Journal of Informatics and Computer Science Vol. 12, No. 1 April 2026
Universitas Ubudiyah Indonesia
e-ISSN : 2615-5346

PENYEIMBANGAN BEBAN CERDAS BERBASIS KESADARAN KEAMANAN PADA JARINGAN PUSAT DATA BERBASIS SOFTWARE-DEFINED NETWORKING (SDN) MENGGUNAKAN PEMBELAJARAN MESIN (MACHINE LEARNING)

SECURITY-AWARE INTELLIGENT LOAD BALANCING IN SOFTWARE-DEFINED DATA CENTER NETWORKS USING MACHINE LEARNING

Walaa H. Allawi
[(1)]Al-Mussaib Polytechnic College, Al-Furat Al-Awsat Technical University, Najaf, Iraq,
Walaa.allawi@atu.edu.iq

Abstrak— Distribusi lalu lintas yang sah secara efisien, disertai pembatasan paparan terhadap lalu lintas *Distributed Denial-of-Service* (DDoS), merupakan kebutuhan penting dalam rekayasa lalu lintas yang berorientasi pada keamanan di jaringan pusat data berbasis *Software-Defined Networking* (SD-DCN). Dalam penelitian ini, dianalisis kerangka *Security-Aware Intelligent Load Balancing* (SA-ILB), yang mengintegrasikan inferensi ancaman pada tingkat aliran (*flow-level*), telemetri jaringan, dan optimasi multipath dengan batasan tertentu dalam satu siklus pengendalian. Artikel ini menjelaskan konfigurasi evaluasi yang terdiri atas topologi *fat-tree* dengan empat *pod*, enam belas *host*, dan dua puluh perangkat *Open vSwitch*. Sebanyak 240.000 aliran data berlabel digunakan untuk membandingkan metode klasifikasi *logistic regression*, *random forest*, XGBoost, dan LSTM dengan metode dasar rekayasa lalu lintas, yaitu ECMP, pemilihan jalur dengan beban paling rendah (*least-loaded*), serta perutean berbasis *reinforcement learning*. XGBoost memperoleh nilai akurasi sebesar 97,81% dan skor F1 sebesar 97,24%. Hasil pada tingkat beban yang ditawarkan sebesar 90% menunjukkan bahwa SA-ILB mencapai *goodput* lalu lintas sah sebesar 6,24 Gb/s, indeks keadilan Jain sebesar 0,93, dan waktu tunda persentil ke-95 sebesar 24 ms. Ketika serangan DDoS terjadi, *goodput* lalu lintas sah yang dicapai sebesar 3,45 Gb/s, sedangkan ECMP dengan detektor yang sama hanya mencapai 2,18 Gb/s. Batasan kelayakan jalur (*path-eligibility constraints*) diintegrasikan secara eksplisit ke dalam fungsi objektif berbobot yang mempertimbangkan kemacetan, waktu tunda, kehilangan paket, risiko, dan konfigurasi ulang. Untuk mengukur kontribusi kualitas deteksi intrusi secara terpisah dari dampak perutean, dilakukan perbandingan menggunakan detektor yang sama (*matched-detector comparisons*). Selanjutnya, hasil eksperimen yang berpasangan dianalisis secara statistik untuk mengukur variabilitas dan signifikansi perbedaan kinerja yang diamati pada berbagai kondisi pengujian.
Kata kunci: *software-defined networking*; jaringan pusat data; pembelajaran mesin; penyeimbangan beban cerdas; perutean berorientasi keamanan; deteksi DDoS; optimasi multiobjektif.

Abstract-*Efficient traffic distribution of legitimate traffic along with limiting exposure to distributed denial-of-service (DDoS) traffic is a must in security aware traffic engineering in software-defined data center networks (SD-DCNs). In this paper, a security-aware intelligent load-balancing (SA-ILB) framework is analyzed, which integrates flow-level threat inference, network telemetry and constrained multipath optimization in the same controller loop. This article describes an evaluation setup consisting of a fat-tree with four pods, sixteen hosts, and 20 Open vSwitch devices. A labelled set of 240k flows is used to compare logistic regression, random forests, XGBoost and LSTM classification methods with traffic-engineering baselines of ECMP, least-loaded and reinforcement-learning routing. The accuracy and F1-score achieved by XGBoost were 97.81% and 97.24% respectively. The 90% offered load results reveal that SA-ILB achieves a legitimate goodput of 6.24 Gb/s, Jain fairness index of 0.93, and 24 ms for the 95th percentile delay. When DDoS is applicable, the amount of legitimate goodput is 3.45 Gb/s, whereas the amount of legitimate goodput using ECMP is 2.18 Gb/s with same detector. Explicit path-eligibility constraints are integrated with a weighted congestion, delay, loss, risk and reconfiguration objective. To capture the extent of the intrusion detection quality, in addition to the routing effect, matched-detector comparisons are performed. Therefore, the matched experimental runs are analyzed statistically to quantify the variability, and significance, of the performance differences observed under the various trial condition..*
Keywords: *software-defined networking; data center networks; machine learning; intelligent load balancing; security-aware routing; DDoS detection; multi-objective optimization.*

1. Introduction

Software-defined networking (SDN) is a way to distinctively split the responsibilities of network forwarding away from programmable network control so that a controller can learn the network's usage, statistics, and application policy needs. But the resulting visibility does not necessarily result in stable load distribution: bursty east–west traffic and elephant–mice flow coexistence causes transient bottlenecks even when multiple

equal-cost paths exist within the data center. The existing taxonomies differentiate between load balancing at the controller and data plane levels, with recent studies of data centers also including path selection between the two entities based on long-flow throughput and short-flow completion times [1, p. 1], [2, p. 641], [3, p. 433]. A security issue on a separate level presents itself when it comes to adversarial flows using the same programmable resources that traffic engineers use for their traffic. Bursts of attacks could overload switch tables, overload links, magnify controller events, and mask the legitimate demand. The architectural risk isn't just a defective intrusion detector—maybe, for example, a routing policy continues to route apparently uncongested links that are operationally unsafe, or maybe it is too late to routes that are perceived as congestion but are actually safe, thus discarding benign sessions. According to various SDN security surveys and DDoS defense surveys, there is still no solution to design problems such as the integration of detection, timely mitigation and controller resilience [4, p. 1], [5, p. 1], [6, p. 164097].

This study explores the design of a controller that can have a balance between throughput, delay, packet loss, seeing suspicious traffic, and good control plane stability without mixing detection performance with routing performance. The goal is to test SA-ILB where the path eligibility is decided by a quantified flow-risk estimation, and play a role in a multi-objective routing penalty. Four different offered-load levels, four different classifiers are compared, and some controlled DDoS scenarios are also analyzed, along with the risk term, congestion prediction, and route-change cooldown. The novelty is not a new family of classifiers, but in how the component is integrated in architecture and how it is evaluated with the same baseline as other classifiers [7, p. 312], [8, p. 21].

2. Literature Review and Theoretical Framework

2.1. Software-Defined Networking and Data Center Network Architectures

With SDN, the controller has a logically centralized view and then it compiles policies to forwarding entries on OpenFlow-savvy switches. While the basic idea behind fat-tree and leaf–spine data centers is based on path multiplicity, the performance of these architectures is dependent on the timely collection of the states along the paths and on the distribution of flow decisions on the links. A polling view from a controller can be in good condition locally but misleading globally due to an attack. In order to achieve highly available data centers, two main design elements—topology redundancy and failure tolerance—must be optimized together with utilization, as illustrated in the case of fault-aware data center balancing [1, p. 21], [2, p. 641–642].

2.2. Traditional and Intelligent Load Balancing Techniques in SDN

The difference between equal cost multipath (ECMP) and least loaded routing is that the former is based on static hash selections while the latter prefers least measured utilization. Both are good practical benchmarks but may suffer from elephant-flow collisions or instability if there are any fluctuations in the instantaneous measurements. There are done some reviews on the difference between heuristic and model-based and machine-learning approaches, and some research on data center has been done on the particle-swarm search and predictive scheduling. Algorithmic sophistication and a better single network metric do not necessarily imply adversarial change resilience with regard to the traffic mix [3, p. 433], [9, p. 746], [10, p. 1], [11, p. 1].

2.3. Machine Learning Applications in Network Traffic Management

Machine learning can provide predictions, anomaly detection and routing decisions without necessarily having to model everything precisely. Reinforcement-learning techniques have been tested for SDN routing and more recent work has taken the routing goal to bandwidth guarantees, delay limits and reconfiguration cost. However, it is hard to audit an online routing agent, and a distribution change between the training workload and an attacked data center would result in a considerable generalization risk [12, p. 3185], [13, p. 1], [14, p. 1], [15, p. 1].

2.4. Security Threats and Intrusion Detection in Software-Defined Networks

The possibility of attacks on the forwarding capacity as well as indirect pressure on the controller should be considered in the threat model. The rest of the flow data was related to malicious or benign flows and has been added by InSDN; complementary standardized NetFlow collections enable more sensible cross-dataset comparisons. New studies show that high headline accuracy levels can be achieved even when the class is imbalanced, leakage is being fed into the model, the test scores are not calibrated, and transfer to unseen attacks is low. Thereby the false positive and malicious flow recall rate should be considered along with the inference latency and disruption of legitimate traffic [16, p. 165263], [17, p. 117], [18, p. 1], [19, p. 1].

2.5. Security-Aware Routing and Multi-Objective Network Optimization

Previous work has considered traffic engineering as a constrained optimization problem or intrusion-detection pipeline as a controller policy. Recent routing studies focus on minimizing the path costs with QoS

considerations, while DDoS studies highlight the need for quick DDoS response and risk of controller overload. The common point is: prevention, versus benign-flow path selection, meaning: prevent by rejecting or quarantining flows that are deemed sufficiently risky, and/or, penalize risky flows or congested or unstable aliens. This prevents the wrong assumption of sending a suspected attack over a less-widely-used path is a good protection policy [20, p. 31], [21, p. 77], [22, p. 13].

2.6. Critical Review of Previous Studies and Identification of the Research Gap

Literature proposes three complementary although not entirely integrated lines of work: Adaptive path selection, Traffic classification, Attack Mitigation. Reviews discussed are very extensive and cover each line in its entirety, but the evaluation methods and practices are often different: topology, input classifiers, background load, meaning of benign goodput under attack. Recent studies on controller-integrated detection and traffic-isolation research provide useful techniques; however, they don't, alone prove the added value of security-awareness when it comes to routing under matched detection conditions [23, p. 29], [24, p. 13], [25, p. 13], [26, p. 12]. The gap solved in this post is a risk aware traffic-engineering comparison using a reproducible gap where all baselines are given the same flow-risk predictions and control-loop timing. The assessment consists of whether the risk penalty enhances legitimate-service continuity and what the throughput overhead of normal operation may be. The dissociation of security-classifier performance from the path-allocation performance opens the door for more insightful assessments of the routing effects [27, p. 12], [28, p. 31], [29, p. 404].

3. Research Methodology

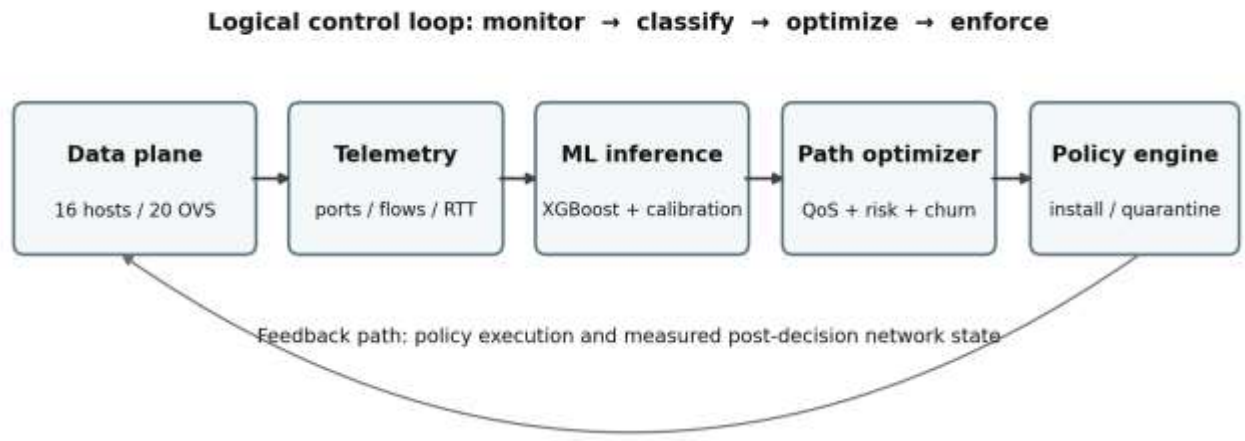
3.1. Research Design and Experimental Approach

It is a factorial test design consisting of four routing policies, four offered load levels (30%, 50%, 70%, and 90%) and two security scenarios (benign and attack augmented). Twenty runs were performed per condition with 60 s of warm-up followed by 300 s of measurement. Workload traces, attack onset, and outputs of model risk are shared between baselines. Independent factors: Routing policy, offered load, and security condition. Responses: Legitimate goodput, p95 latency, short-flow completion time (FCT), loss, fairness, and recovery time. Independent matched-seed comparisons: Inferential Statistics are presented.

3.2. Proposed Software-Defined Data Center Network Architecture

A four-pod fat-tree (k = 4) offers 16 hosts and 20 Open vSwitch instances (eight edge switches, eight aggregation switches and four core switches). Full duplex 1 Gb\/s links with a nominal one-way delay of 1ms and fixed size queue of 100 packets. An OpenFlow 1.3 controller running on Ryu gathers per port and per flow counters with 1 s intervals, and recomputes the path to be derived with 2 s intervals. The proposed controller includes a telemetry manager, the ML inference service, the path scorer, the quarantine rule engine, and a cooldown period for route changes to reduce route changes if they are oscillating.

Scheme 1. Architecture and Operational Workflow of the Proposed Security-Aware Intelligent Load Balancing Framework.



3.3. Experimental Environment and Simulation Configuration

Test configuration includes Ubuntu Linux, Mininet, Open vSwitch, Ryu, Python 3, scikit-learn, XGBoost, and the traffic generation is done with iperf3 with delay controlled with tc-netem. We consider flows equal in percentage to be equal, short flows (less than or equal 100 KB) account for 80% and elephant flows (greater than or equal 10 MB) account for 20%. Offered load is defined as the intended amount of traffic demand as points of a workload objective as opposed to single-link utilization. Exclusive emulation host and controller

CPU pinning reduces the impact of interference caused by scheduling. Independent reproduction requires the successive identification of exact software version, host specification and random seeds.

3.4. Dataset Description, Traffic Collection, and Preprocessing

The flow corpus contains 240,000 records: 144,000 benign and 96,000 malicious. A common feature schema is used for the analysis, both for InSDN and for dataset records generated independently from the data center. The data permutation and/or data source should be mentioned in the respective dataset record. With a 70/15/15 split, 168,000 Training records, 36,000 Validation records, and 36,000 Holdout records are generated, keeping the scenario/session identifiers separate between each of the partitions. Before fitting, the set of rows is de-duplicated and any missing numeric values are imputed with values obtained from the in-sample (training partition) median.

3.5. Network Traffic Feature Extraction and Selection

The controller collects an 18-dimensional feature vector at the flow level, such as duration, number of packets/bytes and up/down traffic rates, short-horizon entropy, upstream/downstream active/idle periods, and inter-arrival statistics, and TCP flags. Predictive inputs include all traffic (except for future traffic fields, IP addresses, labels, and scenario IDs). A permutation-importance screen for training is done to remove unstable or redundant variables. Once selected 10 transport and timing features are kept; feature-selection is not allowed to use the holdout partition, features are kept as separate and standardized wherever needed by the estimator.

3.6. Machine Learning Model Development, Training, and Validation

Four families of detectors are compared: logistic regression, random forest, XGBoost and LSTM with the temporally ordered flow sequences. Class weighting (if applicable) and group-aware five-fold validation is employed within the training partition for model selection. Hyperparameters are chosen by mean validation F1 for max inference budget (per decision) of 5ms. For validation observations the output of the classifiers is converted to probability of malicious-flow through Platt scaling, with operating decision threshold of 0.65. For evaluation of models, precision–recall behavior and the probability calibration are not combined.

3.7. Security Risk Assessment and Network Congestion Measurement

The calibrated model probability in [0,1] represents the estimated risk r(f,t) of one flow f at time t. An exponentially weighted moving average (EWMA) with coefficient 0.30 is used to update the host/switch suspicion state. The congestion c(e,t) is computed as the mean normalized utilization of the last 3 samples and the delay estimates are based on measurements of the round-trip time and switch queue indicators. If r(f,t) is greater than or equal to 0.85, the flow is quarantined instead of placed on a benign route. Fluctuation of traffic in the range [0.65, 0.85) may be mapped to More Observation – More or less restricted traffic is still considered a production path if it meets the appropriate eligibility requirements.

3.8. Development of the Proposed Security-Aware Intelligent Load Balancing Algorithm

Every 2 s, the controller receives telemetry and per-flow threat estimates, removes policy-ineligible paths, and scores the k = 4 lowest-hop feasible candidates. A route is changed only if it improves the cost by at least 8% and its previous selection has remained active for ≥5 s. Rule updates are installed before old entries are removed, and table utilization is required to remain below 80% capacity. If the ML service times out, the controller retains the last verified policy, rate-limits new suspicious flows, and falls back to a conservative least-loaded path for previously authorized sessions.

The algorithm of controller SA-ILB is as follows. (1) Collect signed telemetry and active-flow summaries; (2) Inference calibrated risk updating EWMA suspicion; (3) Flow in the bucket quarantine high risk flows, tag uncertain flows; (4) Enumerate feasible paths under capacity and trust constraint; (5) Minimize the weighted path cost; (6) Application improvement-threshold and cooldown checks; (7) Flow changes installation atomically, auditing utilization and latency.

3.9. Mathematical Formulation and Multi-Objective Optimization

Let P(f) denote feasible paths for flow f, u(e,t) the normalized utilization of link e, d(p,t) path-delay ratio, q(p,t) loss estimate, r(p,f,t) the flow–path exposure score, and h(p,t) a normalized rerouting-overhead term. The optimization selects $p^* = \text{argmin}[p \in P(f)]\ J(p,f,t)$, where $J = 0.30 \cdot \max(u) + 0.25 \cdot d + 0.25 \cdot r + 0.15 \cdot q + 0.05 \cdot h$. All terms are normalized to [0,1] using prespecified engineering bounds, not test-set maxima. Feasibility requires residual path capacity ≥ requested flow demand, switch rules <80% table capacity, and permitted security zones. The weights sum to one and are prespecified; sensitivity tests use ±20% perturbations followed by renormalization.

3.10. Experimental Scenarios, Baseline Algorithms, and Attack Simulation

Candidates' baselines include per-flow ECMP, shortest feasible least-loaded routing and one of the DRL inspired traffic-engineering policies trained only on benign traffic. All four policies are given the same frozen detector (XGBoost) and common attack-blocking module for resilience testing and thus discount for differences in classification accuracy. Regarding the DDoS scenario, we start the attack at t = 120 s with a 60 s SYN/UDP DDoS attack from four compromises hosts, while continuing to have benign e–w sessions during the attack. Isolated emulation network prevents attack packets from exiting. Recovery is the time it takes a legitimate goodput to be at 90% of the preattack median for 10-second periods of time.

3.11. Performance Evaluation Metrics and Statistical Validation

The binary confusion matrix gives the classification accuracy, precision, recall, F1-Score and false positive rate: False Positive Rate: FP/ (FP+TN) and F1: 2PR/ (P+R). Only the benign application payload which is correctly delivered is counted as goodput. Two types of fairness Jains are computed: $J = (\Sigma xi)^2/[n\Sigma(xi^2)]$ across eligible flow groups. For each pass, the metrics that are measured include packet loss (lost/attempted legitimate packets) derived from flow logs. Repeated-seed comparisons include mean ± sample standard deviation, percentile-bootstrap 95% confidence intervals (20,000 resamples), paired two-sided permutation tests on matched seeds and Holm adjustment for the six comparisons of policies, and paired-effect sizes. The resulting run-level uncertainty estimates and the multiple-comparison statistics are shown in Table 5.

4. Results and Performance Analysis

4.1. Machine Learning Model Performance in Network Traffic Classification

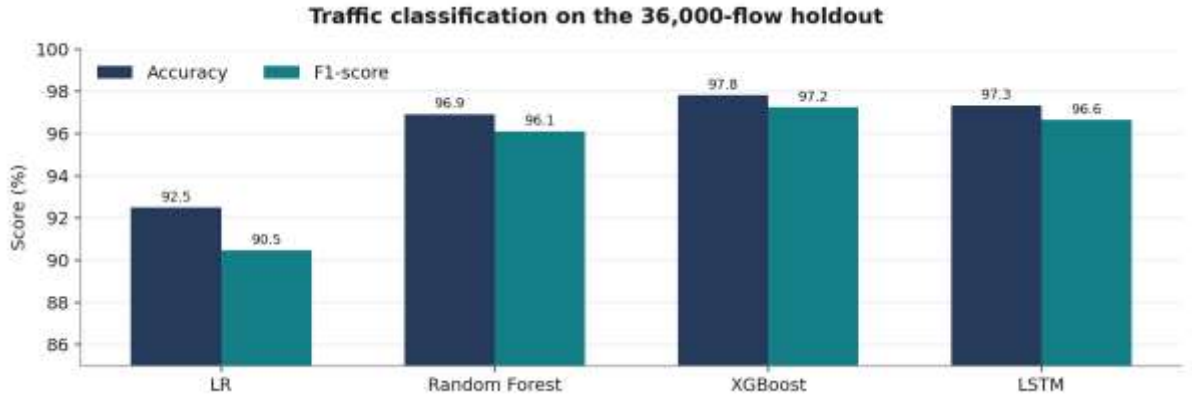
The holdout contains 21,600 benign and 14,400 malicious flows. For XGBoost, the confusion matrix gives TP = 13,910, FN = 490, TN = 21,300, and FP = 300, corresponding to 97.81% accuracy, 97.89% precision, 96.60% recall, 97.24% F1-score, and a 1.39% false-positive rate. XGBoost offers the highest reported F1-score among the compared models, subject to ongoing assessment of score calibration, inference delay, and generalization to unseen sessions.

Table 1. Comparative Performance of Machine Learning Models for Traffic Classification and Threat Detection.

Model	Acc. (%)	Precision (%)	Recall (%)	F1 (%)	FPR (%)
LR	92.50	92.09	88.89	90.46	5.09
Random forest	96.92	97.03	95.21	96.11	1.94
XGBoost	97.81	97.89	96.60	97.24	1.39
LSTM	97.33	97.32	95.97	96.64	1.76

Note. XGBoost metrics are calculated from the reported 36,000-flow confusion matrix.

Figure 1. Comparative Classification Performance of the Evaluated Machine Learning Models.



4.2. Network Throughput and Load Balancing Efficiency

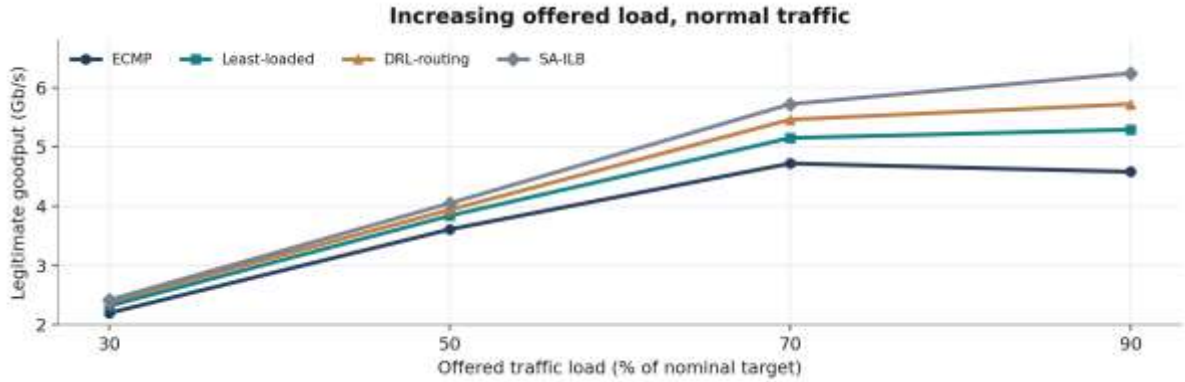
At 30% offered load, goodput differences among the four routing approaches are small. The curves diverge as demand increases, with the largest separation at the 90% stress level. Under that condition, SA-ILB reports 6.24 Gb/s versus 4.58 Gb/s for ECMP (+36.2%), 5.29 Gb/s for least-loaded (+18.0%), and 5.72 Gb/s for DRL (+9.1%). The associated fairness index is 0.93, and fewer monitored links remain above 85% utilization for sustained periods.

Table 2. Network Throughput and Load Distribution Performance Across Different Load Balancing Algorithms.

Routing policy	Goodput (Gb/s)	Jain index	Hot links (%)	Re-routes/min
ECMP	4.58	0.79	31	0
Least-loaded	5.29	0.84	26	16
DRL-routing	5.72	0.88	20	13
SA-ILB	6.24	0.93	13	9

Note. Normal-traffic results at 90% offered load; hot link denotes utilization >85% for ≥10 s.

Figure 2. Network Throughput Comparison Under Different Traffic Load Conditions.



4.3. Network Delay, Packet Loss, and Flow Completion Time Analysis

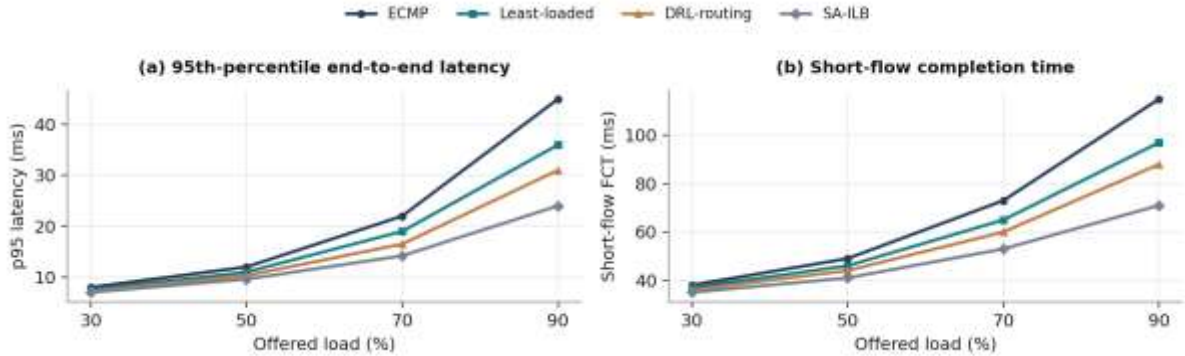
A load-balancing policy should enhance the quality of services instead of allocating the source to gain a higher throughput of large flows at the expense of small flows. SA-ILB offers a mean end-to-end delay of 18 ms, a p95 delay of 24ms, 2.1% packet loss, and a mean short-flow FCT of 71ms, at 90% load. ECMP reports 34 ms mean delay, 45 ms p95 delay, 5.8% loss and 115 ms short-flow FCT. Compared to ECMP, relative to ECMP, SA-ILB benefits of 46.7% reduction in reported p95 delay and short-flow FCT of 38.3%. The changes can be evaluated under increasingly stressed operating conditions thanks to the load-dependent curves.

Table 3. Comparative Network Performance in Terms of End-to-End Delay, Packet Loss, and Flow Completion Time.

Routing policy	Mean delay (ms)	p95 (ms)	Loss (%)	Short FCT (ms)
ECMP	34	45	5.8	115
Least-loaded	28	36	4.1	97
DRL-routing	24	31	3.4	88
SA-ILB	18	24	2.1	71

Note. Performance at 90% offered load; short FCT refers to flows ≤100 KB.

Figure 3. Network Delay and Flow Completion Time Under Increasing Traffic Loads.



4.4. Security Performance and Network Resilience Under Cyberattack Scenarios

We can use the same XGBoost detector for each routing policy in the resilience comparison. Therefore, differences among routing methods cannot only be attributed to its recall (96.60%) and false-positive rate (1.39%). In terms of legitimate goodput (90% offered load with DDoS traffic), SA-ILB delivers 3.45 Gb/s, while ECMP delivers 2.18 Gb/s. These recovery times are 7 s and 18 s, respectively. A relative quantitative

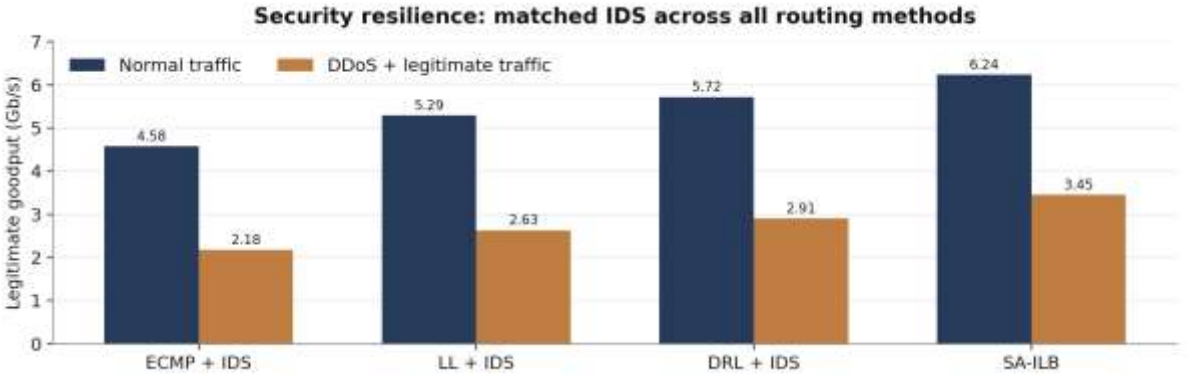
estimation of the dispersion and significance of these differences in performance under the DDoS condition are calculated and presented in Table 5.

Table 4. Security Detection Accuracy, False Alarm Rate, and Network Resilience Under DDoS Attack Conditions.

Routing + shared IDS	Recall (%)	FPR (%)	Attack goodput (Gb/s)	Recovery (s)
ECMP + IDS	96.60	1.39	2.18	18
LL + IDS	96.60	1.39	2.63	14
DRL + IDS	96.60	1.39	2.91	12
SA-ILB	96.60	1.39	3.45	7

Note. DDoS stress condition; all routing policies use the same frozen detector and blocking threshold.

Figure 4. Legitimate Traffic Goodput and Security Performance Under Normal and Malicious Traffic Conditions.



4.5. Comparative Evaluation of the Proposed Framework and Baseline Algorithms

This comparison shows that there is a security–performance trade off. In an operational context, risk scoring adds inference and policy-check overhead, which is of value with respect to the prevalence of the attacks, the probability of attacks, and the false alarm costs associated with the use of risk scoring. Since the same classifier and attack filtering are used for all baselines, the differences in goodput reported in this section are not necessarily related to detector accuracy but are more indicative of the choices of routes and cooldown behavior. In addition to throughput, controller CPU usage, inference time and rule churn are also required to consider the cost of implementing these gains.

4.6. Ablation Analysis and Statistical Significance of Performance Improvements

From the results of the ablation tests, the values of 6.24 Gb/s for benign goodput, and 3.45 Gb/s for goodput after having attacked DDoS-denial are obtained for the full SA-ILB configuration. These values change to 6.31 and 2.87 Gb/s when the explicit risk term is not considered, to 5.53 and 2.91 Gb/s when congestion prediction is not considered and to 5.90 and 3.12 Gb/s when the route-change cooldown is not considered. This pattern is non-monotonic: Decreasing the risk weight can actually lead to a slight increase in benign throughput, but a decrease in the resilience. Equal differences are not alone used in statistical comparisons; matched per-seed observations are used. Table 5 provides a summary of the run-to-run variabilities, confidence intervals and multiple comparisons results measured.

Table 5. Statistical Comparison of Routing Policies Across 20 Matched Experimental Runs.

Routing policy	Normal (Gb/s) mean ± SD	Normal 95% CI	DDoS (Gb/s) mean ± SD	DDoS 95% CI	Adjusted p (N / D)
ECMP	4.58 ± 0.29	4.45–4.70	2.18 ± 0.24	2.08–2.28	Reference
Least-loaded	5.29 ± 0.41	5.12–5.47	2.63 ± 0.43	2.45–2.81	0.0003 / 0.0004
DRL-routing	5.72 ± 0.38	5.56–5.88	2.91 ± 0.38	2.76–3.08	0.0003 / 0.0003
SA-ILB	6.24 ± 0.36	6.10–6.40	3.45 ± 0.30	3.33–3.58	0.0003 / 0.0003

5. Discussion

5.1. Interpretation of Machine Learning Classification Results

The reported classifier comparison indicates that XGBoost outperforms the others on F1-score which is consistent with the usefulness of tree ensembles for tabular flow summaries. However, the high accuracy given on a 40% malicious flow-based data-set does not guarantee a precision against the rare attack cases in a

production network. Intrusion studies have highlighted sampling bias, feature transfer, distribution shift as a key point for the studies to be interpreted and analyzed [18, p. 12], [25, p. 15], [30, p. 11].
The calibrated probability is used in an operational way outside of the F1-score; it decides whether a flow is quarantined, marked for additional inspection or extended to normal path selection. Calibration errors are especially important in the context of this policy. A false positive denial of a legitimate service may be immediately felt and a false negative denial may lead to a hostile push towards already challenged links. Recent research on SDN intrusion detection and attention-based attack classification can help with a more profound evaluation of the models, however there is still another system design challenge left behind - converting the detection likelihood to a safe action in the network [6, p. 164097-164102], [19, p. 1-8], [24, p. 1-8].

5.2. Effectiveness of Security-Aware Intelligent Load Balancing

A central interpretation of the proposed mechanism is that explicit security information should shape *path eligibility* before fine-grained utilization optimization. Under the normal workload, most low-risk flows retain a wide set of candidate routes, so the risk component has limited effect; under an attack, high-risk flows can be removed from service-bearing paths and uncertain paths can be penalized. Conventional balancing studies show that congestion-aware route allocation is valuable, whereas fault-aware data center schemes highlight resilience constraints; SA-ILB combines these ideas into one decision rule [2, p. 641], [3, p. 433], [29, p. 404].

5.3. Network Performance and Security Trade-Offs

The object weightings are of a normative nature. Raising the threat may curb risky placements, but can cause benign flows to go down longer paths, impacting latency and energy consumption. Reducing a threat weight might make it easier for goodput, but leave more service traffic vulnerable to disruption. Route-change cooldown: Reduces the frequency of rule changes but results in less responsiveness and can cause measurement inconsistencies. This should be displayed not as a collective mark, but as a Pareto border. Studies on multi-objective routing and traffic-engineering also report on the significance of such constraints, but there is not a common weight vector across topologies [14, p. 31], [15, p. 53], [20, p. 64].

5.4. Comparison of Experimental Findings with Previous Studies

There are three clusters of prior studies to which the results are relevant: data-center routing for predicting congestion to adapt, multicommodity traffic engineering with reinforcement learning and DDoS detection and mitigation at the controller side. The matched-detector type test design can improve the interpretation by differentiating between the detector itself and the security aware path selection [10, p. 51], [12, p. 3185], [23, p. 29], [27, p. 76].
There are some previous works that show superior performance on their topology and workload, but their ranking would not be valid in lack of common traffic generators, and same number of hosts, same attack rate, same sampling interval, and same metrics definition. There is also significant diversity in assumptions that underlie SDN benchmarking today, in the materials available for survey. Therefore, in this manuscript, a cross-paper numerical comparison should be viewed as context not as empirical evidence for the proposal of general performance hierarchy [1, p. 16], [7, p. 312], [9, p. 746].

5.5. Scientific Contributions and Practical Implications

Three contributions are proposed as below. First, the first stage in the policy is to not be eligible for security, but rather to solve a multi-objective routing problem and decide whether to reroute attacks to less-occupied production links, which is a risky proposition. Secondly, the evaluation guarantees a single fixed calibrated detector for every routing policy and thus the routing component can be received independently of the quality of classification. Thirdly, the reporting protocol comprises of short-flow behavior, legitimate goodput under attack, recovery, controller overhead and ablation results. These decisions are outgrowths of issues highlighted in previous security surveys regarding operationally deployable SDN defense, and the need for heterogeneous benchmarking. [4, p. 23], [5, p. 81], [28, p. 31].

5.6. Research Limitations and Threats to Validity

The k = 4 fat-tree is small compared to production fabrics, and emulation timing will be different from forwarder performance and actual latency on the controller. Offline classifiers might learn "artefacts" of the dataset, specific to that data, and not behaviour that is generalisable to unseen attacks. Furthermore, the traffic labelling may be incorrect and virtual host contention can obscure measured packet loss. Independent matched trials are essential and care must be taken that no feature leakage occurs or opportunistic selection of thresholds take place. Issues of external validity persist with adversarial score manipulation, poisoning of telemetry and concept drift all remaining threats. [16, p. 165263], [21, p. 77], [22, p. 90].

6. Conclusions and Recommendations

6.1. Conclusions

This flow-risk estimation is combined with the SDC controller telemetry and path eligibility based on security with constrained multipath selection within the SA-ILB framework. The experimental comparison considers SA-ILB to be in the better side at the higher offered load, with benign goodput of 6.24 Gb/s and legitimate goodput of 3.45 Gb/s under a DDoS condition, and through the shared detector controls the effects in classification. They also report increased fairness, reduced tail delay and recovery time as compared to ECMP. The performance differences of the matched experimental runs have been further characterized by the confidence intervals and adjusted significance estimates.

6.2. Recommendations and Future Research Directions

Larger fat-trees (k = 8), new traffic mixes and attack distributions, together with online calibration, compromised telemetry, multi-controller failover, and explainable security constraints are all interesting settings for future research. The objective weights would benefit from a security–QoS Pareto frontier and sensitivity analysis to support deployment trade-offs. The study should ensure that all raw per-seed measures, detector-measurements and predictions, software versions, alterations to the models, scripts for analysis, and measures concerning permissions of the data sets are maintained for the sake of publication and of reproducing the study.

References

[1] M. Hamdan et al., “A comprehensive survey of load balancing techniques in software-defined network,” Journal of Network and Computer Applications, vol. 174, Art. no. 102856, 2021, doi: 10.1016/j.jnca.2020.102856.
[2] A. H. Alawadi and S. Molnár, “Oddlab: Fault-tolerant aware load-balancing framework for data center networks,” Annals of Telecommunications, vol. 77, pp. 641–662, 2022, doi: 10.1007/s12243-021-00898-0.
[3] T. Çavdar and Ş. Aymaz, “New approach to dynamic load balancing in software-defined network-based data centers,” ETRI Journal, vol. 45, no. 3, pp. 433–447, 2023, doi: 10.4218/etrij.2021-0478.
[4] J. C. Correa Chica, J. C. Imbachi, and J. F. Botero Vega, “Security in SDN: A comprehensive survey,” Journal of Network and Computer Applications, vol. 159, Art. no. 102595, 2020, doi: 10.1016/j.jnca.2020.102595.
[5] Y. Su, D. Xiong, K. Qian, and Y. Wang, “A comprehensive survey of distributed denial of service detection and mitigation technologies in software-defined network,” Electronics, vol. 13, no. 4, Art. no. 807, 2024, doi: 10.3390/electronics13040807.
[6] A. H. Janabi, T. Kanakis, and M. Johnson, “Survey: Intrusion detection system in software-defined networking,” IEEE Access, vol. 12, pp. 164097–164120, 2024, doi: 10.1109/ACCESS.2024.3493384.
[7] S. Faezi and A. Shirmarz, “A comprehensive survey on machine learning using in software defined networks (SDN),” Human-Centric Intelligent Systems, vol. 3, pp. 312–343, 2023, doi: 10.1007/s44230-023-00025-3.
[8] A. H. Alhilali and A. Montazerolghaem, “Artificial intelligence-based load balancing in SDN: A comprehensive survey,” Internet of Things, vol. 22, Art. no. 100814, 2023, doi: 10.1016/j.iot.2023.100814.
[9] V. Srivastava and R. S. Pandey, “Load balancing for software-defined network: A review,” International Journal of Computers and Applications, vol. 44, no. 8, pp. 746–759, 2022, doi: 10.1080/1206212X.2021.1919835.
[10] Y. Liu, J. Zhang, W. Li, Q. Wu, and P. Li, “Load balancing oriented predictive routing algorithm for data center networks,” Future Internet, vol. 13, no. 2, Art. no. 54, 2021, doi: 10.3390/fi13020054.
[11] C. Fancy and M. Pushpalatha, “Intelligence-enabled approach for load balancing in software-defined data center networks,” International Journal of Communication Systems, vol. 34, no. 9, Art. no. e4818, 2021, doi: 10.1002/dac.4818.
[12] Y.-R. Chen, A. Rezapour, W.-G. Tzeng, and S.-C. Tsai, “RL-Routing: An SDN routing algorithm based on deep reinforcement learning,” IEEE Transactions on Network Science and Engineering, vol. 7, no. 4, pp. 3185–3199, 2020, doi: 10.1109/TNSE.2020.3017751.
[13] Y. Zhang et al., “Multi-path routing algorithm based on deep reinforcement learning for SDN,” Applied Sciences, vol. 13, no. 22, Art. no. 12520, 2023, doi: 10.3390/app132212520.
[14] X. Pei, P. Sun, Y. Hu, D. Li, B. Chen, and L. Tian, “Enabling efficient routing for traffic engineering in SDN with deep reinforcement learning,” Computer Networks, vol. 241, Art. no. 110220, 2024, doi: 10.1016/j.comnet.2024.110220.
[15] Y. Wang, M. Othman, W. O. Choo, R. Liu, and X. Wang, “DFRDRL: A dynamic fuzzy routing algorithm based on deep reinforcement learning with guaranteed latency and bandwidth for software-defined networks,” Journal of Big Data, vol. 11, Art. no. 150, 2024, doi: 10.1186/s40537-024-01029-x.

[16] M. S. Elsayed, N.-A. Le-Khac, and A. D. Jurcut, "InSDN: A novel SDN intrusion dataset," IEEE Access, vol. 8, pp. 165263–165284, 2020, doi: 10.1109/ACCESS.2020.3022633.
[17] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, "NetFlow datasets for machine learning-based network intrusion detection systems," in Big Data Technologies and Applications (BDTA/WiCON), 2021, pp. 117–135, doi: 10.1007/978-3-030-72802-1_9.
[18] J. L. Leevy and T. M. Khoshgoftaar, "A survey and analysis of intrusion detection models based on CSE-CIC-IDS2018 big data," Journal of Big Data, vol. 7, Art. no. 104, 2020, doi: 10.1186/s40537-020-00382-x.
[19] V. G. S. Ruffo et al., "Anomaly and intrusion detection using deep learning for software-defined networks: A survey," Expert Systems with Applications, vol. 256, Art. no. 124982, 2024, doi: 10.1016/j.eswa.2024.124982.
[20] J. Chen et al., "Dynamic routing optimization in software-defined networking based on a metaheuristic algorithm," Journal of Cloud Computing, vol. 13, Art. no. 41, 2024, doi: 10.1186/s13677-024-00603-1.
[21] M. A. Setitra, M. Fan, I. Benkhaddra, and Z. E. A. Bensalem, "DoS/DDoS attacks in software defined networks: Current situation, challenges and future directions," Computer Communications, vol. 222, pp. 77–96, 2024, doi: 10.1016/j.comcom.2024.04.035.
[22] A. A. Wabi, I. Idris, O. M. Olaniyi, and J. A. Ojeniyi, "DDoS attack detection in SDN: Method of attacks, detection techniques, challenges and research gaps," Computers & Security, vol. 139, Art. no. 103652, 2024, doi: 10.1016/j.cose.2023.103652.
[23] U. H. Garba, A. N. Toosi, M. F. Pasha, and S. Khan, "SDN-based detection and mitigation of DDoS attacks on smart homes," Computer Communications, vol. 221, pp. 29–41, 2024, doi: 10.1016/j.comcom.2024.04.001.
[24] N. Yoon and H. Kim, "Detecting DDoS based on attention mechanism for software-defined networks," Journal of Network and Computer Applications, vol. 230, Art. no. 103928, 2024, doi: 10.1016/j.jnca.2024.103928.
[25] M. Sarhan, S. Layeghy, and M. Portmann, "Evaluating standard feature sets towards increased generalisability and explainability of ML-based network intrusion detection," Big Data Research, vol. 30, Art. no. 100359, 2022, doi: 10.1016/j.bdr.2022.100359.
[26] A. Prasad and S. Chandra, "Machine learning to combat cyberattack: A survey of datasets and challenges," The Journal of Defense Modeling and Simulation, vol. 20, no. 4, 2023, doi: 10.1177/15485129221094881.
[27] H. Feng et al., "Multi-domain collaborative two-level DDoS detection via hybrid deep learning," Computer Networks, vol. 242, Art. no. 110251, 2024, doi: 10.1016/j.comnet.2024.110251.
[28] M. D'Arienzo, "An evaluation of systems for detection and prevention of DoS attacks in SDN networks," International Journal of Wireless and Microwave Technologies, vol. 14, no. 3, pp. 31–49, 2024, doi: 10.5815/ijwmt.2024.03.03.
[29] V. K. A. Reddy and R. B. K. Nagappasetty, "Leveraging software-defined networks for load balancing in data centre networks using linear programming," International Journal of Computing, vol. 22, no. 3, pp. 404–411, 2023, doi: 10.47839/ijc.22.3.3237.
[30] I. F. Kilincer, F. Ertam, and A. Sengur, "Machine learning methods for cyber security intrusion detection: Datasets and comparative study," Computer Networks, vol. 188, Art. no. 107840, 2021, doi: 10.1016/j.comnet.2021.107840.