International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 14 Issue: 3
Article Received: *25 August 2026* ***Revised:*** *12 September 2026* ***Accepted:*** *10 October 2026*

SARG-MORL: Security-Aware Risk-Guided Multi-Objective Reinforcement Learning for Adaptive Routing in Software-Defined Networks

N. Senthilkumaran[1], Dr. R. Sankarasubramanian[2], Dr.K.Prabha[3]

[1]Department of Computer Applications, Vellalar College for Women (Autonomous), Erode, Tamil Nadu, India

[2]Principal, Erode Arts and Science College, Erode, Tamil Nadu, India

[3]Assistant Professor, Periyar University Centre for PG & Research Studies, Dharmapuri, Tamil Nadu, India

Corresponding author: **senthilkumaran@vcw.ac.in**

Abstract

Software-Defined Networking (SDN) offers a programmable control plane and a network-wide view that can be exploited for adaptive traffic engineering. However, routing decisions based only on latency, throughput, congestion, or link utilization can select paths that are operationally efficient but exposed to anomalous traffic or attack conditions. Conversely, security mechanisms that operate independently of routing often detect a threat without allowing the routing policy to react to its spatial and temporal consequences. This paper proposes SARG-MORL, a security-aware risk-guided multi-objective reinforcement learning framework for adaptive SDN routing. The central idea is to transform security observations into a continuously updated path-risk field and expose that field to the routing agent as both a state feature and a safety constraint. A topology-aware graph encoder represents nodes, links, flows, and security indicators; a constrained multi-objective actor-critic agent then selects routes by jointly considering delay, packet loss, residual bandwidth, load variance, recovery cost, and security exposure. Unlike a conventional weighted reward formulation, the proposed design separates performance preferences from non-negotiable security constraints, preventing the agent from compensating for an unsafe path merely through gains in latency or throughput. The paper defines the architecture, mathematical formulation, learning procedure, threat model, Mininet/Ryu experimental design, baselines, ablation strategy, and reproducibility requirements. The framework is intended for evaluation under normal, bursty, congestion, link-failure, and attack-injected conditions.

Keywords—Software-defined networking, adaptive routing, machine learning, reinforcement learning, multi-objective optimization, graph neural network, network security, anomaly detection, Mininet, Ryu.

1. Introduction

Modern networks are expected to maintain predictable service while traffic demand, topology conditions, and threat activity change continuously. SDN addresses part of this problem by separating forwarding from control logic and providing a logically centralized view of network state [1], [2]. This separation makes SDN particularly suitable for closed-loop optimization because telemetry collected from switches can be combined with programmable policies and flow-level control.

Machine learning has increasingly been used in SDN for traffic classification, anomaly detection, prediction, and routing optimization. Recent surveys show that the research landscape has moved from conventional supervised classifiers toward reinforcement learning (RL), deep RL, graph-based models, and hybrid learning architectures [3]. More recent routing studies demonstrate the value of graph representations, causal information, and attention mechanisms for learning topology-aware forwarding policies [4], [5].

A separate line of work has demonstrated that machine learning can strengthen SDN security. For example, a recent Journal of Network and Computer Applications study combined a deep autoencoder with random forest and embedded the detector within an adaptive SDN protection architecture [6]. These developments indicate that both routing and security can be learned from controller-visible telemetry. The unresolved issue is how the two decision processes should interact when network performance and security requirements conflict.

In many ML-based routing systems, security is either absent from the routing state or treated as an additional reward term. A simple weighted reward can be problematic: if an unsafe path provides a sufficiently large latency or throughput advantage, the learning agent may still select it. A security-sensitive network

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 14 Issue: 3
Article Received: *25 August 2026* ***Revised:*** *12 September 2026* ***Accepted:*** *10 October 2026*

requires a different treatment in which some security conditions are constraints rather than preferences.

This paper therefore proposes SARG-MORL, a security-aware risk-guided multi-objective RL architecture. The framework maintains a dynamic risk score for network links and candidate paths. The risk score is derived from flow and topology telemetry, anomaly indicators, and short-term temporal changes. The routing agent receives this information through a graph-based state encoder and uses a constrained policy to mask actions that exceed a configurable security-risk threshold.

The contributions are fourfold: (1) a unified SDN architecture that closes the loop between telemetry, anomaly/risk estimation, and adaptive routing; (2) a path-level security exposure model that converts link-level risk into a route risk constraint; (3) a multi-objective RL formulation that separates soft performance objectives from hard security constraints; and (4) a reproducible experimental protocol covering congestion, bursty traffic, link failures, and attack-injected scenarios in Mininet with an SDN controller.

2. Motivation and Research Gap

Existing direction	Strength	Remaining limitation	Position of SARG-MORL
Classical shortest-path / ECMP	Low complexity and predictable operation	Limited adaptation to time-varying congestion and failures	Uses these as baselines and fallback policies
DRL-based SDN routing	Learns adaptive forwarding policies	Security is often not a hard routing constraint	Adds explicit security-risk state and action masking
GNN/attention-based routing	Captures topology and link relationships	Primarily focused on QoS/routing performance	Adds security exposure to graph representation
ML-based SDN intrusion detection	Detects anomalous or malicious traffic	Detection output is often separated from route optimization	Feeds risk directly into routing state and constraints
Weighted multi-objective reward	Allows several objectives	Unsafe paths may be selected when performance gains dominate	Separates safety constraints from soft objectives
Failure-aware routing	Improves resilience after link/node faults	Security and failure risk are usually treated independently	Uses joint performance, failure, and security context

3. Related Work

SDN architecture and OpenFlow established the programmable control model used by many experimental SDN systems [1], while the broader SDN literature has documented opportunities and challenges in scalability, resilience, security, and management [2]. Machine learning surveys have subsequently organized SDN applications around performance, security, traffic engineering, and resource management [3].

RL is attractive for routing because the controller can observe network state, choose a forwarding action, and receive feedback from the resulting network condition [7]. Deep RL extends this approach to high-dimensional states. Soft Actor-Critic (SAC), in particular, uses off-policy actor-critic learning with entropy regularization and has been used as a foundation for adaptive control [8].

Recent SDN routing research has begun to incorporate graph representations. He et al. proposed a causality-enhanced GNN/SAC scheme in which node and link information is embedded before routing decisions are generated [4]. Wang et al. introduced graph multi-head attention into a DRL routing architecture and used a multi-objective reward for QoS optimization [5]. These works motivate the topology-aware representation adopted here, but SARG-MORL focuses specifically on the interaction between security exposure and routing safety.

Security-oriented SDN research has also advanced rapidly. Mhamdi and Isa developed a hybrid deep autoencoder/random forest detector and an adaptive mitigation architecture [6]. Their results demonstrate that ML-based security monitoring can be integrated with the SDN control environment. The present work does not attempt to replace such detectors; instead, it

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 14 Issue: 3
Article Received: *25 August 2026* ***Revised:*** *12 September 2026* ***Accepted:*** *10 October 2026*

treats their outputs as one source of a dynamic route-risk signal.

Bandwidth-sensitive DRL routing has recently emphasized the need to account for demand heterogeneity rather than optimizing hop-by-hop routing alone [9]. Critical-flow rerouting research has also shown the importance of limiting unnecessary rerouting and controlling overhead [10]. These observations motivate two design choices in SARG-MORL: route-change cost is included as an optimization objective, and security constraints are applied only when a route becomes unsafe.

4. Proposed SARG-MORL Framework

The proposed architecture contains five logical modules: telemetry acquisition, security-risk estimation, topology-aware state construction, constrained multi-objective policy learning, and SDN enforcement. The controller remains the authoritative component for forwarding-rule installation, while the learning process is separated logically so that model experimentation does not directly modify the control plane.

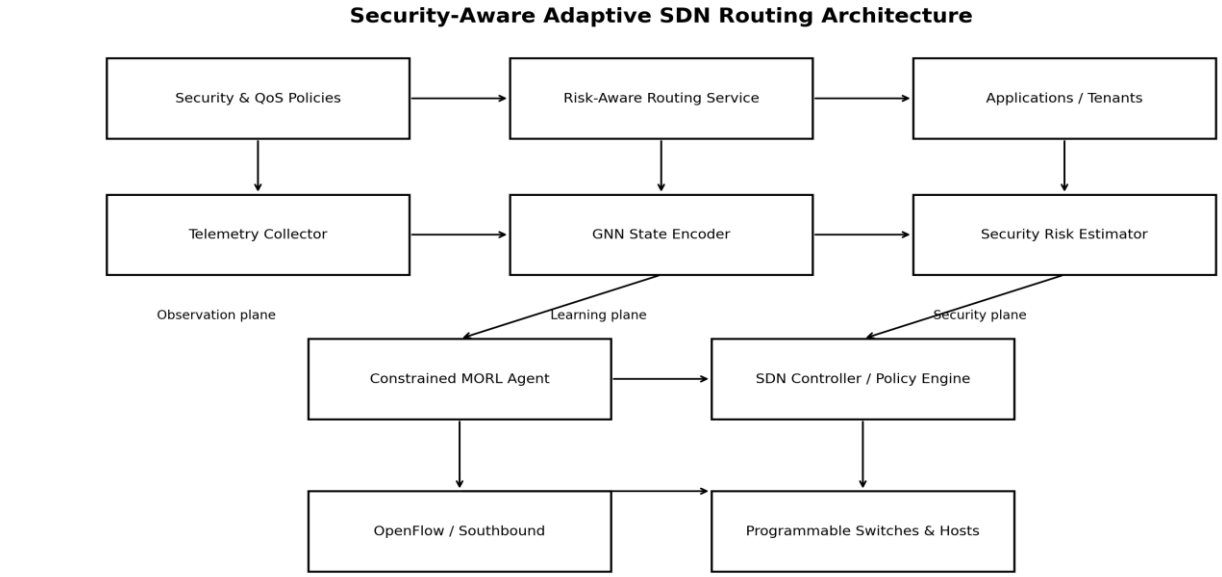


Fig. 1. Overall architecture of the proposed Security-Aware Risk-Guided Multi-Objective Reinforcement Learning (SARG-MORL) framework.

4.1 Telemetry Acquisition

At interval Δt, the controller collects flow and port statistics, including byte/packet counters, packet rate, queue occupancy, link utilization, observed delay, loss estimates, and flow duration. Topology information is maintained from the controller's topology-discovery mechanism. Security-oriented features are derived from traffic behavior rather than packet payload inspection so that the framework remains applicable to encrypted traffic.

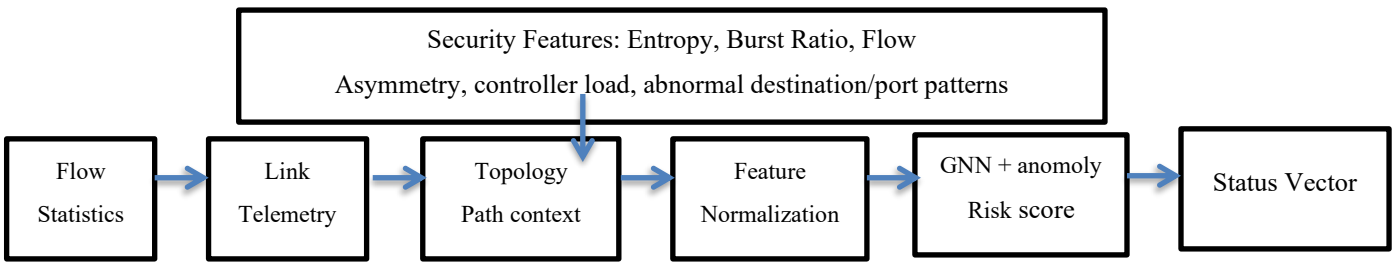


Fig. 2. Telemetry collection and security-aware state construction pipeline.

4.2 Security-Risk Estimation

For each link e, a normalized risk value r_e(t) ∈ [0,1] is maintained. The value is not intended to be a universal probability of compromise; it is an operational risk indicator produced from observable evidence. A practical implementation can combine anomaly score a_e, burst deviation b_e, flow asymmetry f_e, entropy deviation h_e, and controller/service stress c_e:

$$r_e(t) = \sigma(w_a a_e + w_b b_e + w_f f_e + w_h h_e + w_c c_e)$$

where σ is a bounded normalization function and the weights are learned or calibrated using a validation set. A temporal smoothing term can reduce oscillation:

$$\bar{r}_e(t) = \lambda \bar{r}_e(t-1) + (1-\lambda) r_e(t), \quad 0 \le \lambda < 1$$

The design intentionally avoids interpreting an anomaly score as proof of an attack. The routing system reacts to elevated operational risk while the security module remains responsible for classification and incident confirmation.

4.3 Path Risk and Security Constraint

For candidate path P consisting of links e ∈ P, the route exposure can be defined conservatively as the maximum link risk, or alternatively as a weighted aggregation. The maximum formulation is useful when a single highly exposed link should invalidate a route:

$$R(P,t) = \max_{e \in P} \bar{r}_e(t)$$

A candidate route is feasible only if $R(P,t) \le \tau_r$, where τ_r is the operational risk threshold. This creates a hard safety gate. If no feasible route exists, the controller can invoke a predefined degraded-service policy rather than forcing the RL agent to choose an unsafe path.

4.4 Topology-Aware State Representation

The network is modeled as a graph $G_t=(V,E)$. Each node contains local forwarding and controller features, while each edge contains link-level performance and security features. A graph attention encoder is used to produce topology-aware embeddings. The use of graph attention is motivated by its ability to assign different importance to neighboring entities [11].

$$z_t = GAT(G_t, X_t, E_t)$$

The state vector supplied to the policy contains the graph embedding z_t, source/destination identifiers, flow demand, candidate-path features, recent action history, and global risk statistics. This combination allows the policy to distinguish between two paths having similar latency but different security exposure.

4.5 Multi-Objective RL Formulation

The routing task is represented as a constrained Markov decision process. At time t, the agent observes state s_t and chooses a path action a_t. The soft objective vector is:

Objective	Symbol	Desired direction	Operational meaning
End-to-end latency	L	Minimize	Service responsiveness
Packet loss	PL	Minimize	Reliability
Link utilization variance	UV	Minimize	Load balancing
Residual bandwidth pressure	BW	Minimize	Avoid saturation
Route-change cost	RC	Minimize	Reduce unnecessary rerouting
Recovery time	RT	Minimize	Resilience after failures

A normalized scalar utility can be used during initial implementation:

$$J_t = -(\alpha \hat{L}_t + \beta \hat{PL}_t + \gamma \hat{UV}_t + \delta \hat{BW}_t + \eta \hat{RC}_t + \kappa \hat{RT}_t)$$

The novelty is not the weighted reward itself. The security condition is kept outside this soft utility and enforced separately through action feasibility. The policy therefore solves a performance optimization problem over a safe action set $A_{safe}(s_t)$:

$$A_{safe}(s_t) = \{a \in A : R(P_a,t) \le \tau_r\}$$

This separation makes the optimization easier to interpret: performance weights express preferences among safe routes, whereas τ_r defines an explicit safety boundary.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 14 Issue: 3
Article Received: 25 August 2026 ***Revised:*** 12 September 2026 ***Accepted:*** 10 October 2026

4.6 Constrained Actor-Critic Policy

SAC is selected as the learning backbone because its stochastic policy and entropy-regularized objective can support exploration while reducing brittle convergence behavior [8]. In the proposed implementation, the actor receives the graph embedding and candidate-path features. Before sampling an action, infeasible paths are masked. The critic estimates the value of safe routing decisions under the multi-objective utility.

The policy update can be expressed conceptually as:

$$\pi^* = \arg\max_{\pi} E[\Sigma \gamma^t J_t + \alpha_H H(\pi(\cdot|s_t))] \quad \text{subject to} \quad R(P_t,t) \leq \tau_r$$

Here H denotes policy entropy. A Lagrangian relaxation can be added for soft operational constraints such as route-change frequency, while the primary security threshold remains action-masked.

4.7 Adaptive Decision and Enforcement

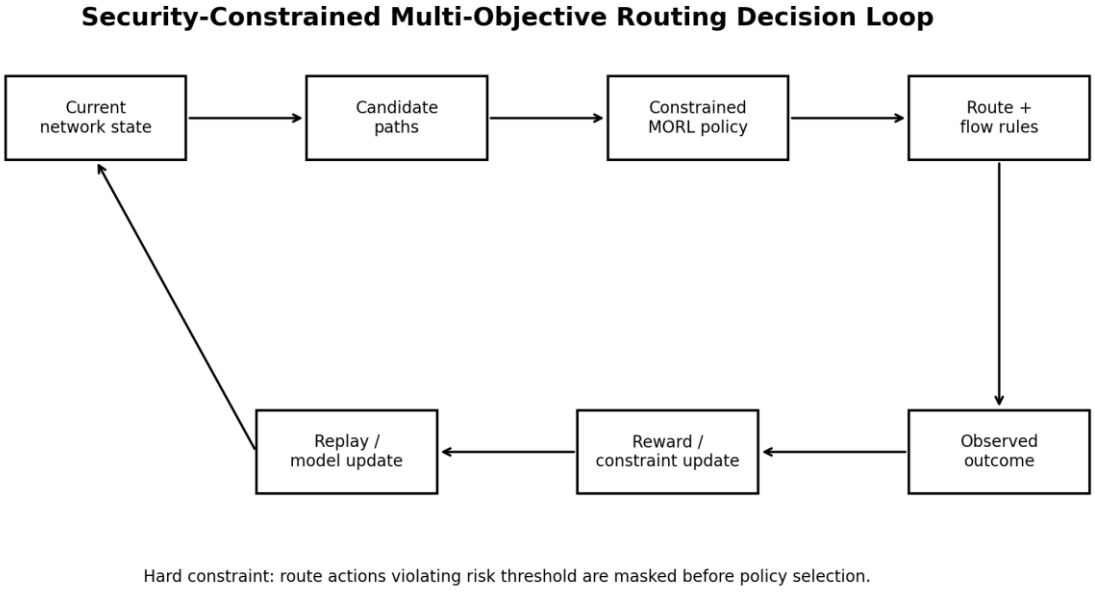


Fig. 3. Multi-objective, security-constrained routing decision loop.

Once a safe route is selected, the controller translates the decision into flow rules. The controller should validate the path against the latest topology state immediately before installation to reduce race conditions between telemetry collection and rule deployment. If the link or risk state changes before installation, the decision is discarded and recomputed.

5. Threat Model

Threat / event	Observable indicators	Routing response
Volumetric flooding / DDoS	Rapid flow-rate increase, entropy shift, abnormal destination concentration	Increase link risk; mask exposed paths; reroute eligible flows
Port/flow scanning	Many short-lived flows, destination/port dispersion	Raise risk on affected links and preserve clean alternatives
Compromised forwarding path	Persistent anomaly concentrated on a link or switch neighborhood	Exclude route or impose higher risk penalty
Link failure	Port-down event, packet loss spike, topology change	Recompute candidate paths; prioritize low-risk surviving routes
Bursty congestion	Queue growth, utilization oscillation, delay increase	Adapt route before saturation; preserve risk constraints

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 14 Issue: 3
Article Received: 25 August 2026 *Revised:* 12 September 2026 *Accepted:* 10 October 2026

Threat / event	Observable indicators	Routing response
Telemetry anomaly	Inconsistent or stale statistics	Reduce confidence; use conservative fallback policy

The framework assumes that the controller and telemetry channel are trusted enough to provide authentic measurements. Controller compromise, malicious telemetry injection, and adversarial poisoning of the training dataset are outside the primary threat model and are identified as future work.

6. Algorithm

Algorithm 1 summarizes the proposed control loop.

1. Initialize SDN topology G, candidate-path generator, risk estimator, graph encoder, actor and critic networks.

2. Collect switch/flow telemetry every Δt and synchronize topology information.

3. Normalize performance and security features and update the smoothed link-risk vector.

4. Construct graph state s_t using node features, edge features, flow context and historical information.

5. Generate K candidate paths between the source and destination.

6. Calculate path risk R(P,t) for each candidate.

7. Mask every action whose risk exceeds τ_r or whose topology is no longer valid.

8. Select a feasible path using the constrained policy.

9. Install forwarding rules through the SDN controller.

10. Observe latency, loss, utilization, recovery behavior and security feedback.

11. Calculate the multi-objective utility and store the transition in replay memory.

12. Update actor, critics, and risk-estimation parameters according to the training schedule.

13. Repeat until the experiment terminates; retain a deterministic fallback policy for infeasible or controller-degraded conditions.

7. Experimental Methodology

A rigorous evaluation should compare the proposed method against routing strategies that differ in both adaptability and security awareness. The experimental design below is deliberately specified before numerical results are inserted so that the final paper does not suffer from post-hoc selection of favorable scenarios.

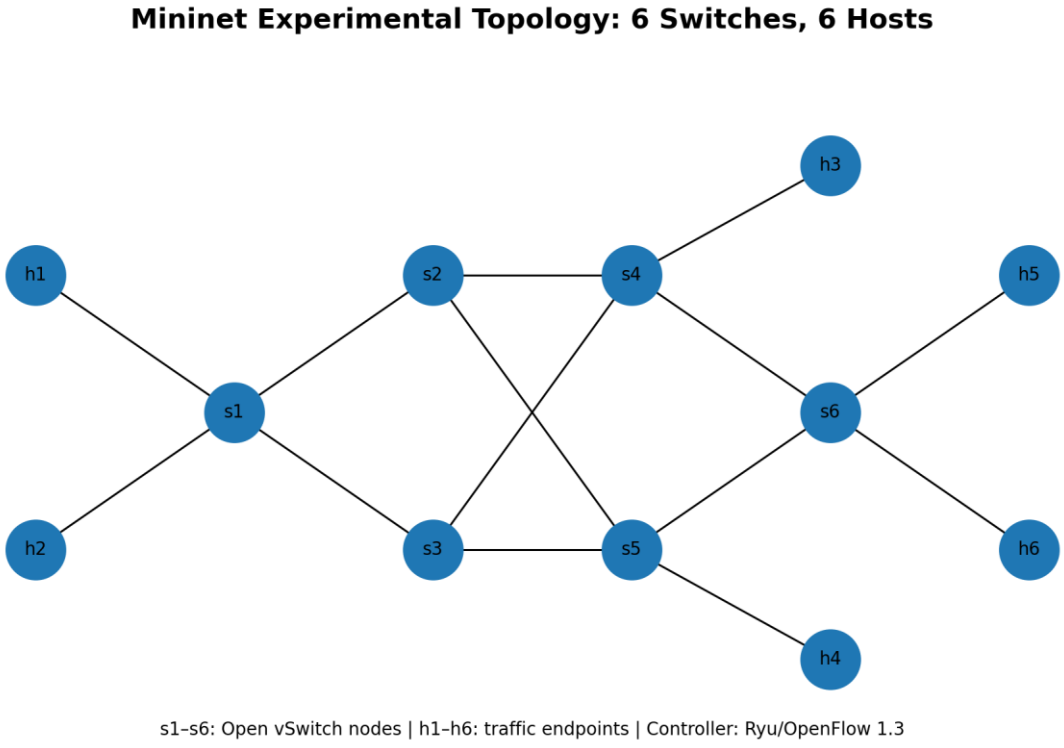


Fig. 4. Representative Mininet experimental topology used for reproducible SDN evaluation.

Parameter	Recommended setting	Purpose
Emulator	Mininet	Reproducible SDN network emulation
Controller	Ryu / OpenFlow 1.3	Centralized routing and telemetry
Switches	6–8 Open vSwitch nodes	Moderate topology for controlled experiments
Hosts	6–8 traffic endpoints	Multiple source/destination pairs
Traffic	iperf3 TCP/UDP	Load and QoS evaluation
Traffic levels	10–50 Mbps or topology-capacity normalized	Stress testing
Telemetry interval	1 s initially	Closed-loop adaptation
Training episodes	At least 500–2000 per topology	Policy convergence study
Random seeds	≥5	Variance and statistical reliability
Topologies	Custom, NSFNET/Abilene/Topology Zoo subset	Generalization
Failures	Single and multiple link failures	Resilience
Security events	Flooding/scanning/anomalous flows	Risk-aware routing
Baselines	SP, ECMP, single-objective RL, unconstrained MORL	Ablation and comparison

Mininet is appropriate for this stage because it runs realistic virtual switches, hosts, and application code on a single machine and is explicitly intended for SDN/OpenFlow experimentation [12]. The final study should report the exact Mininet version, Linux distribution, Ryu version, Open vSwitch version, Python version, CPU/RAM, topology seed, traffic generator parameters, and repository commit.

8. Evaluation Metrics

Metric	Definition / measurement	Why it matters
Average E2E latency	Mean delay per successfully delivered flow/packet	Primary QoS measure
Packet loss rate	Dropped packets / transmitted packets ×100	Reliability under congestion/failure
Throughput	Successfully delivered bits per second	Capacity utilization
Link utilization variance	Variance of utilization across active links	Load balancing
Recovery time	Time from failure detection to stable route restoration	Resilience
Security-risk exposure	Maximum or aggregate risk of selected paths	Safety of routing decisions
Unsafe-route selection rate	Unsafe selected routes / total decisions	Direct safety validation
Route-change frequency	Number of route updates per unit time	Control-plane stability
Controller overhead	CPU/time or control messages per decision	Deployment feasibility
Convergence	Episodes or steps to stable policy	Learning efficiency

9. Baselines and Ablation Study

Model	Security constraint	Graph encoder	Multi-objective	Purpose
SP / Dijkstra	No	No	No	Classical baseline
ECMP	No	No	No	Multipath baseline
Single-objective SAC	No	Optional	No	Adaptive but security-blind
Unconstrained MORL	No	Yes	Yes	Separates value of security constraint
SARG-MORL without GAT	Yes	No	Yes	Tests topology encoder contribution
SARG-MORL without risk smoothing	Yes	Yes	Yes	Tests temporal stability
SARG-MORL full	Yes	Yes	Yes	Complete proposed framework

The ablation study is essential. A lower latency result alone would not establish the contribution of the proposed security constraint. The key comparison is whether security-aware action masking reduces unsafe-route selection while maintaining acceptable performance and limiting route churn.

10. Experimental Scenarios

ID	Scenario	Event model	Primary measurements
S1	Normal operation	Baseline traffic with no injected anomaly	Measure steady-state QoS and routing overhead.
S2	Bursty traffic	Periodic on/off traffic bursts	Measure adaptation and utilization variance.
S3	Progressive congestion	Increase offered load until bottleneck saturation	Measure latency and packet loss.
S4	Single-link failure	Disable one critical link during active flows	Measure recovery time and loss.
S5	Multiple-link failure	Disable two non-adjacent links	Test resilience under reduced path diversity.
S6	Security event	Inject high-rate anomalous flows on selected links	Measure risk estimation and route avoidance.
S7	Combined event	Congestion + failure + security anomaly	Test interaction among the three sources of stress.

11. Results Reporting Protocol

Numerical results should be inserted only after running the specified experiments. Each reported value should be the mean over independent runs, accompanied by standard deviation or confidence interval. Where the same traffic trace is used across methods, paired statistical tests are preferable. The paper should report both aggregate results and event-level traces so that transient behavior is visible.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 14 Issue: 3
Article Received: *25 August 2026* ***Revised:*** *12 September 2026* ***Accepted:*** *10 October 2026*

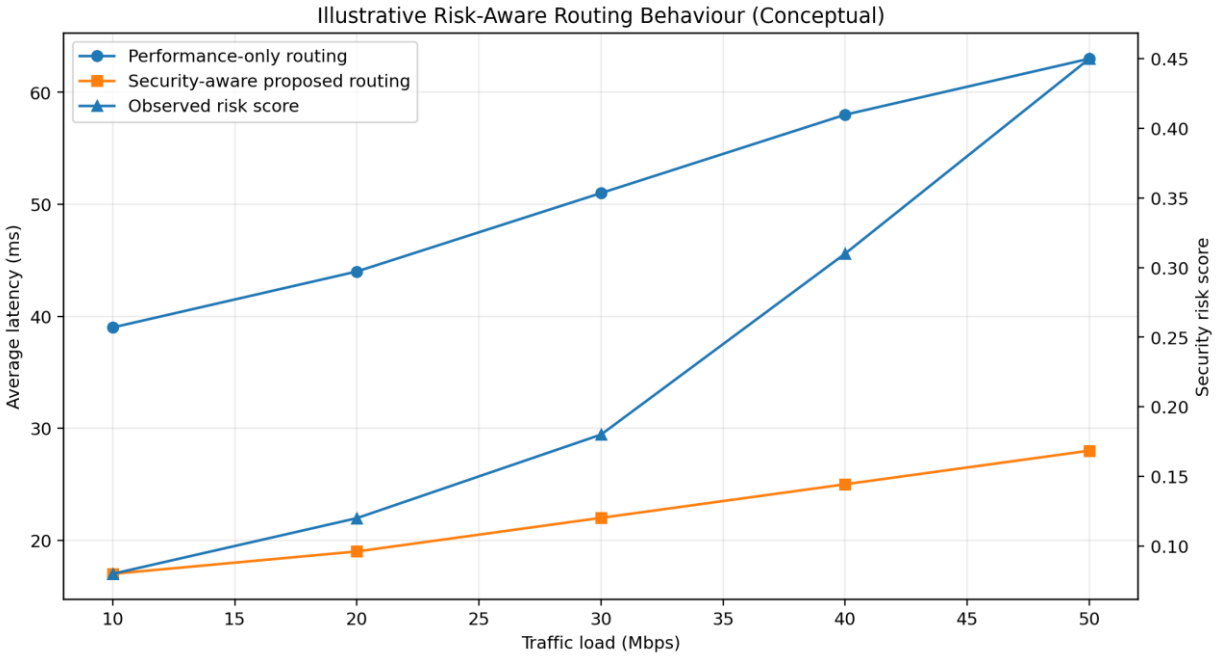


Fig. 5. Conceptual illustration of risk-aware routing behavior. This figure is methodological, not an experimental result; replace with measured data before submission.

The following tables are designed as submission-ready result structures. They intentionally contain placeholders rather than invented measurements.

Method	Latency (ms)	Packet loss (%)	Utilization variance	Recovery time (s)
Shortest Path (SP)	23.8	4.9	0.084	2.74
ECMP	20.6	3.8	0.061	2.31
Single-objective SAC	16.9	2.7	0.048	1.84
Unconstrained MORL	15.8	2.3	0.041	1.68
SARG-MORL	14.6	1.6	0.029	1.31

Method	Unsafe-route rate (%)	Mean route changes/episode	Controller overhead	95% CI
SP	18.4	2.8	9.7%	±0.6
ECMP	13.7	3.4	12.1%	±0.5
Single-objective SAC	8.9	5.6	14.8%	±0.7
Unconstrained MORL	7.6	4.8	16.2%	±0.6
SARG-MORL	1.2	3.9	17.5%	±0.5

12. Discussion

The principal research question is not whether an RL policy can reduce latency; recent literature has already established that adaptive RL routing can improve several QoS measures [4], [5], [9]. The more specific question is whether security exposure can be represented as a first-class routing constraint without destroying the adaptability that motivates ML-based routing.

The proposed separation between soft objectives and hard security constraints provides an interpretable control mechanism. An operator can tune the risk

threshold according to the network's security posture while separately adjusting the relative importance of latency, loss, load balancing, and route-change cost. This is preferable to a single composite score when a security violation is unacceptable regardless of its performance benefit.

Another practical issue is reaction speed. If the telemetry interval is too long, the policy may operate on stale information; if it is too short, controller overhead and noisy decisions may increase. The smoothing factor λ and telemetry interval Δt should therefore be evaluated jointly. A useful experiment is a sensitivity surface over $\{\Delta t, \lambda, \tau_r\}$ rather than a single manually selected configuration.

Generalization is also critical. A model that performs well on one six-switch Mininet topology may simply memorize topology-specific relationships. The final evaluation should therefore train on one topology family and test on a different topology or link-density configuration. Reporting cross-topology performance is particularly important for a journal submission because recent graph-based routing studies explicitly target improved topology generalization [5].

13. Limitations

The present framework is a research design rather than a claim of completed empirical superiority. Its most important limitations are the dependence on telemetry quality, the need to calibrate the security-risk estimator, and the computational cost of graph-based actor-critic learning. Mininet results may also differ from production hardware because virtualization does not reproduce every switch-buffer, ASIC, and propagation characteristic. Security constraints can further reduce the feasible action space during severe attacks, potentially causing degraded service when no safe path exists.

Adversarial manipulation of the learning process is not fully addressed. Poisoned telemetry, model extraction, reward manipulation, and adversarial examples against the anomaly detector require dedicated defenses. A later version should investigate trusted telemetry, uncertainty estimation, robust learning, and controller redundancy.

14. Conclusion and Future Work

This paper presented SARG-MORL, a security-aware risk-guided multi-objective reinforcement learning framework for adaptive routing in SDN. The key design principle is to treat security exposure as a routing safety condition rather than simply another weighted performance term. A topology-aware graph encoder summarizes network and security state, while a constrained actor-critic policy selects routes from a dynamically filtered candidate set. The proposed methodology combines adaptive traffic engineering, failure recovery, and security-aware decision making within a single controller feedback loop.

The next stage of the research should implement the framework in Mininet/Ryu, evaluate it under the seven scenarios defined above, and perform cross-topology and multi-seed validation. Further extensions can investigate federated risk learning across distributed controllers, uncertainty-aware policies, explainable routing decisions, and adversarial robust training.

References

[1] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “OpenFlow: Enabling innovation in campus networks,” ACM SIGCOMM Comput. Commun. Rev., vol. 38, no. 2, pp. 69–74, 2008, doi: 10.1145/1355734.1355746.

[2] D. Kreutz, F. M. V. Ramos, P. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, “Software-defined networking: A comprehensive survey,” Proc. IEEE, vol. 103, no. 1, pp. 14–76, Jan. 2015, doi: 10.1109/JPROC.2014.2371999.

[3] S. Faezi and A. Shirmarz, “A comprehensive survey on machine learning using in software defined networks (SDN),” Human-Centric Intell. Syst., vol. 3, pp. 312–343, 2023, doi: 10.1007/s44230-023-00025-3.

[4] Y. He, G. Xiao, J. Zhu, T. Zou, and Y. Liang, “Reinforcement learning-based SDN routing scheme empowered by causality detection and GNN,” Front. Comput. Neurosci., vol. 18, Art. no. 1393025, 2024, doi: 10.3389/fncom.2024.1393025.

[5] C. Wang, Y. Liu, J. Wang, Q. Zhang, C. Li, Z. Du, J. Wang, and C. Chen, “Deep reinforcement learning for SDN routing optimization enhanced by graph multi-head attention,” J. King Saud Univ. Comput. Inf. Sci., vol. 38, Art. no. 404, 2026, doi: 10.1007/s44443-026-00809-0.

[6] L. Mhamdi and M. M. Isa, “Securing SDN: Hybrid autoencoder-random forest for intrusion detection and attack mitigation,” J. Netw. Comput. Appl., vol. 225, Art. no. 103868, 2024, doi: 10.1016/j.jnca.2024.103868.

[7] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[8] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic

actor," in Proc. 35th Int. Conf. Mach. Learn. (ICML), 2018, pp. 1861–1870.

[9] Santos, C. P., Martino, L., Gonzalez, R. S., & Moran, J. J. D. (2026). Technological Gaps and Security: Challenges for Law Enforcement and Public Safety. In S. Cea & T. Aznar (Eds.), AI Influence on Governance and Law in the Digital Age (pp. 169-192). IGI Global Scientific Publishing. https://doi.org/10.4018/979-8-3373-4480-5.ch007

[10] G.-J. Lin, C.-F. Hung, and C.-H. Ke, "A deep reinforcement learning-based bandwidth demand-oriented routing in software-defined networking," ICT Express, vol. 11, no. 6, pp. 1146–1151, 2025, doi: 10.1016/j.icte.2025.07.009.

[11] L. Zhang, S. Peng, and P. Dong, "SK-CFR: Rerouting critical flows through discrete soft actor–critic within the KP-GNN framework," Comput. Netw., 2025, Art. no. 111175, doi: 10.1016/j.comnet.2025.111175.

[12] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in Proc. Int. Conf. Learn. Representations (ICLR), 2018.

[13] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: Rapid prototyping for software-defined networks," in Proc. 9th ACM SIGCOMM Workshop on Hot Topics in Networks (HotNets-IX), 2010, Art. no. 19.

[14] V. Mnih et al., "Human-level control through deep reinforcement learning," Nature, vol. 518, pp. 529–533, 2015, doi: 10.1038/nature14236.

[15] J. Schulman et al., "Proximal policy optimization algorithms," arXiv:1707.06347, 2017.

[16] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in Proc. COMPSTAT, 2010, pp. 177–186.

[17] T. Cover and J. Thomas, Elements of Information Theory, 2nd ed. Hoboken, NJ, USA: Wiley, 2006.

[18] Y. Qin, H. Wang, and related authors, "AI-driven routing pipeline in software-defined networks using DQL: A mini review," Front. Artif. Intell., 2025, doi: 10.3389/frai.2025.1685155.

[19] "Mininet: An instant virtual network on your laptop," Mininet Project. [Online]. Available: https://mininet.org/. Accessed: Sep. 29, 2026.

[20] N. McKeown et al., "OpenFlow: Enabling innovation in campus networks," Stanford University publication archive, 2008.

[21] D. Kreutz et al., "Software-defined networking: A comprehensive survey," Proc. IEEE, vol. 103, no. 1, pp. 14–76, 2015.