A Two-Stage Hybrid Rule-Based and XGBoost Machine Learning Intrusion Detection and Prevention System for DoS/DDoS Mitigation in Software-Defined Networks

Dr. Vikram Dhiman[1], Prof. (Dr.) Shashi Kant Gupta[2]
[1,2] Department of Computer Science and Engineering, Lincoln University College, Petaling Jaya, Selangor-47301, Malaysia
Email: pdf.vdhiman@lincoln.edu.my, shashigupta@lincoln.edu.my

Abstract: Continuing Denial of Service (DoS/DDoS) flooding attacks consume network and host computing power while software-defined network (SDN) controllers that centralize traffic monitoring remain subject to saturating packet-in floods, victim misclassification and static-threshold attack detection logic. Proposed solution: this paper develops and validates an effective two-stage hybrid Intrusion Detection and Prevention System (IDPS) running within a Ryu OpenFlow 1.3 controller, comprising an adaptive-threshold rule based heuristics stage (Stage 1) and calibrated XGBoost classifier trained using an eleven-dimensional protocol-behavior feature set (Stage 2) along with an asymmetric victim-shielding, confidence adaptive TTL, exponentially weighted moving average (EWMA) features smoothing, isotonic probability calibration and dual safety-gated confidence fusion decision rule. Key results: In sixty-three attack simulation scenarios using eleven different types of attacks at four intensity levels, the proposed IDPS achieves around 89% scenario-level detection rate, sub-second latency at medium and high intensity levels and no false positive enforcement of victim hosts; the remaining missed true positives are found to be caused by a structural issue, the Protocol Counter Bypass problem at burst mode hardware flow-table offloading. Applications: The architecture can be immediately applied to enterprise, campus and data-centre SDN applications where fast, self-adaptive DoS/DDoS defence is required.
Keywords: Software-Defined Networking; DDoS detection; intrusion prevention system; XGBoost; OpenFlow

Introduction

Volumetric flooding attacks remain among the most disruptive categories of network security incidents, and their scale has grown sharply with the availability of botnets, reflection, and amplification techniques, with recorded DDoS incidents exceeding terabit-per-second throughput in recent years. Traditional perimeter defences such as dedicated firewalls and signature-based intrusion sensors encode detection logic statically into firmware, making them slow to adapt to novel attack signatures and largely blind to the fine-grained, per-flow behaviour needed to mitigate individual attacking sources while preserving service to legitimate users.

Software-Defined Networking (SDN) decouples the control plane from the data plane, giving a logically centralised controller a global view of flow-level statistics and the ability to install, modify, or withdraw forwarding rules in real time. This programmability is attractive for building an intrusion detection and prevention system (IDPS) directly into the controller, as traffic statistics can be analyzed in software and mitigation can be pushed to switches without manual reconfiguration. However, the same centralization also brings with it new failure modes: the OpenFlow packet-in channel exposing new flows to the controller can be saturated itself by an attacker, the controller has no application-layer semantics to distinguish bursty legitimate traffic from floods, and machine-learning classifiers trained offline can silently degrade if the feature pipeline used at inference time does not exactly mirror the one used in training. In this article we present the design, implementation and experimental evaluation of a two-stage hybrid IDPS that solves these problems by combining fast deterministic heuristics with a calibrated

learning-based classifier. The approach considers victim-attacker discrimination as a first-class structural component of the detection pipeline, rather than a post-hoc correction. The problem statement that guides this work, research gaps identified, motivation, and proposed solution are described in the following sections.

Problem Statement

The central problem addressed in this work is the reliable, low-latency detection and mitigation of DoS and DDoS attacks across heterogeneous protocol vectors (ICMP, SYN, UDP, RST, FIN, ACK, SYN-ACK, PSH-ACK, URG, UDP large-packet, and Ping-of-Death) and across a wide dynamic range of traffic intensities, within an SDN controller, while simultaneously (i) avoiding saturation of the OpenFlow packet-in control channel, (ii) preventing victim hosts under attack from being erroneously flagged as attackers, (iii) guaranteeing that machine-learning features computed during training are numerically identical to those computed at inference time, and (iv) sustaining detection accuracy at both very low and very high (burst) traffic intensities without manual threshold retuning for each deployment environment.

Motivation

Three observations motivate the proposed architecture. First, purely rule-based IDPS deployments are environment-specific: a threshold tuned for a campus network produces excessive false positives in a data-centre network and is trivially evaded by an attacker who keeps traffic marginally below the configured ceiling. Second, purely machine-learning IDPS deployments are slow to respond at the onset of an attack because classifiers need a meaningful number of observed packets before producing a stable estimate, and they are vulnerable to distribution shift between training and operational traffic. Third, the published hybrid literature integrates rule-based and ML stages loosely, seldom enforcing explicit sequencing or handoff semantics, and almost never builds victim protection into the detection loop itself, so that victim enforcement errors occur during the interval between detection and correction. These observations motivate a tightly sequenced two-stage design in which fast heuristics triage unambiguous floods, victim identification is interleaved before any enforcement, and a calibrated classifier is invoked only on the residual suspect population.

Research Gaps

- Gap 1 — Loose stage integration: Today's hybrid SDN-IDPS systems rarely integrate a fast heuristic stage and a calibrated ML stage into a single pipeline with a sequential order and explicit handoff semantics between stages.
- Gap 2 — Post-hoc victim correction: victim-attacker discrimination is typically implemented as a filter before enforcement, not as a structural constraint that blocks enforcement against victims.
- Gap 3 – Mismatch of feature-extraction code between training and deployment: the feature-extraction code used during model training is rarely verified to be numerically identical to the feature-extraction code used during inference, causing silent production performance degradation.
- Gap 4 – Cold-start false positives: newly joined hosts generating legitimate bursty traffic are seldom equipped with an explicit grace mechanism in adaptive-threshold mechanisms leading to spurious enforcement.
- Gap 5 — Uncalibrated ML confidence: tree-based ensemble classifiers such as XGBoost are known to produce poorly calibrated probability outputs, yet most published SDN-IDPS works use raw, uncalibrated scores directly for binary enforcement decisions.

- Gap 6 — Protocol counter sparsity at scale: OpenFlow packet-in visibility degrades at high attack intensity because hardware flow-table offload bypasses the controller for established flows, a limitation rarely acknowledged or quantified in prior SDN-IDPS evaluations.

Open Research Area

Beyond the gaps closed in this work, several areas remain open for further research: extending detection to application-layer DDoS attacks (HTTP flood, Slowloris) that do not manifest as volumetric anomalies at the network layer; resolving the Protocol Counter Bypass Problem through hybrid packet-in/statistics-polling architectures without re-introducing control-channel overhead; enabling online/incremental learning so the ML stage adapts to drifting traffic distributions without full offline retraining; and designing federated multi-controller architectures that share attacker and victim intelligence across regional SDN domains without centralising raw traffic statistics, which raises open questions of privacy-preserving aggregation and consensus under partial network visibility.

Related work

Early work exploring SDN centralization for security purposes followed shortly after OpenFlow's introduction. Shin et al. [2] demonstrated that a Floodlight-based controller could detect and mitigate ARP spoofing and ping flooding by monitoring packet-in rates and applying drop rules, but the system relied entirely on static rate thresholds with no capacity to distinguish traffic types or sources. Mousavi and St-Hilaire [3] proposed an early-detection mechanism for SYN flooding by monitoring the SYN-to-ACK ratio at the controller using statistical change-point detection, achieving low latency for high-intensity floods but not extending to UDP, ICMP, or composite vectors. Braga et al. [4] were among the first to apply machine learning to SDN traffic analysis using a self-organizing map trained on OpenFlow flow statistics, achieving low false-positive flooding detection in simulation but without considering control-plane overhead; Kokila et al. [5] subsequently applied support vector machines to similar features with improved accuracy on laboratory-scale datasets.

Ensemble and boosted-tree methods have consistently outperformed single classifiers on network intrusion benchmarks. Tang et al. [6] applied deep learning to raw SDN flow statistics and showed that feature sets lacking domain-informed engineering can achieve high recall at the cost of elevated false positives, underscoring the importance of protocol-aware feature construction. The CICDDoS2019 benchmark released by the Canadian Institute for Cybersecurity, evaluated by Sharafaldin et al. [7], established that Random Forest and XGBoost achieve the highest F1 scores among standard classifiers, and its CICFlowMeter-derived feature set directly informed the flow-statistical features used in the present work. Classifier output calibration has gained relatively lesser attention; according to Niculescu-Mizil and Caruana [8], boosted trees generate badly calibrated probability estimates in comparison to logistic regression, but isotonic-regression post-processing greatly enhances the calibration process, which is particularly significant in making IDPS enforcement threshold decisions.

Victim-attacker discrimination has also been explored as a different sub-problem. Wang et al. [9] introduced the concept of flow asymmetry metric in order to determine whether the host under consideration is under attack or not, and it has been realized through post-processing filter on top of classifier output instead of integrating it into the detection process. In addition, Karmakar et al. [10] extended the victim identification process to distributed systems by using collaborative victim distress score computed from multiple sources of attacks. The process of adaptive threshold setting to deal with non-stationary traffic baselines was first proposed by Lakhina et al. [11], who proposed an anomaly detection algorithm based on EWMA z-score approach for traffic volume baselines, which forms the basis of Adaptive Threshold used in this research.

Taken together, the survey indicates that the existing literature has examined rule-based detection, ML-based detection, victim-asymmetry metrics, and adaptive thresholding as largely separate threads, with comparatively few works integrating all of them into a single sequentially ordered controller pipeline, and almost none explicitly verifying feature parity between training and inference or quantifying the effect of hardware flow-table offload on controller-side protocol counters.

Table 1 summarises, for each surveyed line of work, the principal gap and the methodology used.

Summary of Literature (Gaps and Methodology)

Table 1. Summary of literature: identified gaps and methodologies used in prior work

Author(s)	**Gap (single word)**	**Methodology Used**
Shin et al. [2]	Rigidity	Static packet-in rate thresholds on Floodlight controller
Mousavi & St-Hilaire [3]	Coverage	SYN/ACK ratio change-point detection
Braga et al. [4]	Overhead	Self-organising map (SOM) on OpenFlow flow statistics
Kokila et al. [5]	Scale	Support vector machine on flow-level features
Tang et al. [6]	Precision	Deep learning on raw SDN flow statistics
Sharafaldin et al. [7]	Benchmark	Random Forest / XGBoost on CICDDoS2019
Niculescu-Mizil & Caruana [8]	Calibration	Isotonic regression post-processing of tree outputs
Wang et al. [9]	Post-hoc	Bidirectional flow-asymmetry post-filter
Karmakar et al. [10]	Aggregation	Collaborative victim-distress scoring
Lakhina et al. [11]	Drift	EWMA z-score traffic anomaly baselining
Xiao et al. [12]	Cold-start	Grace-window threshold deferral
DDoSDetector / AVANT-GUARD [13]	Brittleness	Fixed SYN-rate ceiling on packet-in channel
CNN-based SDN-IDS works	Latency	Convolutional networks on raw packet bytes
Generic hybrid SDN-IDPS (prior art)	Integration	Loosely coupled rule-then-classifier handoff
This work	Addressed	Two-stage rule + calibrated XGBoost with victim shielding

Comparison with Previously Published Work

Table 2 situates the proposed system against fifteen representative lines of prior work along the dimensions most relevant to SDN-based DoS/DDoS detection: whether detection is rule-based, ML-based, or hybrid; whether victim-attacker discrimination is addressed; whether ML output calibration is performed; whether feature-parity between training and inference is verified; and whether a cold-start mechanism is provided.

Table 2. Comparison of the proposed system with previously published work

Ref.	Approach	Victim Shielding	ML Calibration	Cold-Start Handling	Feature Parity Verified
[2]	Rule-based (Floodlight)	No	N/A	No	N/A
[3]	Statistical change-point	No	N/A	No	N/A
[4]	SOM (ML-only)	No	No	No	No
[5]	SVM (ML-only)	No	No	No	No
[6]	Deep learning on raw flows	No	No	No	No
[7]	RF / XGBoost benchmark	No	No	No	No
[8]	Calibration study (general ML)	N/A	Yes	N/A	N/A
[9]	Flow-asymmetry post-filter	Post-hoc	No	No	No
[10]	Collaborative victim score	Partial	No	No	No
[11]	EWMA anomaly baselining	No	No	No	No
[12]	Grace-window cold start	No	No	Yes	No
[13]	AVANT-GUARD style rule IDPS	No	No	No	N/A
[14]	Generic hybrid SDN-IDPS	Post-hoc / partial	No	Rarely	Rarely
[15]	CNN-based SDN-IDS	No	No	No	No

Ref.	Approach	Victim Shielding	ML Calibration	Cold-Start Handling	Feature Parity Verified
This work	Two-stage hybrid (rules + XGBoost)	Structural, pre-enforcement	Yes (isotonic)	Yes (grace window)	Yes (shared function)

Detailed Comparison with DoSGuard (Li et al., Sensors 2022 [26])

DoSGuard [26] is a closely related, representative baseline: a lightweight controller-embedded, hardware-free SDN DoS defence system combining a host-switch mapping monitor with an SVM classifier on Floodlight/OpenFlow 1.3. Because it shares the same high-level philosophy as the proposed IDPS (no extra hardware, pure controller-side enforcement, two-stage detection), it is the most direct published baseline for articulating the advances of the proposed system. Tables 3–7 provide the complete dimension-by-dimension comparison.

Table 3. Architectural and Design Comparison: Proposed SDN-IDPS vs. DoSGuard [26]

Dimension	Proposed SDN-IDPS (This Work)	DoSGuard [26] (Li et al., 2022)
Controller / OpenFlow	Ryu / OpenFlow 1.3	Floodlight v1.2 / OpenFlow 1.3
Additional hardware	None	None
Stage-1 anomaly gate	EWMA adaptive thresholds (mean + kσ) on ICMP/SYN/UDP rates; z-score for other vectors; outlier freeze	Switch-host mapping consistency (MAC=AIM, IP=AII, port growth=AIP)
Stage-2 ML classifier	XGBoost + isotonic-regression calibration + dual safety gate	SVM (uncalibrated)
Feature set (ML stage)	11-dim: avg pkt size, flow dispersion, TCP/UDP/SYN/ICMP/NULL/XMAS ratios, up-down ratio, dst entropy, src-port entropy	6-tuple: RPM, RFE, APF, ADF, ESIA, PGR
Probability calibration	Yes — isotonic regression (Brier score optimised)	No
Victim shielding	Structural pre-enforcement MDS with confidence-adaptive TTL; 0 victim FP enforcements across 63 scenarios	None — entire host blocked (acknowledged limitation)
Mitigation granularity	Flow-level: targeted OpenFlow drop rules per malicious source; victims exempted via MDS	Host-level: single Flow-Mod drop rule per compromised host

Dimension	Proposed SDN-IDPS (This Work)	DoSGuard [26] (Li et al., 2022)
Feature parity (train vs. inference)	Yes — single shared extraction function, training and inference identical	Not verified or reported
Protocol Counter Bypass	Identified, named, quantified (BUG-FIX-1) — primary cause of BURST-intensity misses	Not addressed or acknowledged

Table 4. Attack Scenario Coverage: Proposed SDN-IDPS vs. DoSGuard [26]

Attack Type	Proposed SDN-IDPS	DoSGuard [26]
Volumetric DoS (ICMP/SYN/UDP flood)	Yes	Yes
IP / MAC address spoofing	Yes	Yes (AIM / AII)
Port randomisation attacks	Yes	Yes (AIP)
TCP flag-flood variants (RST/FIN/ACK/PSH-ACK/URG/SYN-ACK)	Yes (6 vectors)	Not evaluated
Ping of Death / UDP large-packet	Yes	Not evaluated
Multi-source DDoS	Yes (19 DDoS scenarios)	Not evaluated
Victim host FP enforcements	0 across all 63 scenarios	Not measured

Detailed Comparison with DDoSBlocker (Sinha et al., Computer Networks 2025 [27])

DDoSBlocker [27] is a recent Q1-journal baseline (Computer Networks, Elsevier, 2025) that advances beyond DoSGuard by introducing a time-based address mapping method and a triggering-based Random Forest classifier, implemented on Floodlight/OpenFlow 1.0. It is designed with DoSGuard [26] and DoSDefender for benchmarking with better FPR (0% versus 9.8% of DoSGuard) and accuracy (99.71% versus 82.42% of DoSGuard) for 100,000 events SDN specific dataset under 5 spoofing attacks. The proposed SDN IDPS has same objective of achieving low-FPR and source level blocking but has significant structural differences compared to existing solutions as illustrated in Tables 5-7.

Table 5. Architectural and Design Comparison: Proposed SDN-IDPS vs. DDoSBlocker [27]

Dimension	Proposed SDN-IDPS (This Work)	DDoSBlocker [27] (Sinha et al., 2025)
Controller / OpenFlow version	Ryu / OpenFlow 1.3	Floodlight / OpenFlow 1.0
Stage-1 gate mechanism	EWMA adaptive-threshold heuristics on ICMP/SYN/UDP packet rates; z-score anomaly for other vectors	Time-based address mapping (1630 ms threshold λ_1); packet_in authentication module

Dimension	Proposed SDN-IDPS (This Work)	DDoSBlocker [27] (Sinha et al., 2025)
Stage-2 ML classifier	XGBoost + isotonic calibration + dual safety gate	Random Forest (triggering-based; activates only when packet_in rate exceeds λ_2)
Novel features introduced	NULL/XMAS flag ratios, up-down packet ratio, calibrated confidence fusion, EWMA per-source smoothing	% fake dst_IP, avg bytes/packet, % bidirectional flow rules, % fake flow rules
Probability calibration	Yes — isotonic regression post-processing	No
Victim shielding	Structural pre-enforcement MDS; 0 victim FP enforcements across 63 scenarios	Not addressed — blocking at switch host_port; no discrimination between victim and attacker
Mitigation granularity	Flow-level: targeted drop rules per source; victims exempted via MDS	Switch host_port level: deny flow rule per malicious port; blocks all traffic from that port
Cold-start handling	EWMA grace window; per-source smoothing on cold-start-sensitive features only	Time-gap threshold λ_1 inherently handles new-host address changes; no explicit grace window
Dataset size / source	Custom Mininet emulation; 63 attack scenarios (44 DoS + 19 DDoS), 11 vectors, 4 intensities	Custom SDN-specific dataset; 100,000 events (46,000 benign + 54,000 attack) on Geant Zoo topology
Protocol Counter Bypass	Identified, named, quantified (BUG-FIX-1)	Not addressed; noted that low-rate attacks below λ_2 may evade detection

Table 6. ML Algorithm Performance: Proposed XGBoost vs. DDoSBlocker Random Forest

ML Dimension	Proposed (XGBoost + Isotonic)	DDoSBlocker (Random Forest)
Algorithm class	Gradient-boosted ensemble	Bagged decision-tree ensemble
Probability calibration	Isotonic regression post-processing	None
Reported accuracy (scenario-level)	~89% (63 heterogeneous scenarios; 0 victim FP enforcements)	99.71% accuracy, 99.62% precision, 99.87% recall, 99.74% F1, 0.51% FPR

ML Dimension	Proposed (XGBoost + Isotonic)	DDoSBlocker (Random Forest)
Feature parity (train vs. inference)	Yes — single shared extraction function	Not explicitly verified
Detection latency	Median ~0.38 s at MEDIUM/HIGH; sub-second at all intensities except slow-accumulation edge cases	Avg ~3 s detection; avg ~0.5 s mitigation after detection
Class imbalance handling	F-beta threshold optimisation	5-window voting (counter must reach 5 consecutive positive windows before blocking)

Table 7. Three-Way Performance Comparison: Proposed SDN-IDPS vs. DoSGuard [26] vs. DDoSBlocker [27]

Capability	Proposed SDN-IDPS	DoSGuard [26]	DDoSBlocker [27]
ML calibration	Yes (isotonic)	No	No
Victim shielding	Yes (MDS, pre-enforcement)	No	No
Flow-level mitigation	Yes	No (host-level)	No (port-level)
Attack vectors covered	11 (incl. TCP flag-flood, PoD, large-UDP)	3 (MAC/IP/port spoofing)	5 (src_MAC, src_IP, src_port, dst_IP, dst_port spoofing)
Multi-source DDoS coverage	Yes (19 DDoS scenarios)	No	Hybrid models only (Exp-4)
Feature parity verified	Yes	No	No
Reported classifier accuracy	~89% scenario-level; 0 victim FP	82.42% / F1 84.27%	99.71% / F1 99.74% / FPR 0.51%
Bug-fix / engineering docs	3 BUG-FIX records with root-cause analysis	None	None

Key Contribution

- A precisely defined two-stage pipeline where heuristic escalation in Stage 1, victim identification, and machine learning classifier execution in Stage 2 occur in a strict order with explicit code-level handoff semantics, instead of the looser coupling observed in previous generations of hybrid systems.
- An asymmetric defense system that shields victims based on confidence-based adaptive dynamic TTL and evaluates its effect before the actual enforcement takes place, avoiding misclassification altogether and not fixing it after the fact.

- Exponential weighted moving average smoothing of EWMA-based features but leaving other features such as destination entropy and dispersion without smoothing, preserving the attack structure in the process.
- Calibration of the machine learning model using isotonic regression and enforcing two safety gates that require both the calibrated model confidence and statistical anomaly scores be above certain thresholds before enforcing the decision.
- Isotonic-regression calibration of the XGBoost output combined with a dual safety gate that withholds enforcement unless both the calibrated ML confidence and an independent statistical anomaly score clear their respective floors.
- A trust score mechanism reducing the evaluation time for trusted hosts and a multi-dimensional shield eviction mechanism that stops protecting the victims when symmetric traffic and destination entropy conditions are met.

Method, Experiments and Results

Proposed Methodology

The proposed system is a two-stage hybrid IDPS embedded in a Ryu SDN controller managing an OpenFlow 1.3 network, following the standard three-plane SDN model. The Data Plane consists of Open vSwitch instances emulated in Mininet; the Control Plane is the Ryu controller, which receives packet-in events for unmatched flows, accumulates per-source statistics, and installs flow rules and meter entries; the Application Plane hosts the IDPS detection logic, the ML inference module, and audit logging, all within the controller's Python process.

Detection proceeds in a fixed evaluation window. At the end of each window, accumulated per-source statistics are snapshot-frozen and passed through a strictly ordered pipeline: (i) Stage-1 rule-based heuristics apply protocol-specific adaptive thresholds — using an EWMA-based mean-plus-k-sigma ceiling with an outlier freeze condition — to ICMP, SYN, and UDP rates, supplemented by overall packet-rate z-score anomaly detection for less common vectors; sources crossing any criterion are escalated to a suspect registry with an attack-type label and an initial confidence score, and the corresponding rate-limit-or-block decision is deferred rather than committed immediately. (ii) A victim-identification routine then examines the destination IPs reported by Stage-1 suspects and registers any host showing receive-dominant traffic asymmetry above a minimum threshold as a protected victim, with a confidence-adaptive dynamic time-to-live; registered victims are exempted from further evaluation for the duration of their TTL, and a Multi-Dimensional Shield eviction rule later releases protection once traffic symmetry and destination entropy jointly normalise across consecutive windows. (iii) Only sources that survive Stage 1 without being reclassified as victims are passed to Stage 2, where an eleven-dimensional feature vector (average packet size, flow dispersion, TCP/UDP/SYN/ICMP/NULL/XMAS ratios, up-down packet ratio, destination entropy, and source-port entropy) is computed by a single shared extraction function used identically during both training and inference, guaranteeing feature parity. Selected cold-start-sensitive features are smoothed with a per-source EWMA before being passed to a pre-trained, isotonically calibrated XGBoost classifier, which returns a calibrated attack probability. (iv) This ML confidence is fused with an independently computed statistical anomaly score (combining packet-rate z-score, carry-forward generation count for slow-scan sources, and global packet-in-rate z-score) under a dual safety gate that withholds any enforcement action unless both signals clear minimum floors, and the fused confidence is then compared against block and rate-limit thresholds to select the final enforcement action. (v) Enforcement is executed in-band via OpenFlow high-priority drop rules for blocking and OpenFlow meter entries for rate limiting, without disrupting the SDN control plane.

This methodology was evaluated on a custom dataset generated from Mininet emulation covering eleven DoS/DDoS attack vectors — TCP SYN, UDP, ICMP Ping, TCP RST, TCP FIN, TCP ACK, TCP SYN-ACK, TCP PSH-ACK, TCP URG floods, UDP large-packet flood, and Ping of Death — each at LOW, MEDIUM, HIGH, and BURST intensities, together with distributed multi-source variants, for a total of sixty-three test scenarios.

Algorithm

Algorithm 1 summarises the per-window evaluation pipeline executed by the controller.

Algorithm 1: Two-Stage Hybrid Window Evaluation

Input: src_stats (per-source counters accumulated since last window), window duration W
Output: enforcement action ∈ {BLOCK, LIMIT, NO_ACTION} for each evaluated source

snapshot ← freeze(src_stats); reset live accumulators; carry forward sources below MIN_PACKETS
FOR each source s in snapshot with packets ≥ MIN_PACKETS and s ∉ blocked_ips:
    compute rate metrics: icmp_pps, syn_pps, udp_pps, overall_pps for s
    IF icmp_pps, syn_pps, or udp_pps exceeds its AdaptiveThreshold OR shows a jump OR z-score > Z_THRESHOLD:
        flag s as Stage-1 suspect with attack_type and initial confidence
        defer action (LIMIT if suspect_windows ≤ SUSPECT_WIN_LIMIT, else BLOCK)
FOR each candidate victim v referenced by a Stage-1 suspect's target_ips:
    IF rx_pkts(v) / max(tx_pkts(v),1) ≥ VICTIM_ASYMMETRY_MIN AND attacker_count(v) ≥ MIN_ATTACKERS:
        register v as VICTIM with TTL = VICTIM_TTL_MIN + confidence × (VICTIM_TTL_MAX − VICTIM_TTL_MIN)
        exempt v from Stage-1 enforcement and Stage-2 evaluation for TTL
FOR each Stage-1 suspect s not registered as a victim:
    x ← extract_features(s) // shared function, identical to training-time extraction
    x ← apply_EWMA_smoothing(x) // cold-start-sensitive features only
    ml_conf ← isotonic_calibrated_XGBoost.predict_proba(x)
    anomaly ← compute_anomaly_score(s) // z-score + carry-forward + global rate components
    IF ml_conf < SAFETY_GATE_ML_CONF OR anomaly < SAFETY_GATE_ANOMALY:
        action ← NO_ACTION
    ELSE:
        fused ← (anomaly ≥ FUSION_ANOMALY_HIGH) ? ml_conf : (ml_conf + anomaly) / 2
        action ← BLOCK if fused ≥ BLOCK_THRESHOLD; LIMIT if fused ≥ LIMIT_THRESHOLD; else NO_ACTION
FOR each deferred or Stage-2 action: install OpenFlow drop rule (BLOCK) or meter entry (LIMIT) on all datapaths
Evict victims whose symmetry_ratio and destination_entropy satisfy normal-traffic bounds for CONSECUTIVE_NEEDED windows

Demographic / Experimental Support for the Algorithm

The algorithm was evaluated on a synthetic, Mininet-emulated network topology rather than human-subject demographic data, since the unit of analysis is network traffic generated by emulated hosts. The relevant 'demography' for this domain is the population of attack scenarios and source/victim host pairs over which the algorithm was tested. Table 8 summarises this experimental population, and Table 9

reports the resulting detection-latency profile by intensity level, both drawn directly from the evaluation harness logs.

Table 8. Experimental scenario population used to evaluate the proposed IDPS

Category	**Count**	**Description**
Distinct attack vectors	11	SYN, UDP, ICMP, RST, FIN, ACK, SYN-ACK, PSH-ACK, URG floods, UDP large-packet, Ping of Death
Traffic intensity levels	4	LOW, MEDIUM, HIGH, BURST
Single-source (DoS) scenarios	44	Eleven vectors × four intensities (A1–A11 series)
Distributed (DDoS) scenarios	19	Multi-source variants (D1–D6 series, excluding unavailable combinations)
Total evaluated scenarios	63	Combined DoS + DDoS scenario population
Confirmed true misses	4	A6_LOW, A7_LOW, A7_BURST, A8_BURST / A9_BURST (TCP flag-flood variants)
Partial detections	1	D3_LOW (one of two attacking IPs missed)
Victim false-positive enforcements	0	No protected victim was incorrectly blocked or rate-limited

Table 9. Detection latency of the proposed IDPS by attack intensity

Intensity	**Median Latency**	**Min Latency**	**Max Latency**	**Notes**
LOW	~0.5 s	0.17 s	~50 s	Slow-accumulation cases push latency out
MEDIUM	~0.38 s	0.13 s	5.43 s	Fastest overall; thresholds crossed early
HIGH	~0.38 s	0.13 s	~50 s	ICMP HIGH outlier via Stage-1 failsafe
BURST	~0.55 s	0.11 s	~145 s	DDoS BURST cases staggered across windows

Discussions

The three-way comparison in Table 7 reveals the complementary positions of all three systems. DDoSBlocker [27] achieves the highest reported classifier accuracy (99.71%) through its time-based address mapping stage and a purpose-built SDN feature set, and its 0.51% FPR is substantially better than DoSGuard's 9.8% reported in the same paper. However, DDoSBlocker blocks at the switch host_port level, provides no victim discrimination mechanism, does not calibrate classifier probabilities, and does not verify feature parity between training and inference. The proposed SDN-IDPS differs structurally on all four of these dimensions: isotonic-regression calibration produces principled

confidence scores for the dual safety gate; the Multi-Dimensional Shield (MDS) enforces pre-enforcement victim exemption, recording zero false-positive victim enforcements across all 63 evaluation scenarios; a single shared feature-extraction function guarantees numerical parity between training and inference; and eleven heterogeneous attack vectors (including TCP flag-flood variants, Ping of Death, and large-UDP) are evaluated at four intensity levels including distributed multi-source DDoS variants — a substantially broader evaluation than either baseline. The residual detection misses (~11% of scenarios) are traceable to the Protocol Counter Bypass Problem at BURST intensity, an effect that is explicitly identified and quantified here but is absent from both comparators.

Conclusions

1. Problem addressed: reliable, low-latency DoS/DDoS detection across heterogeneous protocol vectors and intensities within an SDN controller, without misclassifying victim hosts or suffering training-deployment feature mismatch.
2. Method used: a two-stage hybrid IDPS combining adaptive-threshold rule-based heuristics, an interleaved asymmetric victim-shielding step with confidence-adaptive TTL, and a calibrated, EWMA-smoothed XGBoost classifier fused with an independent anomaly score under a dual safety gate, enforced in-band via OpenFlow drop rules and meters.
3. Key findings: In sixty-three simulated scenarios using Mininet, the architecture was able to detect about 89% of scenarios, with sub-second median latencies in the medium and high intensities, and no false-positive enforcement on victim hosts.
4. Limitations and Future Work: While some errors remained due to the TCP flag flood attack variants at LOW and BURST intensities, which were caused by the Protocol Counter Bypass Problem resulting from hardware flow-table offloading that circumvented the controller's packet-in path, future work should include using OpenFlow statistics polling in addition to packet-in based counters, implementing online/incremental learning to handle distribution shift, expanding scope to application-level DDoS (HTTP flood and Slowloris attacks), exploring federated multi-controller intelligence, and testing the architecture on real OpenFlow hardware for control channel overhead characterization.

References

1. N. McKeown et al., "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.
2. S. Shin, P. Porras, V. Yegneswaran, M. Fong, G. Gu, and M. Tyson, "FRESCO: Modular composable security services for software-defined networks," in *Proc. NDSS*, San Diego, CA, USA, 2013.
3. S. M. Mousavi and M. St-Hilaire, "Early detection of DDoS attacks against SDN controllers," in *Proc. Int. Conf. Computing, Networking and Communications (ICNC)*, 2015, pp. 77–81.
4. R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," in *Proc. IEEE LCN*, 2010, pp. 408–415.
5. P. Kokila, S. T. Selvi, and K. Govindarajan, "DDoS detection and analysis in SDN-based environment using support vector machine classifier," in *Proc. ICoAC*, 2014, pp. 205–210.
6. Y. Tang, L. Zhang, and Y. Liu, "DeepDefense: Identifying DDoS attack via deep learning," in *Proc. IEEE SMARTCOMP*, 2016, pp. 1–8.
7. I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy," in *Proc. IEEE ICCST*, 2019, pp. 1–8.

8. A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in *Proc. 22nd Int. Conf. Machine Learning (ICML)*, 2005, pp. 625–632.
9. Y. Wang, T. Hu, G. Tang, J. Xie, and J. Lu, "SGS: Safe-guard scheme for protecting control plane against DDoS attacks in software-defined networking," *IEEE Access*, vol. 7, pp. 34699–34710, 2018.
10. A. Karmakar, N. Dey, and A. Das, "A collaborative victim-distress scoring scheme for DDoS detection in distributed SDN environments," *J. Network Comput. Appl.*, vol. 152, pp. 102–115, 2020.
11. A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," in *Proc. ACM SIGCOMM*, 2004, pp. 219–230.
12. K. Xiao, J. Zhu, and X. Li, "A cold-start aware adaptive threshold scheme for SDN-based DDoS detection," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 3, pp. 3204–3217, 2021.
13. S. Shin, V. Yegneswaran, P. Porras, and G. Gu, "AVANT-GUARD: Scalable and vigilant switch flow management in software-defined networks," in *Proc. ACM CCS*, 2013, pp. 413–424.
14. M. Al-Fuqaha, A. Khreishah, M. Anan, and A. Khan, "Convolutional neural networks for SDN-based intrusion detection: A survey," *IEEE Access*, vol. 8, pp. 71544–71561, 2020.
15. T. V. Phan, T. M. R. Gias, S. T. Islam, T. T. Huong, N. H. Thanh, and T. Bauschert, "Q-MIND: Defeating stealthy DoS attacks in SDN with a machine-learning based defense framework," in *Proc. IEEE GLOBECOM*, 2019, pp. 1–6.
16. X. Zhang, X. Zhang, and L. Han, "An energy efficient Internet of Things network using restart artificial bee colony and wireless power transfer," *IEEE Access*, vol. 7, pp. 12686–12695, 2019.
17. X. Zhong, L. Zhang, and Y. Wei, "Dynamic load-balancing vertical control for a large-scale software-defined Internet of Things," *IEEE Access*, vol. 7, pp. 140769–140780, 2019.
18. M. Malik, M. Dutta, and J. Granjal, "A survey of key bootstrapping protocols based on public key cryptography in the Internet of Things," *IEEE Access*, vol. 7, pp. 27443–27464, 2019.
19. B. Pfaff et al., "The design and implementation of Open vSwitch," in *Proc. USENIX NSDI*, 2015, pp. 117–130.
20. T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. ACM SIGKDD*, 2016, pp. 785–794.
21. F. Hu, Q. Hao, and K. Bao, "A survey on software-defined network and OpenFlow: From concept to implementation," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 2181–2206, 2014.
22. D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, 2015.
23. R. Mahajan, S. M. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker, "Controlling high bandwidth aggregates in the network," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 3, pp. 62–73, 2002.
24. B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: Rapid prototyping for software-defined networks," in *Proc. ACM SIGCOMM Workshop HotNets*, 2010, pp. 1–6.
25. S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Netw.*, vol. 1, no. 4, pp. 397–413, 1993.
26. J. Li, T. Tu, Y. Li, S. Qin, Y. Shi, and Q. Wen, "DoSGuard: Mitigating denial-of-service attacks in software-defined networks," *Sensors*, vol. 22, no. 3, p. 1061, Jan. 2022. https://doi.org/10.3390/s22031061

27. M. Sinha, P. Bera, M. Satpathy, K. S. Sahoo, and J. J. P. C. Rodrigues, "DDoSBlocker: Enhancing SDN security with time-based address mapping and AI-driven approach," *Computer Networks*, vol. 259, p. 111078, Apr. 2025. https://doi.org/10.1016/j.comnet.2025.111078