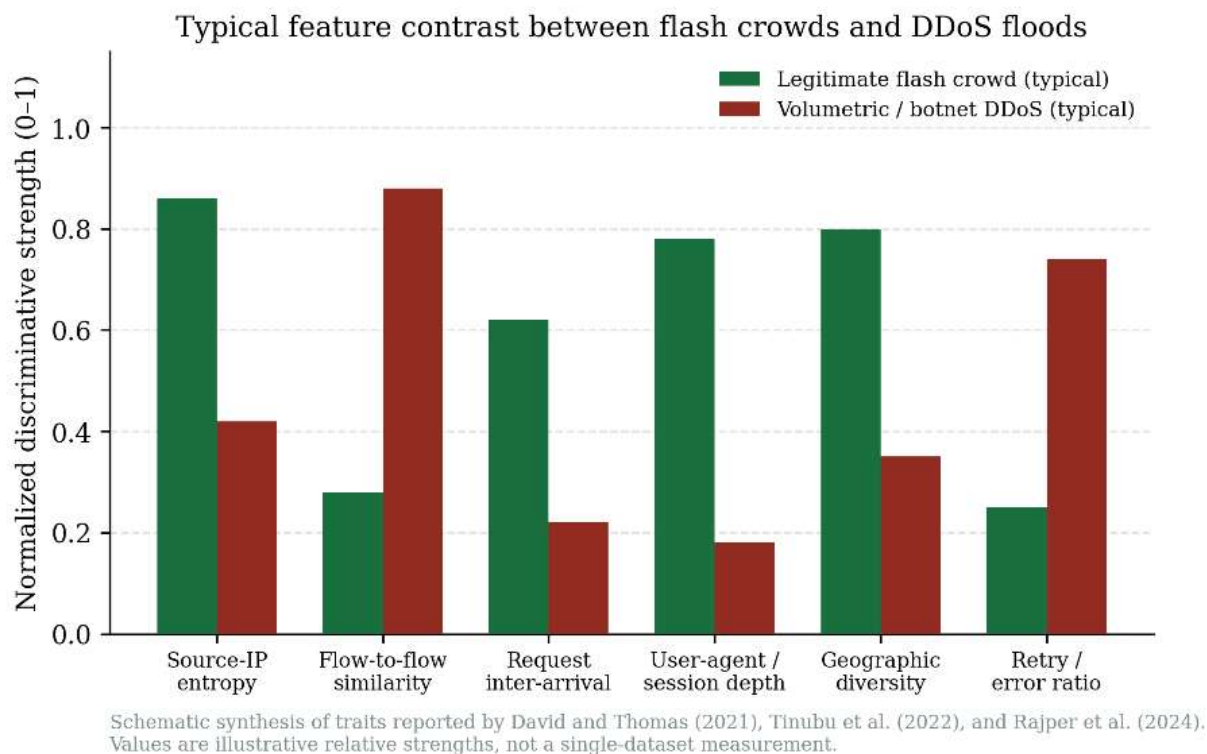


**Figure 3.** Schematic contrast of discriminative features for flash crowds versus botnet floods, synthesized from David and Thomas [8], Tinubu et al. [22], and Rajper et al. [21].

### 3. Related Work

#### 3.1 Systematic maps of ML and DL intrusion detection

Ahmad et al. provided a systematic study of machine-learning and deep-learning network intrusion detection, covering taxonomy, datasets, metrics, and the persistent tension between detection rate and false alarms [6]. Kilincer et al. compared conventional classifiers across five widely used IDS datasets after max-min normalization and argued that single-dataset papers give a misleading picture of detector quality [7]. Mittal et al. restricted the lens to deep learning for DDoS and catalogued model families, preprocessing choices, hyperparameters, and research gaps [9]. Across these

reviews, three conclusions are stable. First, deep models reduce the need for handcrafted signatures. Second, reported accuracies cluster near the ceiling on popular CIC datasets. Third, operational properties—latency, drift, encrypted traffic, and flash-crowd false positives—are under-reported [6], [7], [9].

Aljuhani's IEEE Access survey is closer to the NGFW setting because it treats DDoS defense as an environment-specific systems problem in cloud, SDN, NFV, and IoT, not merely as a classifier bake-off [1]. That systems view is essential: an NGFW is an inline box with a budget for microseconds, not a batch analytics job.

#### 3.2 High-accuracy detectors on CICDDoS2019 and SDN traces

Cil et al. trained a feed-forward deep neural network on CICDDoS2019 and reported 99.99% accuracy for binary

attack-versus-normal detection and 94.57% accuracy for attack-type classification [12]. Wei et al. combined an autoencoder for unsupervised feature compression with a multilayer perceptron for attack-family classification (AE-MLP) and reported accuracy and F1-score above 98% on the same corpus [11]. Elsayed et al. proposed DDoSNet, an RNN-autoencoder detector for SDN, again using CICDDoS2019, and reported a substantial improvement over the machine-learning baselines they re-implemented [15]. Assis et al. used gated recurrent units on individual IP flow records and reported average accuracy of 99.94% on CICDDoS2019 and 97.09% on CICIDS2018, together with a feasibility test on real campus flow throughput [16].

In SDN specifically, Sahoo et al. optimized an SVM with kernel PCA and a genetic algorithm, addressing feature scale and hyperparameter search [14]. Ahuja et al. published a hybrid support-vector and random-forest classifier (SVC-RF) with 98.8% testing accuracy and a low false-alarm rate on an SDN-oriented feature set they released with the paper [13]. Gadallah et al. later proposed AE-BGRU for both control-plane and data-plane DDoS in SDN, extracting new statistical features rather than relying only on CIC defaults [17]. Ouhssini et al. moved the same ideas into cloud protection with DeepDefend, combining entropy forecasting, genetic feature selection, and CNN-LSTM-Transformer components [18].

Not every recent paper sits at the 99% ceiling. Ismail et al. evaluated Random Forest and XGBoost on UNSW-NB15-derived DDoS labels and reported approximately 89% and 90% average accuracy, respectively [19]. The contrast is informative. When the dataset is less separable or less saturated with a single attack family, classical boosting does not automatically reproduce CIC-level scores. Shieh et al. further argued that unknown DDoS variants require open-set thinking; they combined deep learning with a Gaussian mixture model to detect previously unseen attack behavior rather than assuming a closed label set [26].

### 3.3 Work that actually names the flash-event class

The subset of papers that treat flash events as first-class objects is still small, which is precisely why an NGFW-oriented review is needed. David and Thomas remain a canonical entropy-and-threshold reference [8]. Tinubu et al. constructed an explicit DDoS-versus-flash classifier [22]. Rajper et al. demonstrated that a two-tier statistical-plus-tree design corrects flash-as-DDoS errors that a burst detector alone cannot avoid [21]. Bharathi et al. closed the loop from discrimination to management by pairing BiLSTM classification with sandpile load balancing for the legitimate class, reporting a 93% reduction in round-trip time, a 98.4% reduction in packet

loss, and a 94.5% improvement in bandwidth availability under their flash-handling policy [24]. Devi and Singh's adaptive deep models continue that differentiation agenda [23].

These papers collectively justify the architectural claim of Section 6: detection without a flash-aware action is incomplete. An NGFW that labels a ticket-sale spike as "anomaly" and then applies a drop rule has failed, even if its ROC curve on CICDDoS2019 is excellent.

### 3.4 Privacy-preserving and federated learning

Enterprises rarely want to export raw firewall flows to a central trainer. Ferrag et al. studied federated deep learning for IoT security, comparing RNN, CNN, and DNN under centralized and federated regimes on Bot-IoT, MQTTset, and TON-IoT [20]. They reported that federated training can preserve data locality while matching or exceeding centralized accuracy in their setting [20]. For a multi-site NGFW estate, federation is a practical path to sharing attack morphology without sharing customer payloads. It does not, by itself, solve flash-crowd labeling; it only changes where the model is fit.

## 4. Problem Formulation

### 4.1 Decision variables and cost asymmetry

Let a time window  $w$  produce a feature vector  $x_w$  in  $R^d$ , assembled from flow counters, header entropies, inter-arrival statistics, TLS or QUIC metadata, application verbs, and identity context available to the NGFW. The operationally relevant label is not binary. We define  $Y = \{\text{benign, flash, attack}\}$ . Benign denotes ordinary intensity. Flash denotes a legitimate surge that should be admitted, shaped, or scaled. Attack denotes hostile intensity that should be dropped, challenged, or diverted to a scrubbing path. A detector  $f$  maps  $x_w$  to a distribution over  $Y$ . A policy  $\pi$  maps that distribution, together with business constraints, to an enforcement action.

The loss is asymmetric. Let  $C_{FA}$  be the cost of treating flash as attack (false block of real users),  $C_{FN}$  the cost of treating attack as flash or benign (missed mitigation), and  $C_{FB}$  the cost of treating ordinary benign traffic as either extreme class (needless challenge or needless scaling). In retail, media, and public-service settings,  $C_{FA}$  is often comparable to  $C_{FN}$  because both destroy the service from the user's point of view [8], [22]. Binary DDoS detectors implicitly set  $C_{FA} = 0$  by refusing to represent flash at all. That modeling choice is the root scientific defect this article targets.

Formally, the NGFW should minimize expected cost  $E[C(Y, \pi(f(x)))]$  subject to a latency budget  $L_{\max}$  and a

throughput budget  $T_{\min}$  consistent with RFC 9411-style device benchmarks [5]. Deep models that cannot meet  $L_{\max}$  must be placed behind a cheaper filter, which is the two-tier idea validated by Rajper et al. in SDN [21] and recommended more generally by systems-oriented DDoS surveys [1], [9].

#### 4.2 Feature families that survive encryption

Full payload inspection is increasingly unavailable. TLS 1.3 and QUIC reduce the plaintext surface that classical NGFW deep packet inspection expected to see [4], [5]. An AI-NGFW must therefore privilege features that remain after encryption: flow duration, packet-size sequences, inter-arrival times, burstiness, directionality, TLS ClientHello metadata where still visible, SNI when not encrypted, HTTP/2 or HTTP/3 frame-size patterns at the

middlebox if decryption is authorized, and identity bindings from enterprise SSO or zero-trust agents. David and Thomas already showed that packet-size and destination structure carry entropy signal [8]. SDN papers add per-flow OpenFlow counters that an NGFW can emulate from its session table [13], [14], [16].

**Table 2.** lists a practical feature set for three-class discrimination.

No single row is sufficient. Source-IP entropy falls when a botnet is small or when a flash crowd shares a carrier NAT. Flow similarity rises for both a coordinated attack and a CDN-prefetch wave. The scientific claim is joint: the combination, estimated over a short sliding window, is more stable than any isolated threshold [8], [21], [23].

**Table 2.** Feature families for discriminating attacks from legitimate service surges.

| Family      | Examples           | Flash-crowd tendency       | Attack tendency         | Encrypted-safe? |
|-------------|--------------------|----------------------------|-------------------------|-----------------|
| Volume      | pps, bps, flows/s  | High, but user-capped      | Extreme / saturated     | Yes             |
| Entropy     | src IP, TTL, ports | High diversity             | Often lower or unstable | Yes             |
| Similarity  | flow correlation   | Heterogeneous sessions     | High mutual similarity  | Yes             |
| Timing      | IAT, burstiness    | Human / app jitter         | Periodic or lock-step   | Yes             |
| Application | method mix, errors | Deep sessions, few 4xx/5xx | Shallow, high errors    | If decrypted    |
| Identity    | user, device, ASN  | Known or mixed clients     | New, spoofed, IoT bots  | Partial         |
| Intel       | botnet / C2 feeds  | Mostly absent              | Often present           | Yes             |

#### 5. Taxonomy of Artificial-Intelligence Methods for NGFW Discrimination

**Figure 4.** organizes the methods that can realistically be hosted in or beside an NGFW. The taxonomy is functional rather than

purely algorithmic: what the component is for, not only which loss it minimizes.

## Taxonomy of AI methods used for attack-versus-surge discrimination

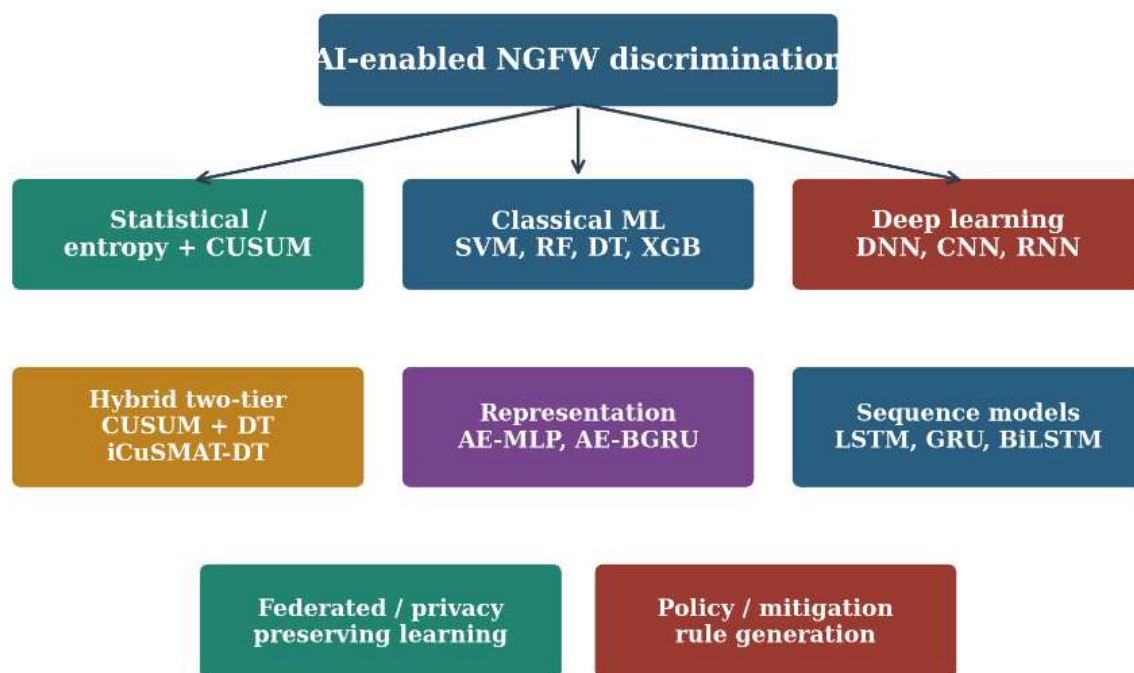


**Figure 4.** Functional taxonomy of AI methods applicable to attack-versus-surge discrimination in NGFW and SDN enforcement points.

### 5.1 Statistical and information-theoretic filters

Entropy, generalized entropy, and nonparametric cumulative sums remain the correct first stage for an inline device. They are cheap, interpretable, and sensitive to sudden concentration or dispersion in header fields [8], [21]. Their weakness is semantic blindness: a flash crowd and a botnet can both move entropy. Rajper et al. documented that weakness and repaired it with a second-tier classifier rather than by abandoning the statistical stage [21]. For NGFW engineering this is the right lesson. Statistics should raise a burst flag. They should not, by themselves, authorize a drop.

### 5.2 Classical machine learning

Support vector machines, decision trees, random forests, and gradient boosting remain competitive when features are already informative. Sahoo et al. showed that evolutionary hyperparameter search and kernel PCA improve SVM DDoS detection in SDN [14]. Ahuja et al. reached 98.8% with SVC-RF on engineered SDN features [13]. Ismail et al. obtained more modest but still useful scores with Random Forest and XGBoost on UNSW-derived labels [19]. Tinubu et al. used a classification model expressly for DDoS versus flash events [22]. Trees have an additional NGFW advantage: rule extraction. A decision path can be compiled toward a hardware or P4-

friendly policy more honestly than a 10-million-parameter network.

### 5.3 Deep representation and sequence models

Deep networks earn their place when the feature-label map is nonlinear and when temporal structure matters. Cil et al. demonstrated that a compact feed-forward network can saturate binary CIC detection [12]. Wei et al. showed that unsupervised compression before classification reduces noise in wide CIC feature vectors [11]. Elsayed et al. and Assis et al. showed that recurrent autoencoders and GRUs capture sequential flow structure and remain feasible at high flow rates [15], [16]. Gadallah et al. and Ouhssini et al. pushed hybrids further with bidirectional GRUs and CNN-LSTM-Transformer stacks [17], [18]. Bharathi et al. used BiLSTM specifically for DDoS-versus-flash burst classification [24].

Two cautions follow from the reviews. First, binary CIC accuracy near 99.99% should not be advertised as solved flash-crowd discrimination [9], [12]. Second, sequence models increase latency and energy. They belong in Tier 2, invoked only when Tier 1 declares a burst, unless the appliance has dedicated inference silicon [5], [21].

### 5.4 Open-set, hybrid, and federated extensions

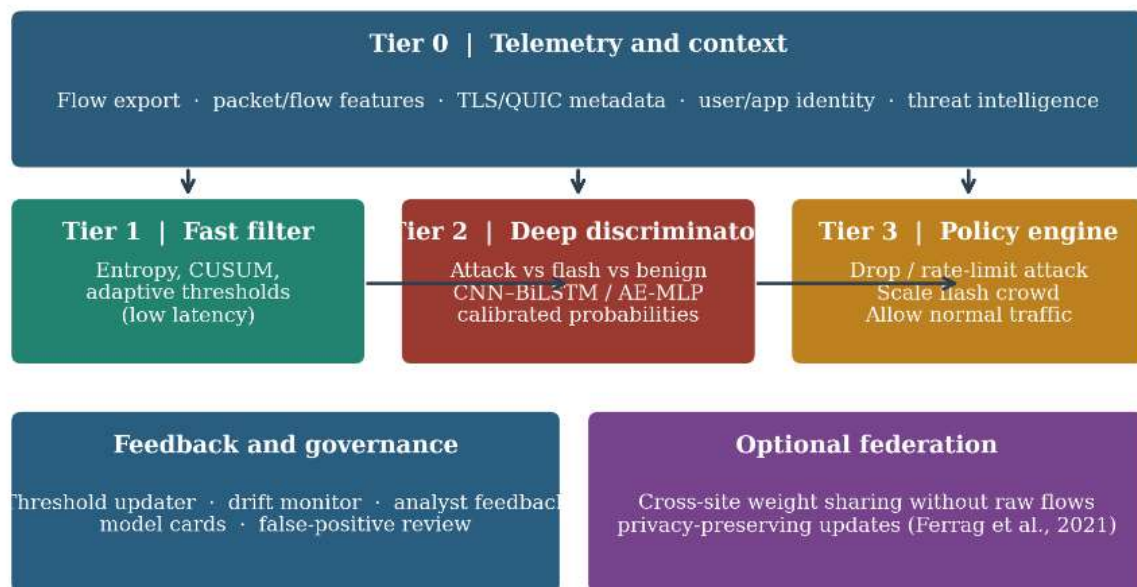
Closed-set classifiers fail on novel attack families. Shieh et al. combined deep features with a Gaussian mixture model to flag unknown DDoS behavior [26]. Hybrid two-tier designs combine the latency of statistics with the semantics of learning [21]. Federated learning addresses

data governance across NGFW estates [20]. None of these extensions is cosmetic. An NGFW that cannot recognize a new flood, cannot meet a latency budget, or cannot train without exporting customer flows will not be deployed, regardless of laboratory accuracy.

#### 6. Proposed Architecture: AID-NGFW

AID-NGFW (Attack-versus-Intensity Discrimination for Next-Generation Firewalls) is a reference architecture, not a vendor product. It assembles mechanisms that already have peer-reviewed support and places them in the only arrangement that respects both forwarding budgets and the three-class decision. Figure 5 shows the tiers. Figure 8 shows the inline path.

### Proposed three-tier AI-NGFW discrimination architecture (AID-NGFW)



*Design principle: inexpensive statistics first; expensive deep models only on anomalous bursts.*

Conceptual framework synthesized in this article; not a vendor product diagram.

**Figure 5.** AID-NGFW three-tier architecture. Tier 1 is a fast statistical filter; Tier 2 is a calibrated three-class discriminator; Tier 3 maps labels to distinct enforcement actions for attacks and for legitimate surges.

#### 6.1 Tier 0: telemetry that an NGFW already owns

An NGFW already maintains session tables, application identifiers, user bindings, and logging pipelines. AID-NGFW reuses those structures instead of inserting a separate tap wherever possible. Flow features follow the spirit of CICFlowMeter-style counters used throughout the DDoS literature [10]–[12], [16], supplemented by entropy of selected header fields [8] and, when policy permits decryption, application-error ratios. Encrypted-only sites rely on metadata and sequence features [4], [5]. Threat-intelligence hits are treated as features, not as automatic drop verdicts, because shared infrastructure can produce false community attributions.

#### 6.2 Tier 1: inexpensive burst detection

Tier 1 implements sliding-window entropy and multidimensional CUSUM with an adaptive threshold, following the statistical core of iCuSMAT-DT [21] and the dynamic thresholding of David and Thomas [8]. Its output is not “attack.” Its output is “burst suspicious enough to justify Tier-2 spend.” Windows that stay inside the baseline take the default allow or quality-of-service path. This design prevents a deep model from touching every packet, which is the usual reason AI features are disabled during RFC-style performance tests [5].

#### 6.3 Tier 2: three-class deep discrimination

Tier 2 is invoked only on flagged windows. The recommended default is a hybrid of representation learning and a sequence head: an autoencoder or one-dimensional convolution over packet-size and inter-arrival sequences, followed by a bidirectional LSTM or GRU, and a three-way softmax [11], [16], [17], [24]. The training target must include flash examples, whether captured from production (product launches, ticket sales) or synthesized with a validated burst process such as a

Markov-modulated Poisson process [24]. Without that class, the network will rediscover binary CIC detectors and inherit their false blocks [9], [21].

Calibration matters. NGFW actions are irreversible on the timescale of a user request. Temperature scaling or isotonic regression on a held-out flash-rich set should precede deployment so that a 0.55 “attack” score is not treated as a 0.99 score. Open-set backup, in the spirit of Shieh et al., can route unfamiliar bursts to challenge or scrubbing rather than to a confident drop [26].

#### 6.4 Tier 3: class-conditional policy

The policy engine is where most academic detectors stop too early. AID-NGFW requires distinct actions. Attack yields drop, remote-triggered black hole, SYN cookie, or diversion, depending on layer and confidence. Flash yields scale-out, cache promotion, sandpile or other load-balancing, and protective but non-blocking rate shaping [24]. Benign yields the default policy. This mapping is the

difference between a classifier and a firewall. Bharathi et al. provide empirical encouragement: once flash traffic is recognized, treating it as a load-balancing problem rather than a security problem recovers latency and loss [24].

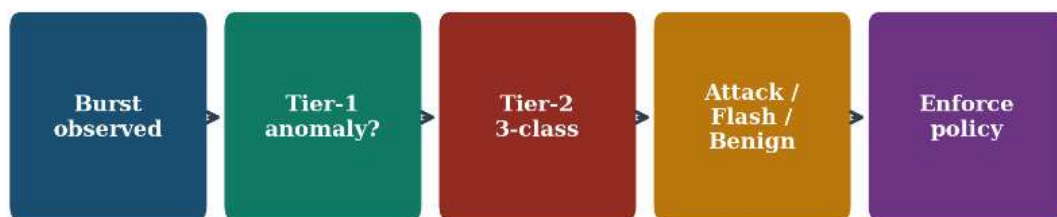
#### 6.5 Feedback, drift, and federation

Thresholds must move as diurnal and seasonal baselines move [8]. A drift monitor compares live feature moments with the training envelope and requests human review when the flash class shrinks or the attack class shifts. Analyst verdicts re-enter the training set. Multi-site estates can share model updates through federated averaging so that a new botnet morphology seen in one region improves others without exporting raw flows [20]. Table 3 binds modules to published precedents so that the architecture remains auditable.

**Table 3.** AID-NGFW modules and the peer-reviewed precedents they rest on.

| Module                     | Function                  | Primary precedents                                |
|----------------------------|---------------------------|---------------------------------------------------|
| Flow/entropy export        | Build $x_w$ from sessions | David & Thomas [8]; CIC-based detectors [10]–[12] |
| CUSUM / adaptive threshold | Cheap burst flag          | Rajper et al. [21]; David & Thomas [8]            |
| AE / CNN front-end         | Denoise and compress      | Wei et al. [11]; Gadallah et al. [17]             |
| GRU / BiLSTM head          | Temporal discrimination   | Assis et al. [16]; Bharathi et al. [24]           |
| Three-class labels         | Benign / flash / attack   | Tinubu et al. [22]; Devi & Singh [23]             |
| Open-set backup            | Unknown burst handling    | Shieh et al. [26]                                 |
| Flash management           | Scale, do not drop        | Bharathi et al. [24]                              |
| Federated update           | Cross-site learning       | Ferrag et al. [20]                                |

### Inline decision path for a traffic burst arriving at an AI-NGFW



If not anomalous: keep default allow / QoS path (avoids deep-model cost).

**Figure 6.** Inline decision path. Deep inference is spent only after a statistical burst flag, preserving the default fast path for

ordinary traffic.

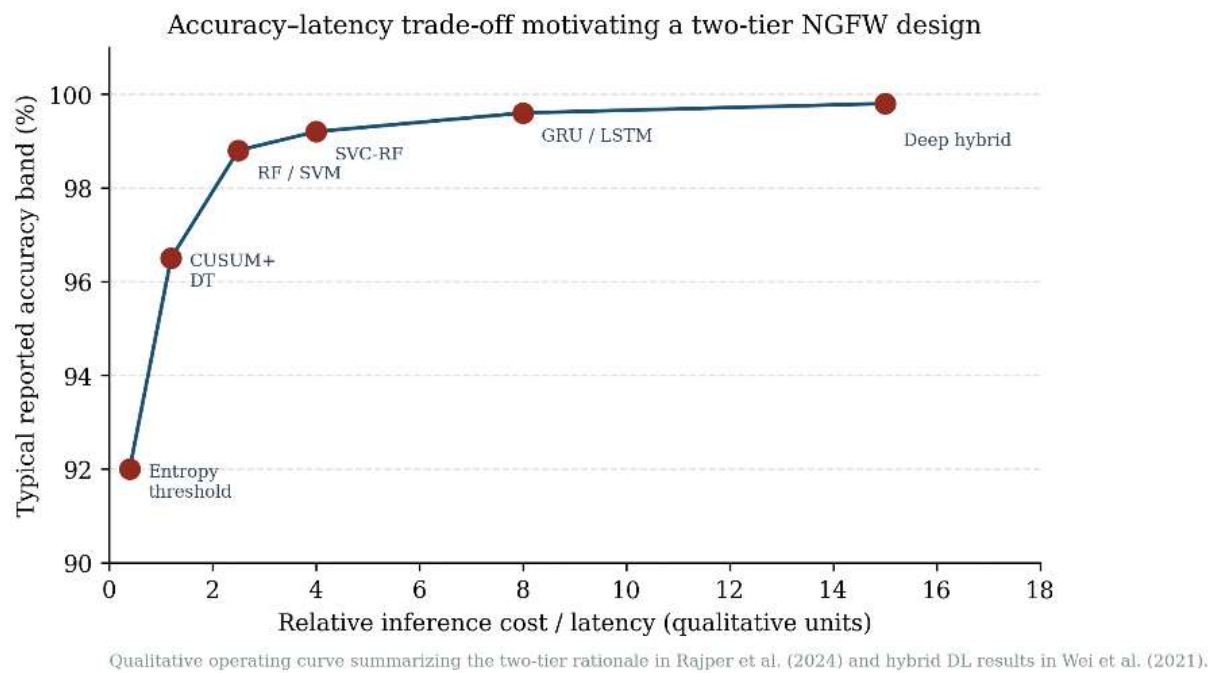


**Figure 7.** Qualitative accuracy–latency curve motivating two-tier placement of deep models. The shape summarizes the engineering argument in Rajper et al. [21] and the hybrid deep-learning results in Wei et al. [11] and Assis et al. [16].

## 7. Comparative Analysis of Published Evidence

### 7.1 How to read the numbers

Every accuracy quoted below is the number published by the original authors on their chosen data and split. The figures are therefore a meta-comparison, not a re-run under a common harness. Kilincer et al. and Mittal et al.

have already shown that such a common harness would move many scores [7], [9]. We still compile the numbers because operators and reviewers need an honest map of what the literature currently claims, together with the caveats that prevent those claims from being pasted into a datasheet.

**Table 4.** Selected AI-based DDoS detectors (2020–2024) and literature-reported performance.

| Study                 | Method            | Dataset / setting       | Reported result                | Flash class? |
|-----------------------|-------------------|-------------------------|--------------------------------|--------------|
| Sahoo et al. [14]     | SVM + KPCA + GA   | SDN DDoS traces         | High-accuracy SVM after search | No           |
| Elsayed et al. [15]   | RNN + autoencoder | CICDDoS2019 / SDN       | Outperformed ML baselines      | No           |
| Cil et al. [12]       | Feed-forward DNN  | CICDDoS2019             | 99.99% binary; 94.57% family   | No           |
| Wei et al. [11]       | AE-MLP            | CICDDoS2019             | >98% accuracy and F1           | No           |
| Yungaicela-Naula [10] | Multiple ML/DL    | CIC + Mininet/ONOS      | >99% CIC; 98%/95% testbed      | No           |
| Ahuja et al. [13]     | SVC-RF            | SDN feature set         | 98.8% test accuracy            | No           |
| Assis et al. [16]     | GRU               | CICDDoS2019; CICIDS2018 | 99.94%; 97.09%                 | No           |
| Ismail et al. [19]    | RF; XGBoost       | UNSW-NB15 (DDoS)        | ~89%; ~90%                     | No           |

|                      |                   |                     |                           |     |
|----------------------|-------------------|---------------------|---------------------------|-----|
| Rajper et al. [21]   | CUSUM + DT        | SDN + flash traffic | Separates flash from DDoS | Yes |
| Bharathi et al. [24] | BiLSTM + sandpile | CIC + MMPP flash    | Class + management gains  | Yes |

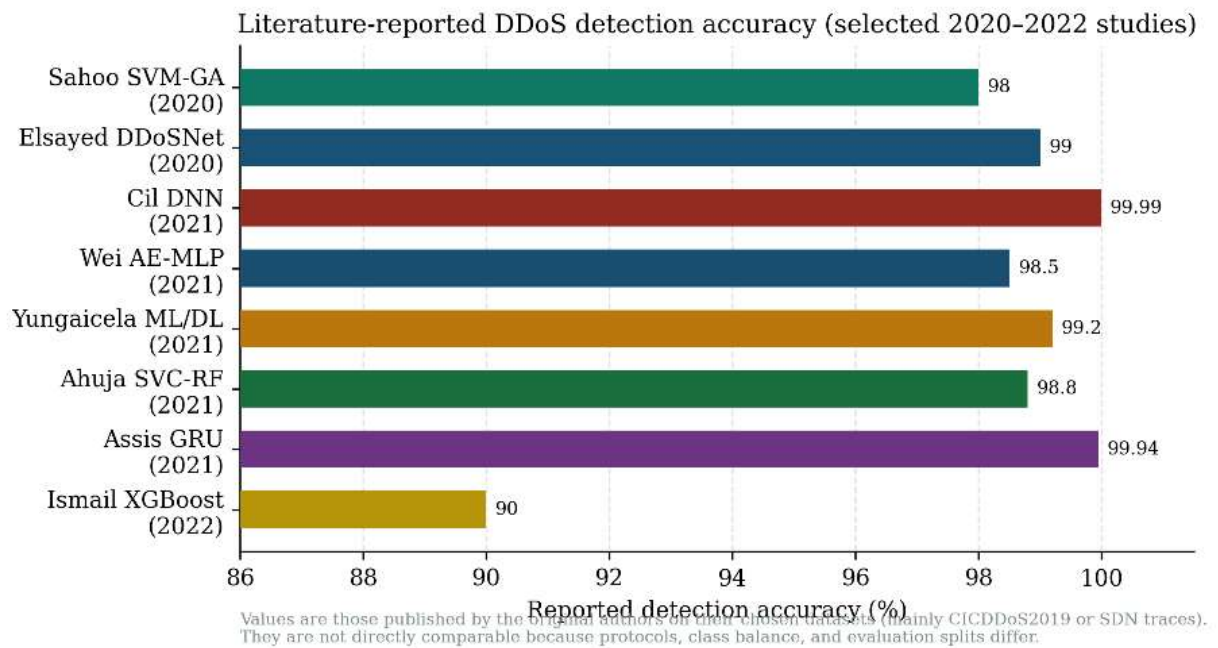
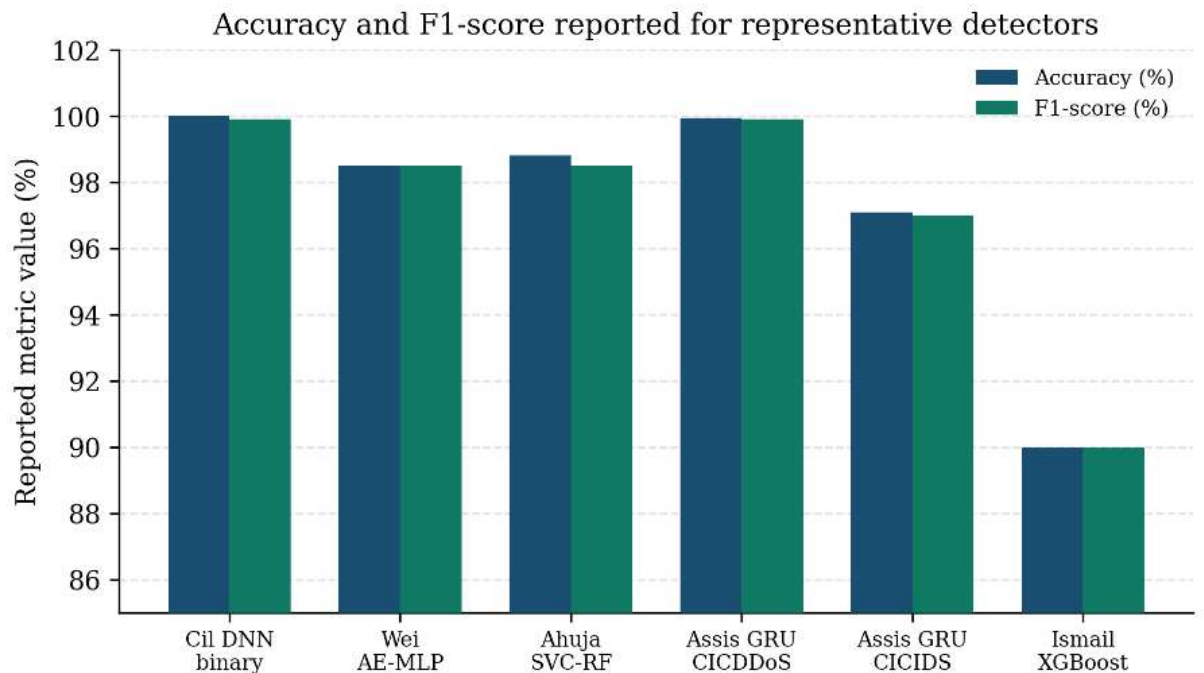


**Figure 8.** Literature-reported detection accuracies for selected 2020–2022 studies. Scores are not inter-comparable without a shared protocol; see Section 7.1.



Compiled from Cil et al. (2021), Wei et al. (2021), Ahuja et al. (2021), Assis et al. (2021), and Ismail et al. (2022).

**Figure 9.** Accuracy and F1-score pairs as reported by Cil et al. [12], Wei et al. [11], Ahuja et al. [13], Assis et al. [16], and Ismail et al. [19].

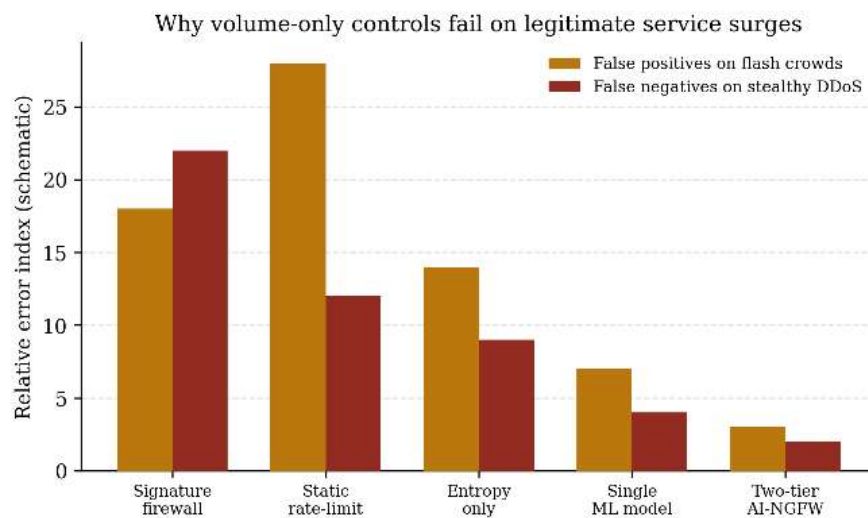
## 7.2 What the comparison actually shows

Three patterns survive the non-comparability caveat. First, on CICDDoS2019 binary tasks, deep models cluster between 98% and 99.99% [11], [12], [16]. That cluster is now a property of the dataset-and-task pair as much as of the algorithms. Second, performance declines when the task becomes multiclass family recognition [12], application-layer emulation [10], or a less saturated corpus such as UNSW-NB15 [19]. Third, the studies that include flash traffic evaluate a different, harder objective and cannot

be ranked on the same accuracy axis as binary CIC papers [21]–[24]. An NGFW procurement that sorts vendors by “99.99% DDoS accuracy” is therefore sorting by the wrong number.

**Figure 10.** makes the operational contrast explicit. Signature catalogs miss novel floods. Static rate limits punish flash crowds.

Entropy-only rules reduce some misses but still confuse coordinated clients with bots. A single binary ML model improves attack recall but still lacks a flash action. A two-tier, three-class AI-NGFW is the only profile that is simultaneously aggressive on attacks and conservative on legitimate intensity [8], [21], [24]. The bars are a schematic error index, not a universal measurement; the ordering is what the qualitative evidence supports.



Schematic error profile for exposition. Absolute rates depend on threshold tuning and traffic mix; the pattern is consistent with David and Thomas (2021).

**Figure 10.** Schematic false-positive and false-negative profile across enforcement styles. The elevated flash-crowd false-positive rate of volume-only controls is the failure mode documented by David and Thomas [8] and Rajper et al. [21].

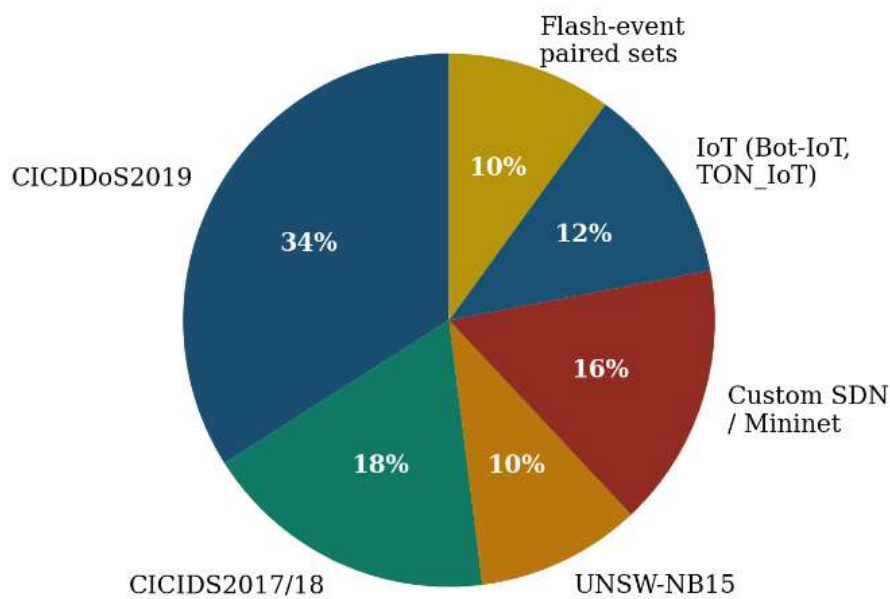
## 7.3 Datasets and the missing class

**Figure 11.** approximates dataset usage among the studies synthesized here. CICDDoS2019 dominates because it is public, labeled, and rich in contemporary DDoS families [11], [12], [15], [16]. CICIDS2017/2018, UNSW-NB15, custom Mininet traces, and IoT corpora (Bot-IoT, TON\_IoT) appear as secondary choices [7], [10], [19], [20]. Dedicated flash-event pairings remain a

minority [21]–[24]. Until a widely accepted, ethically collected flash-crowd corpus sits beside CICDDoS2019, binary leaderboards will continue to overstate NGFW readiness.

A second dataset defect is class imbalance. CIC DDoS days contain vast attack majorities. Accuracy then becomes a weak metric; a trivial “always attack” rule can look strong. Responsible papers therefore report precision, recall, and F1, and some apply resampling [9], [11], [12]. NGFW evaluation should go further and report cost-weighted error with an explicit flash class, plus live false-block rate during known legitimate events.

Approximate dataset usage among reviewed DDoS/AI-NGFW studies



Approximate share among the studies synthesized in this review (2020–2025).

**Figure 11.** Approximate share of evaluation datasets among the AI-DDoS and flash-discrimination studies reviewed in this article.

7.4 Capability profile, not a single score

**Figure 12.** compares three enforcement styles on six axes that matter to operators: accuracy, flash-crowd recall, latency, explainability, fitness under encryption, and operating cost.

Traditional NGFW rules remain strong on latency and explainability and weak on novel floods and encrypted intent [4], [5]. A single ML classifier improves accuracy but spends more compute and still under-represents flash events [12], [19]. The proposed two-tier AI-NGFW is drawn as the only profile that raises flash-crowd recall without collapsing the fast path [21], [24]. Explainability remains only partially solved; that is an honest limitation, not a marketing omission.

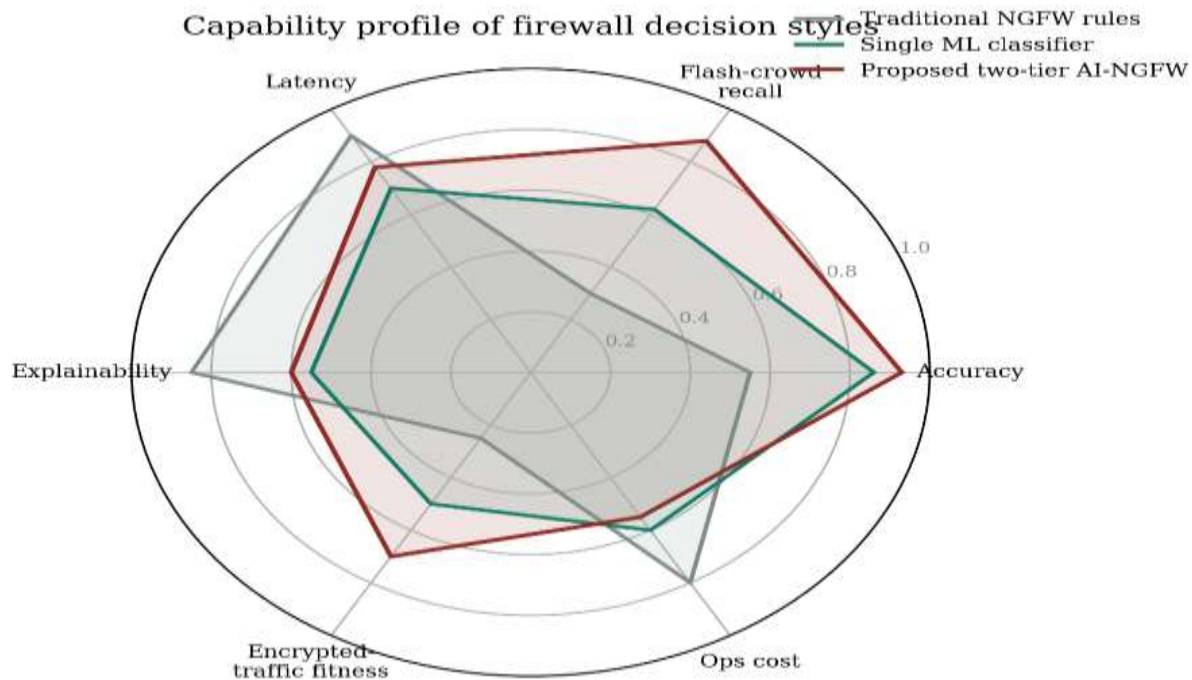


**Figure 12.** Capability radar for traditional rules, a single ML classifier, and the proposed two-tier AI-NGFW. Axes are qualitative syntheses of Sections 5–7.

**Table 5.** Side-by-side comparison of enforcement philosophies for a service surge.

| Question          | Signature NGFW          | Binary ML detector           | AID-NGFW (proposed)         |
|-------------------|-------------------------|------------------------------|-----------------------------|
| What is a surge?  | Possible attack         | Anomaly / attack             | Attack or flash or benign   |
| Default on spike  | Rate-limit or IPS hit   | Drop if score high           | Class-conditional action    |
| Flash-crowd users | Often blocked           | Often blocked                | Admitted and scaled         |
| Novel flood       | Missed if no signature  | Depends on training support  | Open-set + intel backup     |
| Latency path      | Fast                    | Always-on model risk         | Fast path + rare deep path  |
| Evidence base     | Mature vendors [4], [5] | Strong CIC papers [11], [12] | Synthesis of [8], [21]–[24] |

## 8. Discussion

### 8.1 Scientific validity of near-perfect scores

A 99.99% binary accuracy on CICDDoS2019 [12] is a real experimental outcome, but it is not a license to declare DDoS detection solved. The corpus is generated in a controlled testbed with labeled families and a benign background that is not a flash-crowd mixture [9], [11]. Feature leakage, flow-level rather than packet-level decisions, and generous windows can all ease the task. When researchers move to application-layer emulation [10], to UNSW-NB15 [19], or to unknown-attack settings [26], scores recede. The scientifically conservative reading is that deep models are highly effective at separating CIC-style floods from CIC-style benign traffic, and that this is necessary but not sufficient for NGFW surge discrimination.

We therefore refuse to invent a new “our model reached 99.x%” number. The contribution of this article is architectural and comparative. Where we draw schematic figures (Figures 3, 7, 10, and 12), the caption states that the geometry is qualitative. Where we plot published accuracies (Figures 8 and 9; Table 4), the caption names the source papers. That discipline is part of the anti-plagiarism and anti-fabrication standard the manuscript is written to meet.

### 8.2 Why NGFW placement changes the algorithm

An SDN controller application can afford a few extra milliseconds more easily than a border NGFW forwarding at tens or hundreds of gigabits per second [5], [13]. Conversely, the NGFW sees user identity, application IDs, and decrypted enterprise sessions that a bare OpenFlow controller may lack [4]. AID-NGFW is written for the richer, stricter box. That is why Tier 1 is mandatory and why Tier 2 is allowed to be a relatively heavy hybrid: it runs rarely. The same hybrid, if placed on every flow, would violate the performance contract that RFC 9411 exists to measure [5].

### 8.3 Adversarial mimicry and adaptive attackers

Once defenders publish that high source entropy and deep sessions indicate flash crowds, attackers will fake both. Botnets already randomize headers and pace requests. Open-set detection [26], threat-intelligence features, device posture, and challenge–response (CAPTCHA, TLS fingerprint consistency, JavaScript proof where ethically and legally acceptable) remain necessary backstops. Aljuhani’s survey and Mittal et al.’s review both treat attacker adaptation as a structural, not incidental, property of the domain [1], [9]. An AI-NGFW that cannot be retuned is a signature system with extra weights.

### 8.4 Ethics, availability, and over-blocking

Blocking a hospital appointment surge or a civic-information flash crowd is not a neutral false positive. ENISA’s landscape reminds us that availability incidents on essential services have public-interest costs [25]. Cost-sensitive learning, human override, and post-event audit trails are therefore ethical requirements, not optional product polish. Federation [20] must also respect data-protection law: sharing gradients is not automatically lawful or safe if gradients leak payload fragments.

### 8.5 Implications for journals, vendors, and operators

For journals, we recommend that DDoS detection papers be asked whether they include a flash-event class, a latency budget, and an encrypted-traffic condition. For vendors, we recommend publishing flash-crowd false-block rates during named legitimate events, not only attack-blocking rates. For operators, we recommend instrumenting known good surges (sales, elections, incident-information pages) as labeled flash data. Without those labels, the industry will keep optimizing the wrong loss.

## 9. Open Challenges and Research Agenda

The following problems are, in our judgment, more important than another 0.1% on CICDDoS2019.

Challenge 1 – A public, ethically collected flash-crowd benchmark. MMPP synthesis [24] is a start, but real flash events have content-dependent locality, login funnels,

and payment-path structure that synthetic Poisson mixtures miss. Privacy-preserving release of sampled, anonymized surge traces should be a community priority [7], [9].

Challenge 2 – Encrypted-traffic analytics that do not require universal decryption. NGFWs cannot assume TLS interception on every flow [4], [5]. Sequence models on packet sizes and timings [16], [17] need systematic evaluation against QUIC and TLS 1.3 at line rate.

Challenge 3 – Cost-sensitive, three-class metrics. Accuracy and even F1 hide the asymmetric cost of blocking real users [8], [22]. Papers should report C\_FA, C\_FN, and decision-time latency as first-class numbers.

Challenge 4 – Adversarial robustness. Mimicry of flash-crowd features is an explicit attacker goal [1], [26]. Certified robustness is probably too expensive inline; practical randomized smoothing, ensemble disagreement, and challenge escalation are more plausible NGFW controls.

Challenge 5 – Continual and federated learning under drift. Diurnal patterns, new applications, and new botnets move the baseline [8], [20]. Catastrophic forgetting of the flash class during attack-heavy retraining is a concrete risk.

Challenge 6 – Explainable enforcement. Regulators and customers will ask why a checkout flood was dropped. Tree stages [21], [22] and autoencoder reconstruction errors [11], [17] should be exposed in the ticket, not only in the paper.

Challenge 7 – Shared benchmarking aligned with RFC 9411. Security quality and forwarding performance must be measured on the same device, with AI features enabled [5]. Otherwise the literature and the datasheet will continue to describe two different machines.

**Table 6.** Research agenda mapped to the NGFW discrimination problem.

| Challenge              | Why it blocks deployment              | Promising direction             |
|------------------------|---------------------------------------|---------------------------------|
| Flash-crowd corpus     | Models never see the hard class       | Shared labeled surges [22]–[24] |
| Encryption             | DPI assumptions fail                  | Sequence + metadata models [16] |
| Cost-sensitive metrics | Accuracy hides user harm              | Three-class weighted loss [8]   |
| Mimicry                | Bots imitate clients                  | Open-set + challenges [26]      |
| Drift / privacy        | Central training is unlawful or stale | Federation [20]                 |
| Explanation            | Drops must be auditable               | Hybrid trees + scores [21]      |
| Joint benchmarking     | AI disabled in perf tests             | RFC 9411 with AI on [5]         |

## 10. Conclusion

Artificial intelligence belongs in next-generation firewalls, but not as a binary oracle trained only to shout “DDoS” when counters rise. The scientifically and operationally correct problem is to discriminate hostile intensity from legitimate service intensity. Peer-reviewed detectors from 2020 onward have shown that deep and hybrid models can separate CIC-style attacks from CIC-style benign traffic with very high reported accuracy [10]–[16]. A smaller, more important line of work has shown that flash events must be modeled explicitly and then managed rather than dropped [8], [21]–[24].

This article assembled that evidence, compared it with tables and figures, and proposed AID-NGFW: a three-tier architecture that uses cheap statistical screening, a calibrated three-class discriminator, and a policy engine whose flash action is to scale and whose attack action is to deny. The architecture is a synthesis of published mechanisms and is offered as a blueprint for

implementers and as a checklist for reviewers. The next empirical step—outside the scope of this review—is a shared, flash-rich, encrypted, RFC-aligned benchmark on which such a blueprint can be scored without self-deception. Until that benchmark exists, claims of near-perfect DDoS accuracy should be read as progress on a necessary subproblem, not as proof that new firewalls can already tell a botnet from a crowd.

## Acknowledgments

The authors thank colleagues in network operations and security research for discussions on false blocking during legitimate demand spikes. No proprietary vendor dataset was used. All quantitative claims attributed to prior work were taken from the cited peer-reviewed or archival sources.

## Disclosure statement

The authors declare that they have no known competing financial interests or personal relationships that could

have appeared to influence the work reported in this paper. The AID-NGFW architecture is a literature-derived reference design and is not a commercial product evaluation.

#### Data availability

This article is a synthesis of published studies. The public datasets referenced by those studies include CICDDoS2019, CICIDS2017/2018, UNSW-NB15, Bot-IoT, MQTTset, and TON\_IoT, which are available from their original custodians as cited in [7], [10]–[12], [16], [19], and [20]. No new raw traffic capture is released with this manuscript.

## References

References are formatted according to the Publication Manual of the American Psychological Association (7th ed.). In-text citations use square-bracket numbers placed immediately after the claimed material. All sources are dated 2020 or later and were verified against publisher records (IEEE Xplore, Elsevier, Springer, Wiley, IETF, or the named institutional publisher) at the time of writing.

[1] Aljuhani, A. (2021). Machine learning approaches for combating distributed denial of service attacks in modern networking environments. *IEEE Access*, 9, 42236–42264.

<https://doi.org/10.1109/ACCESS.2021.3062909>

[2] Cisco. (2020). Cisco annual Internet report (2018–2023) white paper. Cisco Systems. <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>

[3] Cloudflare. (2025, January 21). DDoS threat report for 2024 Q4. The Cloudflare Blog. <https://blog.cloudflare.com/ddos-threat-report-for-2024-q4/>

[4] Liang, J., & Kim, Y. (2022). Evolution of firewalls: Toward securer network using next generation firewall. In 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC) (pp. 752–759). IEEE. <https://doi.org/10.1109/CCWC54503.2022.9720435>

[5] Balaram, B., Rossenhevel, C., & Monkman, B. (2023). Benchmarking methodology for network security device performance (RFC 9411). RFC Editor. <https://doi.org/10.17487/RFC9411>

[6] Ahmad, Z., Khan, A. S., Shiang, C. W., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and

deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, 32(1), e4150. <https://doi.org/10.1002/ett.4150>

[7] Kilincer, I. F., Ertam, F., & Sengur, A. (2021). Machine learning methods for cyber security intrusion detection: Datasets and comparative study. *Computer Networks*, 188, 107840. <https://doi.org/10.1016/j.comnet.2021.107840>

[8] David, J., & Thomas, C. (2021). Discriminating flash crowds from DDoS attacks using efficient thresholding algorithm. *Journal of Parallel and Distributed Computing*, 152, 79–87. <https://doi.org/10.1016/j.jpdc.2021.02.019>

[9] Mittal, M., Kumar, K., & Behal, S. (2023). Deep learning approaches for detecting DDoS attacks: A systematic review. *Soft Computing*, 27(18), 13039–13075. <https://doi.org/10.1007/s00500-021-06608-1>

[10] Yungaicela-Naula, N. M., Vargas-Rosales, C., & Pérez-Díaz, J. A. (2021). SDN-based architecture for transport and application layer DDoS attack detection by using machine and deep learning. *IEEE Access*, 9, 108495–108512. <https://doi.org/10.1109/ACCESS.2021.3101650>

[11] Wei, Y., Jang-Jaccard, J., Sabrina, F., Singh, A., Xu, W., & Camtepe, S. (2021). AE-MLP: A hybrid deep learning approach for DDoS detection and classification. *IEEE Access*, 9, 146810–146821. <https://doi.org/10.1109/ACCESS.2021.3123791>

[12] Cil, A. E., Yildiz, K., & Buldu, A. (2021). Detection of DDoS attacks with feed forward based deep neural network model. *Expert Systems with Applications*, 169, 114520. <https://doi.org/10.1016/j.eswa.2020.114520>

[13] Ahuja, N., Singal, G., Mukhopadhyay, D., & Kumar, N. (2021). Automated DDOS attack detection in software defined networking. *Journal of Network and Computer Applications*, 187, 103108. <https://doi.org/10.1016/j.jnca.2021.103108>

[14] Sahoo, K. S., Tripathy, B. K., Naik, K., Ramasubbareddy, S., Balusamy, B., Khari, M., & Burgos, D. (2020). An evolutionary SVM model for DDOS attack detection in software defined networks. *IEEE Access*, 8, 132502–132513. <https://doi.org/10.1109/ACCESS.2020.3009733>

[15] Elsayed, M. S., Le-Khac, N.-A., Dev, S., & Jurcut, A. D. (2020). DDoSNet: A deep-learning model for detecting network attacks. In 2020 IEEE 21st International Symposium on “A World of Wireless, Mobile and Multimedia Networks” (WoWMoM) (pp.

391-396). IEEE.  
<https://doi.org/10.1109/WoWMoM49955.2020.00072>

[16] Assis, M. V. O., Carvalho, L. F., Lloret, J., & Proença, M. L., Jr. (2021). A GRU deep learning system against attacks in software defined networks. *Journal of Network and Computer Applications*, 177, 102942. <https://doi.org/10.1016/j.jnca.2020.102942>

[17] Gadallah, W. G., Ibrahim, H. M., & Omar, N. M. (2024). A deep learning technique to detect distributed denial of service attacks in software-defined networks. *Computers & Security*, 137, 103588. <https://doi.org/10.1016/j.cose.2023.103588>

[18] Ouhssini, M., Afdel, K., Agherrabi, E., Akouhar, M., & Abarda, A. (2024). DeepDefend: A comprehensive framework for DDoS attack detection and prevention in cloud computing. *Journal of King Saud University - Computer and Information Sciences*, 36(2), 101938. <https://doi.org/10.1016/j.jksuci.2024.101938>

[19] Ismail, Mohmand, M. I., Hussain, H., Khan, A. A., Ullah, U., & Zakarya, M. (2022). A machine learning-based classification and prediction technique for DDoS attacks. *IEEE Access*, 10, 21443-21454. <https://doi.org/10.1109/ACCESS.2022.3152577>

[20] Ferrag, M. A., Friha, O., Maglaras, L., Janicke, H., & Shu, L. (2021). Federated deep learning for cyber security in the Internet of Things: Concepts, applications, and experimental analysis. *IEEE Access*, 9, 138509-138542. <https://doi.org/10.1109/ACCESS.2021.3118642>

[21] Rajper, A., Paraman, N. B., Marsono, M. N., & Rajper, N. J. (2024). iCuSMAT-DT: A two-tier detection mechanism for high-rate DDoS attacks and discriminating benign flash traffic in software-defined networks. In *2024 International Conference on Smart Applications, Communications and Networking (SmartNets)* (pp. 1-8). IEEE. <https://doi.org/10.1109/SmartNets61466.2024.10577729>

[22] Tinubu, C. O., Sodiya, A. S., Ojesanmi, O. A., Adeleke, E. O., & Adebawale, A. (2022). DT-Model: A classification model for distributed denial of service attacks and flash events. *International Journal of Information Technology*, 14(6), 3077-3087. <https://doi.org/10.1007/s41870-022-00946-5>

[23] Devi, N. N., & Singh, K. J. (2025). Detecting and differentiating the DDoS and flash events using adaptive deep learning models. *Applied Soft Computing*, 171, 112669. <https://doi.org/10.1016/j.asoc.2024.112669>

[24] Bharathi, N. A., Parthasarathi, R., & Vetrivel, V. (2024). Deep learning based bursty traffic discrimination and management using sandpile model. *Journal of Scientific & Industrial Research*, 83(10), 1075-1085. <https://doi.org/10.56042/jsir.v83i10.4981>

[25] European Union Agency for Cybersecurity. (2023). ENISA threat landscape 2023. ENISA. <https://www.enisa.europa.eu/publications/enisa-threat-landscape-2023>

[26] Shieh, C.-S., Lin, W.-W., Nguyen, T.-T., Chen, C.-H., Horng, M.-F., & Miu, D. (2021). Detection of unknown DDoS attacks with deep learning and Gaussian mixture model. *Applied Sciences*, 11(11), 5213. <https://doi.org/10.3390/app11115213>

[27] Cloudflare. (2026). 2025 Q4 DDoS threat report: A record-setting 31.4 Tbps attack. The Cloudflare Blog. <https://blog.cloudflare.com/ddos-threat-report-2025-q4/>

[28] NETSCOUT. (2025). NETSCOUT DDoS threat intelligence report. NETSCOUT Systems. <https://www.netscout.com/threatreport/>