



International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X
Impact Factor: 7.056

Artificial Intelligence for MPLS and Segment Routing Traffic Engineering: A Systematic Review of Machine Learning, Reinforcement Learning, and Deep Reinforcement Learning

Mustafa ElGili Mustafa

Applied College- Huraymila, Computer Programs, Imam Muhammad Ibn Saud University, Saudi Arabia
meelmogadem@imamu.edu.sa

DOI: <https://doi.org/10.47760/ijcsmc.2026.v15i10.002>

Abstract: The growing complexity and variability of network traffic have increased the demand for adaptive traffic engineering in Multiprotocol Label Switching (MPLS) and Segment Routing networks. Conventional approaches, including constraint-based routing, mathematical programming, and heuristic optimization, provide established mechanisms for path selection and resource allocation but may encounter computational and operational limitations under dynamic network conditions. Artificial Intelligence (AI), particularly Machine Learning (ML), Reinforcement Learning (RL), and Deep Reinforcement Learning (DRL), offers alternative approaches to traffic prediction, adaptive routing, congestion management, and policy optimization. This paper presents a systematic literature review framework for AI-assisted traffic engineering in MPLS and Segment Routing networks, guided by PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses.) 2020. The review aims to classify intelligent optimization techniques, examine their integration with forwarding and control architectures, and critically evaluate their effectiveness against conventional routing approaches. Key evaluation dimensions include link utilization, throughput, latency, packet loss, computational overhead, routing stability, generalization, and constraint satisfaction. Particular emphasis is placed on experimental methodologies, baseline fairness, reproducibility, and the distinction between simulation-based improvements and practical deployment feasibility. The proposed synthesis seeks to identify methodological limitations, unresolved technical challenges, and future research opportunities for developing scalable, reliable, and operationally feasible AI-assisted traffic engineering solutions.

Keywords: Multiprotocol Label Switching; Segment Routing; Traffic Engineering; Machine Learning; Reinforcement Learning; Deep Reinforcement Learning; Routing Optimization; Systematic Literature Review



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

I. INTRODUCTION

A. *Background and Research Motivation*

The increasing complexity and variability of communication networks have intensified the need for efficient and adaptive Traffic Engineering (TE). Modern infrastructures support heterogeneous applications, including cloud computing, real-time multimedia, enterprise connectivity, and latency-sensitive services, each with different requirements for bandwidth, delay, packet loss, and reliability. Traffic engineering addresses these challenges through congestion management, path optimization, resource allocation, and service-aware forwarding [1].

Multiprotocol Label Switching (MPLS) provides an established architecture for forwarding traffic through Label Switched Paths (LSPs), enabling explicit traffic steering and network resource management [2]. Conventional MPLS Traffic Engineering (MPLS-TE) employs mechanisms such as Resource Reservation Protocol–Traffic Engineering (RSVP-TE) and Constraint-Based Shortest Path First (CSPF) to establish paths according to specified network constraints [3]. However, their effectiveness depends on path-computation algorithms, network-state accuracy, and the operational cost of modifying forwarding decisions.

Segment Routing (SR) extends path-control flexibility through ordered forwarding instructions [4]. When implemented over the MPLS data plane, Segment Routing over MPLS (SR-MPLS) encodes these instructions using MPLS label stacks [5], while SR Policies support traffic steering through selected candidate paths [6]. Nevertheless, these architectures do not independently determine optimal traffic allocation; efficient path selection remains dependent on network conditions, optimization objectives, and forwarding constraints.

The increasing variability of traffic demand has therefore motivated research into Artificial Intelligence (AI)-assisted traffic engineering. Machine Learning (ML), Reinforcement Learning (RL), and Deep Reinforcement Learning (DRL) offer opportunities for traffic prediction, adaptive routing, congestion management, and data-driven optimization [7]. However, their effectiveness in MPLS and SR-MPLS networks requires systematic evaluation against established optimization approaches and realistic operational requirements.

B. *Limitations of Conventional MPLS Traffic Engineering*

Conventional MPLS traffic engineering relies on constraint-based routing, Linear Programming (LP), Integer Linear Programming (ILP), Mixed-Integer Linear Programming (MILP), multicommodity-flow optimization, and heuristic or metaheuristic algorithms. These methods provide explicit optimization objectives, interpretable decision procedures, and mechanisms for enforcing network constraints.

Nevertheless, several limitations arise in large-scale or dynamic environments. First, combinatorial routing problems may require substantial computational resources as topology size, traffic demands, and path alternatives increase. Second, optimization quality depends on accurate information about traffic matrices, link capacities, and forwarding state, which may be delayed or incomplete. Third, frequent reoptimization can introduce signaling overhead, policy-installation delays, packet reordering, and routing instability. Finally, traffic engineering objectives may conflict; minimizing link utilization does not necessarily minimize latency or service disruption.

These challenges do not render classical optimization ineffective. Mathematical programming and heuristic methods remain essential for network planning, resource management, and comparative benchmarking. Instead, their limitations motivate complementary approaches that exploit historical measurements and learned relationships to support timely routing decisions under changing conditions.

C. *Artificial Intelligence in MPLS and Segment Routing*

AI-based traffic engineering introduces learning mechanisms for analyzing network conditions and improving path-selection decisions. Its principal applications include traffic prediction, congestion identification, adaptive routing, resource allocation, and forwarding-policy optimization.

Supervised ML can estimate traffic demand, link utilization, and congestion using historical measurements, while unsupervised learning can identify traffic patterns, operating states, and anomalous network behavior.



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

These capabilities may improve traffic engineering inputs, although prediction accuracy does not independently establish better routing performance.

RL formulates traffic engineering as a sequential decision-making problem in which an agent observes network conditions, selects routing actions, and learns through performance-based rewards. DRL extends this capability using neural-network function approximation, enabling more complex representations of network states and routing decisions. Xiao et al. [7] surveyed DRL-based traffic engineering applications across routing optimization, congestion control, and resource management.

More specifically, Tian et al. [8] investigated adaptive Segment Routing traffic engineering using DRL to support online routing decisions under changing network conditions. Such research illustrates the potential integration of intelligent optimization with segment-based path control.

However, AI-based routing decisions must remain compatible with the underlying forwarding architecture. In MPLS environments, this includes LSP feasibility, resource reservation, and signaling requirements. SR-MPLS introduces additional considerations involving Segment Identifier availability, segment-list depth, and forwarding-platform capabilities. Therefore, AI should be regarded as an optimization capability complementing MPLS and Segment Routing rather than replacing their forwarding and control mechanisms.

D. Research Gap and Novelty

Existing surveys have examined intelligent traffic engineering and its applications to modern network architectures. Xiao et al. [7] reviewed DRL-based approaches to routing optimization, congestion management, and resource allocation, while Hassan et al. [9] examined AI techniques for enhancing Segment Routing performance.

Despite these contributions, there remains a need for a focused synthesis examining the relationship among learning-based optimization methods, MPLS/SR-MPLS forwarding architectures, and the quality of supporting experimental evidence.

Five analytical challenges motivate the present review.

First, existing research employs different forwarding environments, including conventional MPLS LSPs, SR Policies, SDN-assisted networks, and abstract simulation models. Their results require architectural classification to distinguish directly demonstrated MPLS performance from broader networking evidence.

Second, reported improvements may depend on weak or unequal comparison baselines. Superior performance relative to shortest-path routing does not necessarily establish an advantage over optimized CSPF, mathematical programming, or competitive heuristic methods.

Third, computational assessments may emphasize online inference while overlooking training, dataset preparation, hyperparameter optimization, feasibility validation, and retraining costs.

Fourth, practical deployment requires routing feasibility, compliance with network constraints, stable policy installation, and compatibility with forwarding hardware. Favorable simulation outcomes do not independently establish these properties.

Fifth, evidence comparability is limited by differences in network topologies, traffic matrices, simulation environments, experimental repetition, parameter reporting, and reproducibility.

Accordingly, the proposed contribution is an integrated analytical framework that evaluates AI-assisted MPLS and SR-MPLS traffic engineering across three dimensions: optimization methodology, forwarding and control architecture, and experimental evidence quality. This perspective is intended to complement existing surveys by distinguishing algorithmic performance from architectural compatibility and practical deployment feasibility.

E. Research Objectives and Research Questions

The primary objective of this systematic review is to investigate the application of ML, RL, and DRL to MPLS and Segment Routing traffic engineering and critically assess their effectiveness relative to established mathematical, heuristic, and rule-based optimization approaches.



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

The review aims to classify intelligent traffic engineering methods, examine their integration with MPLS and SR-MPLS architectures, compare reported performance against conventional baselines, evaluate computational and operational requirements, assess methodological rigor, and identify unresolved research challenges.

The following research questions guide the review:

RQ1: Which ML, RL, and DRL techniques have been investigated for traffic prediction, routing optimization, resource allocation, and adaptive control in MPLS and Segment Routing networks?

RQ2: How are AI-based optimization methods integrated with conventional MPLS, SR-MPLS, SDN-assisted control, and Path Computation Element architectures?

RQ3: To what extent do AI-based techniques improve link utilization, throughput, delay, packet loss, and other traffic engineering outcomes compared with classical mathematical and heuristic baselines?

RQ4: What computational, training, scalability, and generalization challenges affect intelligent MPLS and Segment Routing traffic engineering?

RQ5: How do existing approaches address routing feasibility, operational constraints, policy-installation overhead, and network stability?

RQ6: What simulation platforms, topologies, traffic datasets, experimental procedures, and statistical methods are used, and how reproducible are the reported results?

RQ7: What methodological weaknesses, deployment limitations, and unresolved challenges should guide future research on reliable AI-assisted MPLS traffic engineering?

These research questions establish a framework for evaluating both the algorithmic effectiveness and operational feasibility of intelligent traffic engineering.

F. Paper Organization

The remainder of this paper is organized as follows. Section II describes the systematic review methodology, including the search strategy, eligibility criteria, screening procedures, data extraction, and quality assessment. The methodology is designed to follow PRISMA 2020 reporting guidance [10], with completed review procedures and selection outcomes documented when available.

Section III introduces the technical foundations of MPLS, SR-MPLS, conventional traffic engineering, and programmable network control. Section IV classifies ML, RL, DRL, and hybrid intelligent optimization approaches. Section V critically compares intelligent techniques with classical traffic engineering methods across performance, computational efficiency, scalability, robustness, and operational feasibility. Section VI identifies research gaps, while Section VII outlines future research directions. Section VIII presents the review limitations, and Section IX concludes the paper.

II. SYSTEMATIC REVIEW METHODOLOGY

This study is designed as a systematic literature review of Artificial Intelligence (AI)-based traffic engineering in Multiprotocol Label Switching (MPLS) and Segment Routing networks. It focuses on Machine Learning (ML), Reinforcement Learning (RL), Deep Reinforcement Learning (DRL), and hybrid intelligent optimization methods applied to traffic prediction, routing optimization, congestion management, resource allocation, and adaptive network control.

The methodology emphasizes transparent study identification, reproducible screening, structured evidence extraction, and critical evaluation of experimental results. Particular attention is given to distinguishing research that explicitly implements or models MPLS and Segment Routing over MPLS (SR-MPLS) from broader investigations involving generic IP routing, Software-Defined Networking (SDN), or Segment Routing over IPv6 (SRv6).

The review is designed to follow PRISMA 2020 reporting guidance [10], with search reporting informed by PRISMA-S [11] and evidence synthesis guided by principles of structured narrative synthesis and Synthesis



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

Without Meta-analysis (SWiM) [12]. These frameworks support systematic evaluation while accommodating the methodological and technical heterogeneity of networking research.

A. Review Protocol and PRISMA Guidelines

A review protocol will establish the research questions, eligibility criteria, information sources, search procedures, screening rules, data extraction framework, quality-assessment criteria, and synthesis methods before formal study screening.

The protocol will address the seven research questions introduced in Section I.E, covering AI techniques, architectural integration, comparative performance, computational efficiency, operational feasibility, reproducibility, and research gaps.

PRISMA 2020 will guide reporting of study identification, screening, eligibility assessment, and final inclusion [10]. PRISMA-S will inform the documentation of database-specific search strategies, restrictions, execution dates, and supplementary searches [11].

Protocol amendments will be recorded with their rationale. Any public registration or repository deposit will be reported only if completed.

B. Search Strategy and Information Sources

The literature search will cover IEEE Xplore, Scopus, Web of Science Core Collection, ACM Digital Library, and ScienceDirect. These sources provide coverage of computer networking, telecommunications, artificial intelligence, and traffic engineering research.

Backward reference searching and forward citation tracking will supplement database searches. Google Scholar may provide additional discovery opportunities, while relevant IETF Request for Comments (RFC) documents will be consulted to establish architectural and protocol context. Standards will not automatically qualify as primary experimental evidence.

The proposed search period extends from 1 January 2015 to 18 October 2026. Earlier foundational studies may support theoretical discussions but will be excluded from the main time-bounded empirical synthesis unless a justified protocol amendment is documented.

All search activities will be recorded with their execution dates, database restrictions, and retrieved record counts.

C. Search Terms and Boolean Expressions

The search strategy combines three conceptual categories: network architecture, intelligent optimization, and traffic engineering.

The proposed core Boolean expression is:

("MPLS" OR "Multiprotocol Label Switching"

OR "SR-MPLS" OR "Segment Routing")

AND

("Machine Learning" OR "Artificial Intelligence"

OR "Reinforcement Learning"

OR "Deep Reinforcement Learning"

OR "Deep Q-Network" OR "Q-Learning"

OR "Actor-Critic")

AND

("Traffic Engineering" OR "Routing Optimization"

OR "Path Selection" OR "Path Computation"

OR "Congestion Control" OR "Load Balancing"

OR "Resource Allocation" OR "Traffic Prediction")



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

Supplementary searches will target specialized expressions, including “MPLS reinforcement learning,” “SR-MPLS deep reinforcement learning,” “Segment Routing traffic engineering machine learning,” and “MPLS traffic prediction neural network.”

Search expressions will be adapted to each database's syntax, indexing structure, and available filters. The final manuscript or supplementary materials will document executed queries, search fields, restrictions, and execution dates to support reproducibility [11].

D. Eligibility Criteria

Eligible studies must investigate an ML, RL, DRL, or hybrid intelligent method addressing a traffic engineering function and provide sufficient information about the network architecture, optimization objective, and evaluation procedure.

The primary synthesis will prioritize research involving MPLS LSP selection, SR Policy computation, congestion management, bandwidth allocation, routing optimization, or adaptive traffic control.

Table 1. Proposed study eligibility criteria

Criterion	Inclusion	Exclusion from primary synthesis
Publication period	2015–18 October 2026	Earlier studies, except contextual references
Publication type	Peer-reviewed journal and full conference papers	Preprints, technical reports, and standards as primary evidence
Network architecture	Explicit MPLS or relevant Segment Routing TE implementation or model	Generic IP/SDN routing without identifiable MPLS or relevant SR-TE mechanisms
AI approach	ML, RL, DRL, or hybrid intelligent optimization	Studies without an identifiable AI component
Application	Routing, traffic engineering, congestion management, resource allocation, or adaptive control	Unrelated AI applications
Traffic prediction	Explicitly connected to evaluated TE decisions	Prediction-only results without routing evaluation
Experimental evidence	Sufficient architectural, methodological, and outcome information	Insufficient information to assess the method or relevant results

SRv6-only studies will not be interpreted as direct evidence of SR-MPLS performance. Where a study evaluates multiple forwarding architectures, results will be extracted separately when possible. Contextual studies will be distinguished from primary empirical evidence.

Study Identification, Screening, and Selection

Retrieved records will be imported into a reference-management or screening system, preserving bibliographic information and source databases. Duplicate records will be identified using persistent identifiers and bibliographic matching.

Study selection will comprise two stages. First, titles and abstracts will be screened for relevance to AI-assisted traffic engineering and the specified networking architectures. Second, potentially eligible full texts will be assessed against the predefined inclusion and exclusion criteria.

Full-text assessment will examine the actual forwarding architecture, AI method, optimization function, evaluation design, and relevance of reported outcomes.

When multiple reviewers participate, the review will document the independent-screening arrangement and disagreement-resolution procedure actually used. Reasons for full-text exclusions will be recorded.

The completed PRISMA flow diagram will report the number of retrieved records, duplicates, screened records, excluded reports, full texts assessed, and included studies [10]. No screening counts or completed reviewer procedures are assumed at this stage.

E. Data Extraction Framework

A standardized extraction form will capture information needed to evaluate technical contributions, experimental methods, reported performance, and implementation limitations.



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

Table 2. Proposed data extraction framework

Dimension	Information to be extracted
Bibliographic characteristics	Authors, year, venue, study type
Network architecture	MPLS, SR-MPLS, SDN-assisted MPLS, or other relevant configuration
Intelligent approach	Learning paradigm, algorithm, features, state and action representations
Optimization objective	Congestion, utilization, delay, throughput, resource allocation, or other TE objectives
Experimental environment	Simulator, emulator, testbed, or hardware implementation
Network characteristics	Topology, number of nodes and links, traffic matrices, offered load
Training configuration	Training/testing procedures, reward design, hyperparameters, random seeds
Comparison methods	Shortest-path routing, CSPF, mathematical optimization, heuristics, or other AI methods
Performance outcomes	Link utilization, throughput, delay, jitter, packet loss, and resource consumption
Operational outcomes	Training cost, inference latency, policy-installation time, stability, constraint violations
Reproducibility	Availability of code, data, configurations, and implementation details
Study limitations	Generalization, experimental constraints, bias, and deployment feasibility

Quantitative results will be recorded with their original units, comparison baselines, and experimental conditions. Relative improvements will be calculated only when the necessary baseline values and metric definitions are available.

When publications report multiple algorithms, topologies, or experimental scenarios, the results will be extracted separately where necessary to preserve meaningful comparisons.

F. Methodological Quality Assessment

The methodological quality of eligible empirical studies will be assessed using a structured appraisal framework tailored to intelligent network optimization.

The framework will examine research-objective clarity, architectural relevance, algorithm transparency, baseline appropriateness, experimental realism, outcome reporting, constraint satisfaction, and reproducibility.

For ML-, RL-, and DRL-based studies, additional attention will be given to training/testing separation, data leakage, hyperparameter selection, convergence reporting, independent experimental repetitions, and evaluation under unseen traffic conditions.

Constraint satisfaction will be assessed independently of routing performance. A method reporting improved utilization without demonstrating feasible forwarding decisions will provide limited evidence of operational effectiveness.

Criterion-level assessments will be preferred over reliance on a single aggregate quality score. Where summary ratings are used, their definitions and interpretation will be specified in advance.

Quality appraisal will inform evidence interpretation and sensitivity analyses rather than automatically excluding studies with incomplete methodological reporting.

G. Thematic Classification and Evidence Synthesis

Included studies will be classified according to their primary learning paradigm: supervised learning, unsupervised learning, reinforcement learning, deep reinforcement learning, or hybrid intelligent optimization.

Secondary classifications will address forwarding architecture, control-plane integration, traffic engineering function, optimization objective, experimental platform, and deployment constraints.

Because studies are expected to differ in topology, traffic demands, performance metrics, and comparison baselines, structured narrative synthesis is planned as the principal synthesis method.



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

The analysis will distinguish directly evaluated MPLS or SR-MPLS systems from contextual networking research. Secondary surveys and protocol specifications will inform interpretation but will not be counted as independent experimental findings.

Where sufficiently comparable quantitative results are available, findings will be organized into evidence tables. Meta-analysis will be considered only when outcome definitions, baselines, experimental conditions, and study independence permit meaningful aggregation.

Otherwise, synthesis will follow relevant SWiM principles, including transparent grouping, consistent outcome reporting, and explicit acknowledgment of heterogeneity [12].

H. Comparative Evaluation of Classical and Intelligent Optimization

The review will critically compare AI-based approaches with established traffic engineering methods, including shortest-path routing, CSPF, LP/ILP/MILP, heuristic optimization, and alternative learning-based techniques.

Comparisons will distinguish improvements relative to basic routing from improvements demonstrated against competitive resource-aware optimization methods.

Evaluation dimensions will include solution quality, maximum link utilization, throughput, delay, packet loss, scalability, constraint satisfaction, computational overhead, and routing stability.

For learning-based approaches, offline training cost and online inference latency will be treated separately. Where available, controller overhead, path-validation time, and policy-installation delay will also be assessed.

Comparative conclusions will be qualified according to the equivalence of network topology, traffic conditions, optimization objectives, computational budgets, and forwarding constraints.

I. Heterogeneity, Bias, and Sensitivity Analysis

Heterogeneity will be assessed across network architectures, topology sizes, traffic distributions, optimization objectives, learning algorithms, experimental platforms, and comparison baselines.

Potential sources of bias include selective reporting, weak baselines, test-data leakage, insufficient experimental repetitions, and omission of computational or operational overhead.

Where sufficient evidence is available, sensitivity analyses will assess whether conclusions remain consistent when restricting the synthesis to studies with stronger methodological reporting, explicit MPLS implementations, competitive baselines, or unseen traffic evaluations.

Multiple results from the same study will not be treated as independent contributions without considering their shared experimental conditions.

If the evidence does not support meaningful sensitivity analysis, that limitation will be reported explicitly.

J. Reproducibility and Transparency

Reproducibility will be evaluated according to the availability and completeness of algorithm descriptions, source code, datasets, traffic matrices, network topologies, simulation configurations, random seeds, and training procedures.

The review's own methods will also be documented, including database search strategies, screening decisions, extraction fields, quality-assessment criteria, and synthesis procedures.

Where permitted, supplementary materials will provide a structured evidence dataset containing bibliographic information, study characteristics, appraisal results, and links to available replication resources.

Any restrictions on data, source code, or full-text access will be reported transparently.

K. Scope and Methodological Limitations

The review focuses primarily on peer-reviewed research published between January 2015 and October 2026, with earlier foundational publications retained for theoretical context.

This scope prioritizes contemporary intelligent traffic engineering but may exclude earlier relevant contributions. Restricting direct evidence to MPLS and relevant Segment Routing architectures also limits generalization from generic IP networking and SRv6-only studies.



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

Additional limitations may arise from database coverage, terminology differences, inaccessible publications, heterogeneous experimental methods, and incomplete reporting.

Accordingly, the review will emphasize the strength, relevance, and methodological quality of the available evidence rather than assuming universal superiority of a particular AI technique.

Overall, the proposed methodology integrates algorithmic classification, architecture-aware evidence assessment, baseline fairness, operational feasibility, and reproducibility. It is designed to support a rigorous evaluation of AI-assisted MPLS and Segment Routing traffic engineering while explicitly acknowledging the limitations of the underlying research evidence.

III. TECHNICAL FOUNDATIONS

A. MPLS and SR-MPLS Traffic Engineering

Multiprotocol Label Switching (MPLS) and Segment Routing over MPLS (SR-MPLS) provide fundamental architectures for traffic engineering (TE) in modern communication networks. Both support label-based packet forwarding and controlled path selection but differ in path establishment, signaling, and state management. These differences are particularly relevant to Machine Learning (ML), Reinforcement Learning (RL), and Deep Reinforcement Learning (DRL), as they influence the feasibility, scalability, and operational complexity of AI-assisted routing optimization.

Conventional MPLS Traffic Engineering. MPLS forwards packets along Label Switched Paths (LSPs) according to their Forwarding Equivalence Classes (FECs). Ingress routers assign labels, intermediate Label Switching Routers (LSRs) forward packets using label-based tables, and egress routers perform label disposition [2]. Conventional MPLS Traffic Engineering (MPLS-TE) extends this architecture by enabling path selection based on bandwidth availability, administrative policies, and network constraints rather than routing metrics alone.

Resource Reservation Protocol–Traffic Engineering (RSVP-TE) supports explicitly routed LSPs and resource reservation through signaling mechanisms [3]. Constraint-Based Shortest Path First (CSPF) is commonly used to identify feasible paths that satisfy predefined requirements. However, independently computed paths may cause resource contention and uneven link utilization. Mathematical programming and heuristic optimization can address these limitations through coordinated traffic allocation, although their effectiveness depends on computational complexity, network conditions, and optimization objectives.

Segment Routing over MPLS. Segment Routing introduces a source-routing approach in which packet-forwarding instructions are represented by ordered segments [4]. SR-MPLS implements these instructions using MPLS labels, maintaining compatibility with the MPLS forwarding data plane [5]. Segment Identifiers (SIDs), including Prefix-SIDs, Adjacency-SIDs, and Binding SIDs, represent forwarding instructions or associated paths and policies.

Unlike RSVP-TE, SR-MPLS can steer traffic through segment lists without maintaining conventional per-tunnel signaling state at every intermediate router. This simplifies certain control-plane operations and supports flexible path management. However, routers must maintain the necessary forwarding information, while segment-list depth, hardware capabilities, and policy constraints can limit path feasibility.

SR Policies and Programmable Traffic Engineering. Segment Routing Policies (SR Policies) enable traffic steering through candidate paths represented by segment lists [6]. These policies may be configured locally or computed by a controller, providing a practical mechanism for implementing dynamically optimized routing decisions. Nevertheless, SR-MPLS does not inherently guarantee optimal resource allocation, congestion avoidance, or bandwidth reservation. Such objectives require appropriate path-computation algorithms and complementary resource-management mechanisms.

Architectural Comparison and AI Implications. RSVP-TE establishes explicit LSPs through signaling and can support bandwidth reservations, whereas SR-MPLS emphasizes segment-based forwarding and policy-driven



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

path control. Although SR-MPLS can reduce conventional per-tunnel state in intermediate routers, it introduces constraints related to segment-list computation, label-stack depth, and forwarding-platform capabilities. Consequently, improvements in network performance must be attributed carefully to the optimization algorithm, forwarding architecture, or their interaction.

AI-assisted traffic engineering can enhance both architectures through traffic prediction, congestion detection, adaptive path selection, and resource optimization. In conventional MPLS-TE, ML, RL, and DRL algorithms may optimize LSP selection and modification, while in SR-MPLS, they may select SR Policy candidate paths or generate feasible segment lists. However, frequent routing adjustments can introduce computational overhead, packet reordering, and network instability.

Therefore, evaluating AI-assisted traffic engineering requires consideration of both network performance and operational feasibility. Key evaluation criteria include throughput, delay, packet loss, link utilization, computational overhead, routing stability, constraint satisfaction, and policy-deployment requirements.

Overall, MPLS-TE and SR-MPLS offer complementary foundations for intelligent traffic engineering. MPLS-TE provides established signaling and resource-management mechanisms, while SR-MPLS enables flexible, policy-driven traffic steering. Understanding their architectural differences is essential for critically evaluating AI-based routing algorithms and their practical deployment in dynamic networks.

B. *Classical Routing Optimization Baselines*

Classical routing optimization methods provide essential benchmarks for evaluating intelligent Traffic Engineering (TE) in Multiprotocol Label Switching (MPLS) and Segment Routing over MPLS (SR-MPLS) networks. These methods include shortest-path routing, Constraint-Based Shortest Path First (CSPF), mathematical programming, multicommodity-flow optimization, and heuristic or metaheuristic algorithms. They support path selection, bandwidth allocation, load balancing, and congestion management while providing established baselines for assessing Machine Learning (ML), Reinforcement Learning (RL), and Deep Reinforcement Learning (DRL) approaches.

Shortest-Path and Constraint-Based Routing. Shortest-path algorithms, particularly Dijkstra's Shortest Path First (SPF), determine forwarding paths by minimizing accumulated link metrics. Equal-Cost Multipath (ECMP) enables traffic distribution across multiple equal-cost paths, improving resource utilization in suitable conditions. Although computationally efficient, these mechanisms do not inherently optimize network-wide traffic distribution under changing demands. Interior Gateway Protocol (IGP) link-weight optimization offers a stronger baseline by adjusting routing metrics to improve load distribution and reduce congestion [13].

CSPF extends shortest-path routing by incorporating network-state information and constraints, such as available bandwidth, administrative policies, and link attributes. In conventional MPLS-TE, computed paths can be established through Resource Reservation Protocol-Traffic Engineering (RSVP-TE), which provides signaling and resource-reservation mechanisms rather than independent routing optimization [3, 14]. However, sequential path computation may produce suboptimal network-wide allocations when multiple traffic demands compete for shared resources.

Mathematical Programming and Multicommodity-Flow Optimization. Linear Programming (LP) and multicommodity-flow models formulate traffic engineering as a constrained resource-allocation problem. Integer Linear Programming (ILP) and Mixed-Integer Linear Programming (MILP) additionally support discrete routing decisions, unsplittable flows, and operational restrictions. Typical objectives include minimizing maximum link utilization, balancing traffic loads, and satisfying bandwidth and path constraints.

A common objective is minimizing maximum link utilization:

$$\min_x U_{max} = \min_x \max_{e \in E} \left(\frac{L_e(X)}{C_e} \right)$$



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

where E denotes the set of network links, $L_e(X)$ represents the traffic load on link e under routing decision

X , and C_e denotes its capacity. The optimization is subject to applicable flow-conservation, capacity, and

routing-feasibility constraints. These formulations offer interpretable solutions and, where applicable, optimality guarantees or solution-quality bounds. However, computational complexity can increase substantially with network size, traffic demands, and discrete routing constraints. Furthermore, solutions based on splittable flows may not be directly applicable to architectures requiring unsplittable traffic or restricted forwarding paths.

Heuristic and Metaheuristic Optimization. Greedy algorithms, (k)-shortest-path methods, local search, genetic algorithms, simulated annealing, and ant colony optimization provide alternatives for computationally demanding routing problems. These methods seek feasible or near-optimal solutions while balancing computational efficiency and optimization quality. Although generally more flexible than exact optimization, their performance depends on algorithm configuration, network topology, and traffic conditions. Consequently, well-configured heuristics represent important competitive baselines for evaluating AI-assisted routing.

Classical Optimization in SR-MPLS. Classical optimization techniques can also compute paths for SR-MPLS networks, where forwarding instructions are represented through Segment Identifier (SID) lists and implemented using SR Policies [6]. Compared with RSVP-TE, SR-MPLS can reduce conventional per-tunnel signaling state at intermediate routers. However, path computation must account for SID availability, segment-list depth, and forwarding-platform limitations. SR-MPLS therefore provides a programmable traffic-steering architecture rather than an independent optimization algorithm.

Implications for AI-Based Traffic Engineering. Classical routing methods face limitations related to computational scalability, traffic uncertainty, reoptimization frequency, and operational overhead. Mathematical programming may incur substantial computational costs, while heuristics and constraint-based methods may not achieve globally optimal traffic allocation. Nevertheless, these limitations do not automatically establish the superiority of learning-based methods, which introduce additional costs associated with training, inference, generalization, and decision validation.

Fair comparisons should therefore consider appropriately configured classical baselines under equivalent network topologies, traffic matrices, routing constraints, and optimization objectives. Evaluation metrics should include link utilization, throughput, delay, packet loss, computational overhead, routing stability, and constraint satisfaction. For learning-based methods, offline training costs should be distinguished from online inference latency, while classical solvers should be evaluated using documented computational budgets and configurations.

Overall, classical optimization methods provide essential reference points for determining whether ML, RL, and DRL approaches achieve meaningful improvements in routing efficiency, adaptability, scalability, and operational feasibility within MPLS and SR-MPLS networks.

C. SDN, PCE, and Programmable Path Control

Software-Defined Networking (SDN), Path Computation Elements (PCEs), and programmable path-control mechanisms provide essential architectural support for integrating Artificial Intelligence (AI) into Multiprotocol Label Switching (MPLS) and Segment Routing over MPLS (SR-MPLS) Traffic Engineering (TE). These technologies facilitate network-state monitoring, coordinated path computation, automated policy deployment, and adaptive routing. Their integration is particularly relevant to Machine Learning (ML), Reinforcement Learning (RL), and Deep Reinforcement Learning (DRL), which require mechanisms for observing network conditions and implementing routing decisions.

Software-Defined Networking. SDN separates network-control functions from packet forwarding, enabling programmable network management through logically centralized controllers. In MPLS and SR-MPLS networks, controllers can coordinate topology discovery, traffic monitoring, path selection, congestion



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

management, and forwarding-policy updates. This architecture facilitates AI-driven traffic optimization, although its effectiveness depends on network-state accuracy, controller scalability, communication latency, and forwarding-device capabilities.

Path Computation Elements. The PCE architecture, defined in RFC 4655, supports constraint-based path computation using network topology, resource availability, and operational requirements. The Path Computation Element Communication Protocol (PCEP), specified in RFC 5440, enables communication between Path Computation Clients (PCCs) and PCEs. Stateful PCE operation extends conventional path computation by maintaining information about established Label Switched Paths (LSPs), supporting coordinated optimization and dynamic path management. RFC 8231 defines stateful PCEP extensions, while RFC 8281 enables PCE-initiated LSP establishment. However, these capabilities depend on reliable state synchronization, authorization, and communication.

Programmable Path Control in SR-MPLS. SR-MPLS represents forwarding instructions through ordered Segment Identifier (SID) lists implemented using SR Policies. RFC 8664 extends PCEP to support Segment Routing path computation and control, enabling controllers and PCEs to coordinate segment-based paths. This programmability allows AI algorithms to generate routing decisions that can be translated into forwarding policies. Nevertheless, computed paths must satisfy operational constraints, including SID availability, Maximum SID Depth (MSD), and forwarding-device capabilities.

Network-state acquisition and configuration management further support programmable TE. Border Gateway Protocol–Link State (BGP-LS), specified in RFC 9552, distributes topology and TE information to external controllers. NETCONF and RESTCONF provide standardized interfaces for retrieving and modifying supported device configurations. Combined with network telemetry, these mechanisms support traffic monitoring and automated policy deployment. However, accurate congestion assessment requires traffic measurements beyond topology information alone.

Closed-Loop AI-Assisted Traffic Engineering. Integrating SDN, PCE, and programmable forwarding enables closed-loop traffic engineering, in which network measurements inform intelligent routing decisions, controllers implement feasible paths, and subsequent observations provide feedback for optimization. ML, RL, and DRL can support traffic prediction, adaptive path selection, and congestion-aware policy management. However, practical deployment faces challenges involving telemetry delays, computational overhead, policy-installation latency, routing instability, and inconsistencies between intended and actual forwarding behavior. Consequently, intelligent controllers require feasibility validation, stability safeguards, and recovery mechanisms.

Overall, SDN and PCE architectures provide the computational and control infrastructure necessary for AI-assisted MPLS and SR-MPLS traffic engineering. Their primary contribution lies in enabling programmable path computation, validation, deployment, and monitoring rather than independently guaranteeing optimal network performance. These architectural foundations support the formulation of intelligent traffic engineering problems in terms of network states, routing actions, optimization objectives, and operational constraints, as discussed in Section III.E.

D. Formulating Intelligent Traffic Engineering Problems

Intelligent Traffic Engineering (TE) in Multiprotocol Label Switching (MPLS) and Segment Routing over MPLS (SR-MPLS) networks can be formulated as a constrained optimization and sequential decision-making problem. Its objective is to determine feasible routing configurations that improve network performance under dynamic traffic demands and operational constraints. Machine Learning (ML), Reinforcement Learning (RL), and Deep Reinforcement Learning (DRL) support this objective through traffic prediction, path selection, resource allocation, and adaptive routing control. A consistent mathematical formulation is essential for evaluating these approaches against classical optimization baselines.



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

Network Model and Optimization Objectives. A communication network can be represented as a directed graph $G = (V, E)$, where V denotes routers, E represents directed links, and C_e is the capacity of link e . Let

K denote the set of traffic demands, $d_k(t)$ the demand associated with commodity k at time t , and $x_{kp}(t)$

the proportion of demand k assigned to candidate path p . The resulting link load is expressed as:

$$L_e(t) = \sum_{k \in K} d_k(t) \sum_{p \in P_k} x_{kp}(t) a_{ep}$$

where a_{ep} if path p traverses link e , and zero otherwise. The routing variables satisfy $\sum_{p \in P_k} x_{kp}(t) = 1$

and $x_{kp}(t) \geq 0$, subject to supported traffic-splitting mechanisms. A common optimization objective is

minimizing maximum link utilization:

$$\min_{x(t)} U_{\max}(t) = \min_{x(t)} \max_{e \in E} \frac{L_e(t)}{C_e}$$

Although this objective promotes balanced resource utilization, intelligent TE may also consider end-to-end delay, throughput, packet loss, routing stability, and reconfiguration costs. Multiobjective optimization must therefore distinguish performance objectives from mandatory forwarding constraints.

ML-Based Traffic Prediction. Supervised ML approaches typically estimate future traffic demands, link utilization, or congestion using historical network measurements and topology information. These predictions inform subsequent routing optimization and resource allocation. However, prediction accuracy alone does not guarantee improved network performance; effectiveness depends on whether predicted information leads to better routing decisions under realistic traffic conditions.

RL-Based Sequential Routing Optimization. RL models adaptive routing as a Markov Decision Process (MDP), represented by (S, A, P, R, γ) , where S denotes network states, A represents routing actions, P

describes state transitions, R defines rewards, and γ is the discount factor. States may include link utilization,

traffic demands, queue occupancy, and congestion indicators, while actions may involve selecting paths, adjusting traffic-splitting ratios, or modifying forwarding policies.

The objective is to learn a policy $\pi(\alpha|s)$ that maximizes expected cumulative reward:

$$\pi^* = \arg \max_{\pi} E_{\pi} \left[\sum_{t=0}^T \gamma^t r_t \right]$$



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

where r_t is the reward at time t , and T denotes the decision horizon. Reward functions may combine

congestion reduction, throughput improvement, delay minimization, and penalties for excessive routing changes. However, inappropriate reward design may produce undesirable behavior, while incomplete network observations can require partially observable or history-based formulations. DRL-Based Routing Optimization. DRL extends RL through neural-network function approximation, enabling adaptive decision-making in high-dimensional network environments. Techniques such as Deep Q-Networks (DQN), policy-gradient methods, and actor-critic algorithms can learn routing policies from complex network observations. Nevertheless, large routing-action spaces present computational and scalability challenges. Candidate-path restrictions, hierarchical decision-making, and graph-based representations can improve tractability but may limit the range of routing solutions considered.

Architectural Constraints and Operational Feasibility. Intelligent routing decisions must satisfy the forwarding requirements of the underlying architecture. In MPLS-TE, relevant constraints include bandwidth availability, Label Switched Path (LSP) establishment, and administrative policies. SR-MPLS additionally requires valid Segment Identifier (SID) lists, compliance with Maximum SID Depth (MSD), and feasible SR Policy configurations. RFC 9256 defines the architectural framework for SR Policy candidate-path selection and validity. Consequently, AI-generated routing decisions should undergo constraint validation before deployment.

Practical evaluation must also distinguish offline training costs from online inference and policy-deployment latency. Frequent routing adjustments may introduce computational overhead, packet reordering, and forwarding instability. Therefore, intelligent TE should be assessed using optimization quality, computational efficiency, scalability, generalization, constraint satisfaction, and routing stability.

Overall, intelligent MPLS and SR-MPLS traffic engineering requires consistent definitions of network states, routing actions, optimization objectives, reward functions, and forwarding constraints. While ML supports predictive optimization, RL and DRL enable sequential routing decisions. Their effectiveness ultimately depends on the alignment between learning algorithms, forwarding capabilities, and operational requirements. This formulation provides the foundation for reviewing ML-, RL-, DRL-, and hybrid-based traffic engineering approaches in Section IV.

IV. AI-BASED MPLS AND SEGMENT ROUTING OPTIMIZATION

Artificial Intelligence (AI) introduces data-driven approaches to Traffic Engineering (TE) in Multiprotocol Label Switching (MPLS) and Segment Routing over MPLS (SR-MPLS) networks. Unlike conventional optimization methods based on predefined routing metrics, mathematical models, and heuristics, AI techniques learn from historical traffic measurements, network states, and optimization outcomes to support traffic prediction and adaptive routing. The principal approaches include supervised learning, unsupervised learning, Reinforcement Learning (RL), Deep Reinforcement Learning (DRL), and hybrid optimization. These methods differ in their learning mechanisms, computational requirements, and decision-making capabilities. Their effectiveness must therefore be evaluated in terms of network performance, scalability, forwarding feasibility, computational overhead, and adaptability to changing traffic conditions.

E. Supervised Learning for Traffic Prediction and Routing

Supervised Machine Learning (ML) supports traffic prediction, network-performance estimation, and routing optimization by learning relationships between network measurements and known target outputs. Common techniques include regression models, Support Vector Machines (SVMs), Random Forests (RFs), Artificial Neural Networks (ANNs), Long Short-Term Memory (LSTM) networks, and Graph Neural Networks (GNNs). These methods can predict traffic demands, congestion, and Quality of Service (QoS) indicators or approximate



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

routing decisions generated by conventional optimization algorithms. However, supervised learning does not independently guarantee optimal routing performance.

Traffic Prediction. Supervised models use historical traffic matrices, link-utilization measurements, and temporal patterns to forecast future network demands. Regression and tree-based models capture relationships among network variables, while LSTM networks model temporal dependencies. Traffic prediction can be expressed as:

$$\hat{\mathbf{D}}(t+h) = f\vartheta(\mathbf{D}(t), \mathbf{D}(t-1), \dots, \mathbf{D}(t-w+1))$$

where $\hat{\mathbf{D}}(t+h)$ represents the predicted traffic-demand matrix at forecasting horizon h , $f\vartheta$ denotes the prediction model with parameters ϑ , and w is the historical observation window. These forecasts can support

proactive path selection, bandwidth allocation, and congestion management, although prediction errors may reduce routing efficiency under unexpected traffic changes.

Le et al. [26] proposed a multi-time-step Segment Routing approach integrating traffic prediction and compressive sensing to optimize routing across multiple intervals. Their experiments reported more than 80% of the reference optimal performance in reducing maximum link utilization while limiting routing changes and monitoring overhead. These findings demonstrate the potential of prediction-assisted routing under the evaluated experimental conditions.

Network-Performance Prediction. Supervised learning can also estimate delay, jitter, packet loss, and congestion for different network configurations. GNNs are particularly suitable for representing network topology and routing relationships. Rusek et al. [27] introduced RouteNet, a GNN-based model that predicts network-performance indicators from topology, routing configurations, and traffic information. The model demonstrated predictive capabilities across different network conditions, including unseen topologies. However, its evaluation in an SDN environment does not directly establish operational feasibility in MPLS or SR-MPLS networks.

Learning-Based Routing Optimization. Supervised models can approximate routing decisions generated by Linear Programming (LP), Mixed-Integer Linear Programming (MILP), or heuristic algorithms. Salman and Wang [28] investigated neural-network models trained on optimization-derived traffic-splitting decisions, demonstrating near-optimal traffic allocation with rapid adaptation in simulated SDN environments. Similarly, Zuo et al. [29] developed a sequence-to-sequence path-planning model incorporating attention mechanisms and beam search to generate connected candidate paths. Nevertheless, applying such approaches to MPLS and SR-MPLS requires additional validation of bandwidth constraints, Label Switched Path (LSP) feasibility, and Segment Identifier (SID) requirements.

Integration and Evaluation Challenges. In MPLS and SR-MPLS networks, supervised models can operate alongside SDN controllers or Path Computation Elements (PCEs) to support traffic forecasting, feasible path computation, and SR Policy selection. However, model predictions must be translated into valid forwarding decisions through appropriate control and signaling mechanisms.

Major challenges include limited training-data representativeness, changing traffic patterns, topology modifications, prediction errors, and model generalization. Evaluation must distinguish forecasting accuracy, measured using metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), from routing effectiveness, assessed through link utilization, throughput, delay, packet loss, routing stability, and constraint satisfaction. Training, inference, and retraining costs should also be considered.



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

Overall, supervised learning can improve traffic awareness, network-performance estimation, and routing-decision efficiency. However, its practical value depends on translating accurate predictions into feasible, stable, and efficient MPLS or SR-MPLS routing configurations. Consequently, supervised models should be evaluated as components of constraint-aware traffic engineering frameworks rather than standalone guarantees of network optimization.

F. *Unsupervised Learning for Traffic Characterization*

Unsupervised Machine Learning (ML) identifies hidden patterns, similarities, and anomalies in network traffic without requiring predefined target labels. In Multiprotocol Label Switching (MPLS) and Segment Routing over MPLS (SR-MPLS) networks, these techniques support traffic characterization, congestion identification, anomaly detection, and network-state classification. Common approaches include K-means clustering, Density-Based Spatial Clustering of Applications with Noise (DBSCAN), Principal Component Analysis (PCA), Self-Organizing Maps (SOMs), and autoencoders. However, their contribution to Traffic Engineering (TE) depends on integrating the extracted information with routing optimization and resource-allocation mechanisms.

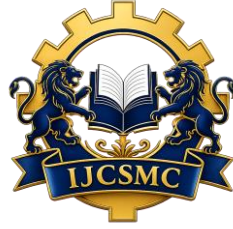
Traffic Clustering and Characterization. Clustering algorithms group network flows or operating states according to similarities in bandwidth consumption, flow duration, packet characteristics, link utilization, and temporal traffic patterns. K-means partitions observations into predefined clusters, while DBSCAN identifies density-based groups and potential outliers. These approaches can reveal recurring congestion conditions and representative traffic-demand patterns, although their effectiveness depends on feature selection, parameter configuration, and traffic-data characteristics.

Guo et al. [30] proposed a Weight Adjustment algorithm for Segment Routing (WASAR), integrating Ultra-scalable Spectral Clustering (U-SPEC) with Deep Reinforcement Learning (DRL) and Linear Programming (LP). Historical Traffic Matrices (TMs) were clustered into representative demand patterns to support offline DRL optimization, while LP determined online traffic-splitting decisions. Experiments using Abilene, CERNET, and GEANT topologies demonstrated improvements in Maximum Link Utilization (MLU) under the evaluated conditions. However, performance depended on clustering parameters, network topology, and Segment Routing deployment ratios, indicating that the benefits resulted from the combined optimization framework rather than clustering alone.

Traffic Classification and Feature Extraction. Unsupervised learning can identify statistical similarities among application flows and network-service characteristics. Salau and Beyene [31] investigated K-means clustering alongside supervised classifiers in Software-Defined Networking (SDN). Their results showed substantially lower labeling accuracy for K-means (approximately 30%) than for the best-performing supervised Decision Tree classifier (99.81%). These findings demonstrate that unsupervised clusters do not necessarily correspond to predefined application or Quality of Service (QoS) categories.

Dimensionality-reduction methods, including PCA and autoencoders, can transform high-dimensional network measurements into compact representations. PCA captures major directions of data variation, while autoencoders learn nonlinear features through reconstruction. Such representations may reduce the complexity of traffic analysis and subsequent optimization, although important congestion indicators or rare events may be lost during feature reduction.

Anomaly Detection and Network-State Monitoring. Unsupervised techniques also identify abnormal traffic patterns and deviations from normal operating conditions. Carvalho et al. [32] investigated unsupervised traffic profiling and adaptive anomaly detection, while Mazel et al. [33] combined subspace clustering and correlation analysis to characterize anomalies in backbone networks. Similarly, Hwang et al. [34] proposed D-PACK, integrating Convolutional Neural Networks (CNNs) and autoencoder-based learning for early traffic anomaly detection. These approaches can improve network-state awareness, but anomaly detection does not



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

independently optimize routing. Furthermore, abnormal traffic may result from congestion, failures, security incidents, or measurement errors, requiring validation before routing adjustments.

Integration and Evaluation Challenges. In MPLS and SR-MPLS networks, unsupervised learning can support intelligent TE by identifying representative traffic patterns and network operating states. These outputs may inform SDN controllers, Path Computation Elements (PCEs), or optimization algorithms responsible for selecting feasible Label Switched Paths (LSPs) or SR Policies. However, routing decisions must satisfy bandwidth requirements, Segment Identifier (SID) constraints, forwarding capabilities, and operational stability.

Major challenges include sensitivity to feature scaling, clustering parameters, changing traffic distributions, and model generalization. Evaluation should therefore consider clustering stability, anomaly-detection accuracy where reference labels are available, computational overhead, and downstream TE performance, including MLU, throughput, delay, packet loss, and routing stability. Comparisons with equivalent optimization methods operating without unsupervised preprocessing are necessary to establish its added value.

Overall, unsupervised learning provides complementary capabilities for traffic characterization, anomaly identification, and network-state representation. Its primary contribution is improving the information available to routing optimization rather than directly generating optimal paths. Consequently, its effectiveness in MPLS and SR-MPLS depends on integration with constraint-aware optimization algorithms and validation under realistic network conditions.

G. Reinforcement Learning for Adaptive Routing

Reinforcement Learning (RL) provides an adaptive decision-making framework for Traffic Engineering (TE) in Multiprotocol Label Switching (MPLS) and Segment Routing over MPLS (SR-MPLS) networks. Unlike supervised learning, RL learns routing policies through interactions with a network environment and feedback from selected actions. Common approaches include Q-learning, State–Action–Reward–State–Action (SARSA), and temporal-difference learning. These methods support adaptive path selection, load balancing, congestion management, and Quality of Service (QoS) optimization under changing network conditions.

RL-Based Routing Formulation. Adaptive routing is commonly modeled as a Markov Decision Process (MDP), in which an agent observes network states, selects routing actions, receives rewards, and updates its policy. States may include link utilization, available bandwidth, queue occupancy, and congestion indicators, while actions may involve selecting Label Switched Paths (LSPs), adjusting traffic-splitting ratios, or modifying Segment Routing policies. Rewards evaluate routing decisions according to throughput, delay, congestion, and stability objectives.

Q-learning estimates the expected cumulative reward associated with each state–action pair using the update rule:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t) \right]$$

where $Q(s_t, a_t)$ represents the estimated action value, α denotes the learning rate, γ is the discount factor,

and r_t is the immediate reward. Q-learning follows an off-policy approach, whereas SARSA updates action

values according to the next action selected by the current policy. Their effectiveness depends on state representation, exploration strategies, reward design, and network dynamics.

RL-Based Routing and Segment Routing Optimization. Boyan and Littman [38] introduced Q-routing, demonstrating the potential of learning-based forwarding decisions under dynamic traffic conditions. Although



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

foundational, this study predates the review's 2015–2026 inclusion period and does not directly address MPLS or SR-MPLS optimization.

Wumian et al. [35] proposed RLSR, a QoS-aware RL framework integrating SARSA with Segment Routing-enabled Software-Defined Networking (SDN). The approach employed loop-free path exploration, aggregate action selection, and reusable network-state knowledge to reduce learning overhead. Experimental results showed a reduction in maximum link utilization from 71.67% to 50% compared with the evaluated baseline, while knowledge reuse reduced training episodes by up to 27%. However, these findings depend on specific experimental conditions and do not independently establish superiority over competitive mathematical optimization methods.

Wang et al. [36] introduced Smart Node Selection (SNS), combining RL-based intermediate-node selection with Linear Programming (LP) for traffic allocation. Experiments demonstrated reduced computation time and control overhead relative to the evaluated LP-based approaches while maintaining favorable network performance. This study highlights the potential of combining RL with classical optimization rather than replacing it entirely.

Similarly, Ke et al. [37] proposed a Q-learning Widest-Path Routing Algorithm (Q-WPRA) that uses bandwidth-related rewards for adaptive path selection. Mininet-based SDN experiments demonstrated improvements over the evaluated shortest-path and widest-path baselines. However, the study did not directly validate MPLS signaling or SR-MPLS forwarding constraints, limiting its applicability to architecture-specific traffic engineering.

Challenges and Operational Considerations. RL-based traffic engineering faces challenges involving reward design, exploration–exploitation trade-offs, scalability, and routing stability. Conflicting objectives, such as minimizing delay while balancing link utilization, require carefully designed reward functions. Moreover, uncontrolled exploration may introduce congestion, packet loss, or service disruption. Offline training can reduce these risks, although learned policies may not generalize to unseen traffic conditions or network failures.

In MPLS and SR-MPLS environments, RL-generated actions must satisfy bandwidth constraints, LSP feasibility, Segment Identifier (SID) requirements, and forwarding-policy compatibility. Such constraints require explicit validation rather than reliance on reward penalties alone. Practical evaluation should also consider training convergence, computational overhead, inference latency, policy-installation costs, routing changes, and recovery performance.

Overall, RL enables adaptive routing through experience-based policy optimization and network-performance feedback. Existing studies demonstrate promising results in QoS-aware path selection, Segment Routing optimization, and hybrid RL–LP approaches. However, scalability limitations, operational constraints, and differences in experimental architectures restrict the generalizability of reported improvements. These challenges motivate Deep Reinforcement Learning (DRL), which employs neural-network function approximation to address more complex routing problems, as discussed in Section IV.E.

H. Deep Reinforcement Learning for Traffic Engineering

Deep Reinforcement Learning (DRL) extends conventional Reinforcement Learning (RL) by integrating neural networks into sequential decision-making. In Multiprotocol Label Switching (MPLS) and Segment Routing over MPLS (SR-MPLS) networks, DRL supports adaptive path selection, congestion management, traffic allocation, and Quality of Service (QoS) optimization. Common approaches include Deep Q-Networks (DQNs), Deep Deterministic Policy Gradient (DDPG), Proximal Policy Optimization (PPO), and actor–critic methods. Unlike tabular RL, DRL can approximate routing policies using high-dimensional network observations, although its effectiveness depends on state representation, action-space design, training stability, and forwarding constraints [7].

DRL-Based Routing Formulation. DRL agents observe network conditions, including link utilization, bandwidth availability, traffic demands, and congestion levels, and select actions such as path adjustments,



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

traffic-splitting modifications, or routing-policy updates. DQN-based approaches estimate action values for discrete routing decisions, whereas actor-critic and policy-gradient methods support more complex control strategies. A general policy-gradient formulation is expressed as:

$$\nabla_{\theta} J(\theta) = E_{\pi_{\theta}} [\nabla_{\theta} \log \pi_{\theta}(a_t | s_t) A^{\pi_{\theta}}(a_t | s_t)]$$

where θ denotes policy parameters, $\pi_{\theta}(a_t | s_t)$ represents the probability of selecting action a_t in state s_t ,

and $A^{\pi_{\theta}}(a_t | s_t)$ is the advantage function. Although this formulation supports policy improvement through

performance feedback, routing feasibility and operational constraints require explicit enforcement.

DRL-Based Segment Routing Optimization. Tian et al. [8] proposed Adpt-SRTE, integrating PPO with an action-branching neural network to address large routing-action spaces. The framework combines offline training on historical Traffic Matrices (TMs) with online decisions based on link utilization. Experiments using real-world topologies reported Maximum Link Utilization (MLU) reductions of up to 33% relative to the evaluated baselines, with online decisions completed within tens of milliseconds. However, these improvements depend on training conditions, candidate-path representations, and experimental assumptions, while direct SR-MPLS compatibility requires verification.

Chen et al. [39] investigated DRL-based traffic splitting in hybrid Internet Protocol/Segment Routing (IP/SR) networks. Their approach dynamically adjusted traffic distribution across multiple paths, achieving throughput improvements of up to 9% in the evaluated simulations. These findings demonstrate the potential of adaptive traffic allocation, although performance depends on network architecture, forwarding capabilities, and the selected comparison methods.

Tian et al. [40] introduced Weight Adjustment-Segment Routing Traffic Engineering (WA-SRTE), applying DRL to jointly optimize routing weights, node deployment, and traffic paths in partially deployed Segment Routing over IPv6 (SRv6) networks. Their experiments indicated that enabling SRv6 at approximately 20–40% of nodes could achieve performance comparable to full deployment under the evaluated conditions. However, these results represent SRv6-specific evidence and should not be directly generalized to SR-MPLS.

Xu et al. [41] developed DRL-TE, an actor-critic framework incorporating traffic-engineering-aware exploration and prioritized experience replay. Packet-level simulations demonstrated improvements in end-to-end delay and network utility compared with the evaluated baselines. Nevertheless, the study provides general DRL traffic engineering evidence rather than direct validation of MPLS or SR-MPLS forwarding requirements.

Hybrid DRL and Mathematical Optimization. DRL can also be combined with classical optimization to improve routing efficiency and constraint handling. Guo et al. [30] integrated unsupervised traffic clustering, DRL-based routing optimization, and Linear Programming (LP) for online traffic splitting in hybrid Segment Routing networks. Such approaches illustrate how learning-based methods can complement mathematical optimization, although their effectiveness must account for the computational overhead of each processing stage.

Challenges and Operational Considerations. DRL-based traffic engineering faces challenges involving large state and action spaces, computationally intensive training, reward design, generalization, and routing stability. Although neural networks improve representation capabilities, they do not eliminate the combinatorial complexity of routing optimization. Candidate-path restrictions, hierarchical decisions, and graph-based representations may improve scalability but can limit routing flexibility.

In MPLS and SR-MPLS environments, DRL-generated decisions must satisfy bandwidth availability, Label Switched Path (LSP) feasibility, Segment Identifier (SID) validity, Maximum SID Depth (MSD), and applicable SR Policy constraints. Feasibility filtering and integration with Path Computation Elements (PCEs) can help



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

prevent invalid routing actions. Evaluation should distinguish offline training costs from online inference, path validation, and policy-installation latency. Comparative experiments should also consider network scalability, unseen traffic conditions, link failures, routing stability, and competitive classical optimization baselines.

Overall, DRL offers promising capabilities for adaptive routing, congestion management, and traffic allocation. However, reported improvements remain dependent on experimental conditions, forwarding architecture, and comparison baselines. Its practical effectiveness in MPLS and SR-MPLS networks therefore requires architecture-aware validation, operational safeguards, and integration with conventional traffic engineering mechanisms.

I. Hybrid AI and Mathematical Optimization

Hybrid Artificial Intelligence (AI) and mathematical optimization combine the adaptability of learning-based methods with the constraint-handling capabilities of conventional Traffic Engineering (TE) algorithms. In Multiprotocol Label Switching (MPLS) and Segment Routing over MPLS (SR-MPLS) networks, these approaches integrate Machine Learning (ML), Reinforcement Learning (RL), or Deep Reinforcement Learning (DRL) with Linear Programming (LP), Mixed-Integer Linear Programming (MILP), and heuristic optimization. Their objective is to improve routing efficiency, resource allocation, and computational scalability while maintaining forwarding feasibility.

Hybrid Optimization Frameworks. Learning-based methods can support mathematical optimization through traffic prediction, candidate-path selection, and initial routing decisions. Mathematical optimization can subsequently refine these decisions and enforce routing constraints. A general hybrid formulation is:

$$X^*(t) = \arg_{x \in X_{feasible}(\hat{D}(t))} F(x, \hat{D}(t), G)$$

where $\hat{D}(t)$ represents predicted traffic demands, G denotes network topology, F is the optimization

objective, and $X_{feasible}$ defines routing configurations satisfying the specified constraints. However, feasibility

under predicted demands does not necessarily guarantee feasibility under actual traffic conditions, requiring additional validation or capacity margins.

Learning-Assisted Routing Optimization. Wang et al. [36] proposed Smart Node Selection (SNS), combining RL-based intermediate-node selection with LP for traffic allocation. Their experiments demonstrated reduced computation time and control overhead while maintaining competitive TE performance. However, restricting candidate nodes may exclude globally optimal routing solutions.

Wang et al. [42] introduced RS4SR, a recommendation-based framework that exploits historical LP solutions and flow clustering to accelerate Segment Routing optimization. Experiments indicated that near-optimal routing configurations could be obtained within a two-second computation limit on large networks. Nevertheless, performance depends on the relevance of historical solutions to changing traffic conditions.

Guo et al. [30] developed the Weight Adjustment algorithm for Segment Routing (WASAR), integrating unsupervised traffic clustering, DRL, and online LP optimization. The framework used representative Traffic Matrices (TMs) for DRL-based optimization of routing weights and Segment Routing deployment, while LP determined traffic-splitting ratios. Experiments involving Abilene, CERNET, and GEANT topologies demonstrated improvements in Maximum Link Utilization (MLU). However, effectiveness depended on traffic characteristics and hybrid routing assumptions.

Optimization Refinement and Safe Learning. Lu et al. [43] proposed Graph-based Reinforcement Learning for Traffic Engineering (GRL-TE), combining Graph Neural Networks (GNNs), Advantage Actor-Critic (A2C), and the Alternating Direction Method of Multipliers (ADMM). The learning component generated traffic



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

allocations, while ADMM refined solutions to improve constraint satisfaction. Experiments across six Wide Area Network (WAN) topologies reported an average demand-satisfaction rate of 89.36% and computational improvements over the evaluated LP-based methods. However, these findings concern general Software-Defined Networking (SDN) rather than verified MPLS or SR-MPLS implementations.

Wang et al. [44] investigated safe training for DRL-based traffic engineering to reduce harmful exploration and improve learning convergence. Simulation results demonstrated faster convergence while maintaining the safety conditions considered in the study. Although not specifically validated for MPLS or SR-MPLS, the approach highlights the importance of incorporating operational constraints into intelligent routing optimization.

Challenges and Operational Considerations. Hybrid optimization introduces trade-offs involving computational complexity, prediction accuracy, solution quality, and operational feasibility. Candidate-path restrictions may accelerate optimization but reduce the available solution space, while combining multiple algorithms introduces additional processing and configuration overhead.

In MPLS-TE, routing solutions must satisfy Label Switched Path (LSP) feasibility, bandwidth availability, and signaling requirements. SR-MPLS additionally requires valid Segment Identifier (SID) lists, Maximum SID Depth (MSD) compliance, and feasible SR Policies. Consequently, hybrid frameworks should integrate constraint validation with Path Computation Elements (PCEs) or SDN controllers.

Evaluation should compare hybrid methods against standalone LP/MILP, established heuristics, and learning-only approaches under equivalent network conditions. Important metrics include MLU, throughput, delay, packet loss, computational overhead, routing stability, and constraint satisfaction. Ablation studies should identify the contribution of individual components, while experiments under unseen traffic conditions and network failures should assess robustness and generalization.

Overall, hybrid AI and mathematical optimization offer a promising approach to scalable, adaptive, and constraint-aware traffic engineering. By combining data-driven decision-making with established optimization mechanisms, these frameworks can improve routing efficiency and computational performance. However, their practical benefits depend on forwarding compatibility, solution quality, computational costs, and rigorous experimental validation.

J. Multi-Agent and Transfer Learning Approaches

Multi-Agent Reinforcement Learning (MARL) and Transfer Learning (TL) offer promising approaches for improving the scalability, adaptability, and generalization of intelligent Traffic Engineering (TE) in Multiprotocol Label Switching (MPLS) and Segment Routing over MPLS (SR-MPLS) networks. MARL distributes routing decisions among cooperating agents, while TL reuses previously acquired knowledge to accelerate adaptation to changing network conditions. However, their effectiveness depends on coordination mechanisms, knowledge transferability, computational overhead, and forwarding compatibility.

Multi-Agent Reinforcement Learning. MARL decomposes network-wide optimization into smaller tasks assigned to agents controlling network regions, routing nodes, or traffic demands. Agents coordinate their actions to improve collective objectives, including load balancing, congestion reduction, and Quality of Service (QoS). A cooperative MARL objective can be expressed as:

$$\max_{\pi_1 \dots \pi_N} E \left[\sum_{t=0}^T \gamma^t R(s_t, a_t^1, \dots, a_t^N) \right]$$



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

where N represents the number of agents, π_i denotes the policy of agent i , s_t is the network state, a_t^i represents the selected action, R is the joint reward, and γ is the discount factor. Although cooperation can improve network-wide optimization, conflicting local decisions may introduce congestion or routing instability.

Guo et al. [45] proposed a Cooperative Multi-Agent Reinforcement Learning (CMRL) framework for distributed traffic engineering in hybrid Software-Defined Networking (SDN). Using partial link-utilization observations and difference-based rewards, the framework coordinated local routing decisions to improve network-wide performance. Experiments demonstrated advantages under changing traffic demands and network failures, although direct applicability to MPLS and SR-MPLS forwarding was not established.

Luan et al. [46] introduced MATE, a hierarchical multi-agent Deep Reinforcement Learning (DRL) framework for inter-domain routing and QoS-aware traffic allocation. The approach combined network abstraction, coordinated routing decisions, and multipath traffic splitting. Evaluations demonstrated favorable Maximum Link Utilization (MLU), QoS satisfaction, and robustness, achieving an approximation ratio below 1.3 in 97% of evaluated scenarios on a 16-domain topology. However, performance depended on topology abstraction, inter-agent coordination, and forwarding assumptions.

Cao et al. [47] proposed SROD-LC, integrating DRL with Load Centrality theory for Segment Routing optimization. The framework selected critical network nodes and employed cooperating agents to adjust Segment Identifier (SID) lists and redistribute traffic. Simulations using GEANT and germany50 topologies reported MLU reductions of approximately 21–35% relative to the evaluated baselines. Nevertheless, the findings were simulation-based, and direct SR-MPLS forwarding compatibility requires further verification.

Transfer Learning and Network Generalization. TL accelerates routing adaptation by transferring previously learned parameters, representations, or policies to new network environments. This approach may reduce retraining requirements following changes in topology, traffic demands, or link capacities. However, transfer effectiveness depends on similarities between source and target environments.

Awad et al. [48] developed DP-DRRouting, combining Deep Deterministic Policy Gradient (DDPG), double critics, prioritized experience replay, and policy transfer. Their SDN-based experiments reported throughput improvements of up to 50% and energy-efficiency improvements of up to 12% compared with Open Shortest Path First (OSPF), alongside improved adaptation following link failures. However, these results do not establish superiority over competitive MPLS or SR-MPLS optimization methods.

Ding et al. [49] proposed GROM, integrating Graph Neural Networks (GNNs) with DRL to improve routing generalization across network topologies. Experiments using five real-world topologies demonstrated the ability to generate routing policies for unseen networks without retraining under the evaluated conditions. Although promising, cross-topology generalization differs from conventional TL involving explicit knowledge transfer and target-domain adaptation. Furthermore, the study evaluated SDN routing rather than verified MPLS or SR-MPLS forwarding.

Challenges and Operational Considerations. MARL introduces challenges involving inter-agent communication, coordination overhead, convergence stability, and conflicting routing decisions. Centralized Training with Decentralized Execution (CTDE), parameter sharing, and hierarchical control can improve coordination but may increase computational and synchronization requirements.

TL faces additional limitations, including negative transfer and reduced effectiveness under substantially different network conditions. Consequently, transferred policies should be compared with models trained from scratch and established optimization baselines to assess adaptation efficiency and generalization.



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

In MPLS and SR-MPLS networks, routing decisions must satisfy bandwidth availability, Label Switched Path (LSP) feasibility, SID validity, Maximum SID Depth (MSD), and applicable forwarding-policy constraints. Evaluation should therefore consider MLU, throughput, delay, computational overhead, inter-agent communication, adaptation time, routing stability, and constraint satisfaction under unseen traffic conditions and network failures.

Overall, MARL and TL provide complementary mechanisms for distributed routing optimization and knowledge reuse. Existing studies demonstrate promising improvements in scalability, adaptability, and network performance. However, their practical effectiveness in MPLS and SR-MPLS requires architecture-specific validation, reliable coordination, robust generalization, and comparisons with competitive classical optimization methods. These considerations establish the foundation for the comparative and critical analysis presented in Section V.

V. COMPARATIVE AND CRITICAL ANALYSIS

The effectiveness of Artificial Intelligence (AI)-based Traffic Engineering (TE) in Multiprotocol Label Switching (MPLS) and Segment Routing over MPLS (SR-MPLS) networks requires systematic comparison with established routing optimization methods. Although Machine Learning (ML), Reinforcement Learning (RL), and Deep Reinforcement Learning (DRL) have demonstrated promising results in adaptive routing, congestion management, and resource allocation, their performance depends on network architecture, traffic conditions, optimization objectives, and experimental assumptions. Therefore, critical evaluation must consider solution quality, computational efficiency, scalability, generalization, routing stability, constraint satisfaction, and operational feasibility. Particular attention is required to distinguish direct MPLS/SR-MPLS evidence from findings obtained in broader Software-Defined Networking (SDN) and Internet Protocol (IP) environments.

A. Classical versus AI-Based Optimization

Classical and AI-based routing optimization differ in their decision-making mechanisms, computational requirements, and adaptability. Conventional approaches, including Constraint-Based Shortest Path First (CSPF), Linear Programming (LP), Integer Linear Programming (ILP), Mixed-Integer Linear Programming (MILP), and heuristic algorithms, employ explicitly defined objectives and constraints. In contrast, AI-based methods learn from historical measurements or network interactions to predict traffic conditions and generate routing decisions. While classical methods offer interpretable optimization models and explicit constraint handling, learning-based approaches may support faster decisions after training. However, neither approach guarantees superior performance across all networking conditions.

Optimization Quality and Baseline Selection. Classical shortest-path routing and Equal-Cost Multipath (ECMP) provide efficient forwarding but do not inherently optimize network-wide resource utilization. Mathematical programming and advanced heuristics offer stronger baselines by considering multiple traffic demands and routing constraints. Hartert et al. [50] introduced the Declarative and Expressive Forwarding Optimizer (DEFO), combining constraint programming with Segment Routing to optimize traffic distribution in carrier-grade networks. This demonstrates that conventional optimization can achieve substantial improvements without relying on AI.

Bernárdez et al. [25] compared a Multi-Agent Reinforcement Learning (MARL) and Graph Neural Network (GNN)-based traffic engineering approach with DEFO. Their experiments demonstrated broadly comparable optimization performance while reducing decision time from minutes in some DEFO scenarios to seconds. These findings suggest that learning-based methods can improve computational responsiveness, although their superiority depends on optimization objectives, network conditions, and comparison baselines.

Computational Efficiency and Adaptability. Classical optimization may require repeated computation as network conditions change, particularly for complex ILP and MILP formulations. In contrast, trained ML and DRL models can potentially generate routing decisions through rapid inference. Tian et al. [8] proposed Adpt-



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

SRTE, combining Proximal Policy Optimization (PPO) with action branching for adaptive Segment Routing. Their experiments reported Maximum Link Utilization (MLU) reductions of up to 33% relative to the evaluated baselines, alongside low online decision latency. However, computational comparisons must account for offline training, online inference, path validation, and policy-deployment costs.

Hybrid approaches further demonstrate the complementary roles of AI and mathematical optimization. Wang et al. [36] combined RL-based Smart Node Selection (SNS) with LP to reduce routing-computation complexity, while Wang et al. [42] employed historical optimization solutions to accelerate Segment Routing decisions. Although these methods improve computational efficiency under the evaluated conditions, candidate restrictions and historical-data dependence may reduce solution quality under unfamiliar traffic patterns.

Constraint Satisfaction and Forwarding Feasibility. Mathematical optimization can explicitly enforce link-capacity, flow-conservation, and routing constraints. Learning-based methods, however, may generate infeasible decisions unless supported by constraint-aware action selection or validation mechanisms. This distinction is particularly important in SR-MPLS, where routing configurations must satisfy Segment Identifier (SID) availability, Maximum SID Depth (MSD), and SR Policy requirements. Moreno et al. [51] demonstrated the importance of architecture-aware optimization through ILP formulations and heuristics for Segment Routing traffic engineering. Their findings emphasize the need to compare AI-based routing with conventional methods that account for practical forwarding constraints.

Scalability, Generalization, and Routing Stability. Classical optimization can accommodate new network instances without retraining, although computational requirements may increase with problem size. AI-based approaches may offer rapid decisions after training but can experience reduced performance under unseen topologies, changing traffic distributions, or network failures. GNN-based models offer potential improvements in cross-topology generalization [25], although such capabilities require independent experimental validation.

Both approaches may also introduce routing instability through frequent path reconfiguration. Classical reoptimization can trigger unnecessary routing changes, while RL and DRL policies may respond unpredictably to measurement noise or poorly designed rewards. Consequently, evaluations should consider routing-change frequency, convergence behavior, controller overhead, and policy-installation latency alongside conventional Quality of Service (QoS) metrics.

Comparative Synthesis. Table 3 summarizes the principal strengths and limitations of classical, AI-based, and hybrid optimization methods. The comparison is qualitative and does not represent a quantitative ranking of experimental performance.

Table 3. Comparative Assessment of Classical, AI-Based, and Hybrid Traffic Engineering Optimization

Evaluation Dimension	Classical Optimization	AI-Based Optimization	Hybrid Optimization
Solution quality	Exact, bounded, or approximate solutions, depending on the method	Depends on learning quality and generalization	Combines learned decisions with optimization
Computational efficiency	Repeated optimization may be expensive	Potentially rapid inference after training	Reduces search complexity but adds integration costs
Traffic adaptability	Requires updated inputs and reoptimization	Supports learned adaptive decisions	Combines adaptation with constrained optimization
Constraint satisfaction	Explicit enforcement of modeled constraints	Requires additional validation	Integrates learning with feasibility checks
Scalability	Depends on formulation and problem size	Depends on state/action representation	Supports decomposition and candidate reduction
Generalization	Can solve new network instances	Requires validation under unseen conditions	Depends on learning and optimization design



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

Routing stability	Depends on reoptimization policies	Depends on learned policy behavior	Can incorporate reconfiguration constraints
Interpretability	Explicit mathematical objectives and constraints	Complex learned policies may lack transparency	Optimization stages can improve interpretability
MPLS/SR-MPLS feasibility	Can incorporate forwarding constraints	Requires architecture-aware validation	Combines learning with constrained path computation

The comparison indicates that AI-based traffic engineering offers potential advantages in adaptability and online decision efficiency, whereas classical optimization retains strengths in explicit constraint enforcement and solution-quality assessment. Hybrid approaches may provide useful trade-offs by combining learning-based acceleration with mathematical optimization.

Nevertheless, meaningful comparisons require equivalent network topologies, traffic matrices, optimization objectives, computational budgets, and forwarding constraints. Evaluation should distinguish offline training, inference, solver execution, and policy-deployment overhead. Where suitable optimization bounds are available, solution-quality gaps should also be reported.

Overall, classical and AI-based optimization provide complementary rather than universally competing approaches to MPLS and SR-MPLS traffic engineering. Establishing their relative advantages requires rigorous architecture-specific benchmarking, realistic operational constraints, and reproducible experiments. These considerations provide the foundation for comparing individual ML, RL, and DRL approaches in Section V.B

B. *ML versus RL versus DRL*

Machine Learning (ML), Reinforcement Learning (RL), and Deep Reinforcement Learning (DRL) provide complementary approaches to intelligent Traffic Engineering (TE) in Multiprotocol Label Switching (MPLS) and Segment Routing over MPLS (SR-MPLS) networks. Conventional supervised and unsupervised ML primarily support traffic prediction, network-state characterization, and performance estimation. In contrast, RL learns adaptive routing policies through environmental interactions and reward feedback, while DRL extends RL using neural networks to address complex state and action spaces. Their effectiveness depends on learning requirements, computational efficiency, scalability, generalization, and forwarding feasibility.

Learning and Decision-Making Capabilities. Supervised ML predicts traffic demands and network-performance indicators using historical data, whereas unsupervised ML identifies recurring traffic patterns and operating conditions. Le et al. [26] demonstrated the value of prediction-assisted Segment Routing optimization by incorporating anticipated traffic variations into routing decisions.

Conventional RL algorithms, including Q-learning and SARSA, optimize routing through sequential interactions. Wumian et al. [35] proposed RLSR, a SARSA-based Segment Routing framework that reduced Maximum Link Utilization (MLU) from 71.67% to 50% under the evaluated conditions. However, tabular RL faces scalability limitations as network states and candidate routing actions increase.

DRL addresses some of these limitations through neural-network function approximation. Techniques such as Deep Q-Networks (DQNs), Proximal Policy Optimization (PPO), and actor-critic methods support high-dimensional network representations and more complex routing decisions. Tian et al. [8] introduced Adpt-SRTE, combining PPO with action branching and reporting MLU reductions of up to 33% relative to the evaluated baselines. Nevertheless, DRL introduces greater training complexity, hyperparameter sensitivity, and potential convergence instability.

Computational Efficiency and Generalization. Supervised ML typically provides rapid predictions after training but requires downstream routing optimization. RL learns routing policies through repeated interactions, while DRL may provide fast online decisions after computationally intensive training. Bernárdez et al. [25] demonstrated that combining Multi-Agent RL (MARL) with Graph Neural Networks (GNNs) could achieve



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

performance comparable to the classical DEFO optimizer while reducing decision time under selected network conditions.

Scalability remains challenging across all approaches. Conventional RL suffers from large state–action spaces, while ML and DRL models may require retraining or adaptation under unseen traffic distributions and topologies. GNN-based representations can improve structural generalization, but their effectiveness requires independent experimental validation.

Routing Feasibility and Operational Reliability. None of these learning approaches inherently guarantees feasible routing decisions. MPLS-TE requires valid Label Switched Paths (LSPs), bandwidth availability, and applicable signaling constraints, while SR-MPLS additionally requires valid Segment Identifier (SID) lists, Maximum SID Depth (MSD) compliance, and SR Policy compatibility. Therefore, intelligent routing systems require constraint-aware optimization, feasibility validation, and appropriate programmable control mechanisms.

Table 4. Comparative Characteristics of ML, RL, and DRL for MPLS and SR-MPLS Traffic Engineering

Dimension	Conventional ML	Conventional RL	DRL
Learning mechanism	Prediction and pattern discovery	Reward-based interaction	Neural-network-based RL
Primary TE function	Traffic prediction and characterization	Adaptive routing decisions	Complex routing optimization
Typical methods	RF, SVM, LSTM, PCA	Q-learning, SARSA	DQN, PPO, DDPG
Training requirements	Historical datasets	Repeated interactions	Extensive training and tuning
Scalability	Depends on model and data	Limited by state–action size	More flexible, but computationally demanding
Generalization	Data-dependent	Policy-dependent	Potentially improved through GNNs
Interpretability	Model-dependent	Tabular policies are inspectable	Often less transparent
Constraint satisfaction	Requires downstream validation	Requires explicit constraints	Requires explicit constraints
Main limitation	Prediction does not ensure better routing	Scalability and exploration	Training cost and stability

Overall, no learning paradigm is universally superior. ML supports predictive network management, RL enables sequential routing adaptation, and DRL provides more expressive policy approximation. However, fair comparisons require equivalent traffic conditions, optimization objectives, and forwarding constraints. Their practical effectiveness should be assessed through architecture-specific experiments and competitive classical optimization baselines.

C. Throughput, Delay, Packet Loss, and Link Utilization

Throughput, end-to-end delay, packet loss, and link utilization are fundamental performance indicators for evaluating intelligent Traffic Engineering (TE) in MPLS and SR-MPLS networks. Although ML, RL, and DRL methods have demonstrated improvements in selected metrics, their results depend on network topology, traffic conditions, optimization objectives, and experimental configurations. Consequently, improvements in one performance indicator should not be interpreted as simultaneous improvements across all Quality of Service (QoS) dimensions.

Link Utilization. Link utilization measures the proportion of available capacity consumed by traffic. Maximum Link Utilization (MLU) identifies the most heavily loaded link and is expressed as:

$$U_{max}(t) = \max_{e \in E} \frac{L_e(t)}{C_e}$$



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

where $L_e(t)$ represents traffic load on link e , and C_e denotes its capacity. Minimizing MLU can reduce congestion and improve resource distribution, although it does not necessarily minimize delay or maximize throughput.

Tian et al. [8] reported MLU reductions of up to 33% using Adpt-SRTE, while Wumian et al. [35] demonstrated a reduction from 71.67% to 50% using RL-based Segment Routing. Guo et al. [30] also reported improved MLU through hybrid clustering, DRL, and mathematical optimization. Similarly, Le et al. [26] achieved more than 80% of the reference optimal MLU-reduction performance using prediction-assisted routing. However, these findings are not directly comparable because of differences in experimental conditions and baselines.

Throughput. Throughput measures the amount of successfully delivered data over a specified period. Aggregate throughput is calculated as:

$$T_{agg} = \frac{\sum_{k \in K} B_k^{delivered}}{\Delta t}$$

where $B_k^{delivered}$ denotes successfully delivered data for demand k , and Δt represents the measurement interval.

Chen et al. [39] reported throughput improvements of up to 9% using DRL-based traffic splitting in hybrid IP/Segment Routing networks. Xu et al. [41] also demonstrated improvements in network utility and throughput-related performance using DRL-TE. Nevertheless, aggregate throughput may conceal unfair bandwidth allocation among competing flows.

End-to-End Delay. End-to-end delay includes propagation, transmission, processing, and queuing delays. Mean delay is expressed as:

$$D_{avg} = \frac{1}{N_{received}} \sum_{i=1}^{N_{received}} (t_i^{rx} + t_i^{tx})$$

where t_i^{rx} and t_i^{tx} denote packet transmission and reception times, respectively.

Xu et al. [41] demonstrated reduced end-to-end delay through actor-critic DRL, while Rusek et al. [27] introduced RouteNet, a GNN-based model for estimating network-performance indicators. However, prediction accuracy differs from actual delay reduction, and minimizing MLU may increase delay when alternative routes are longer. Evaluation should therefore consider mean delay, high-percentile latency, and jitter where relevant.

Packet Loss. Packet Loss Ratio (PLR) measures the proportion of transmitted packets that fail to reach their destination:

$$PLR = \frac{N_{sent} - N_{received}}{N_{sent}} \times 100\%$$

where N_{sent} and $N_{received}$ denote transmitted and successfully received packets within a consistently defined observation period.

Packet loss may result from congestion, buffer overflow, transmission errors, or network failures. Intelligent routing can potentially reduce congestion-related loss through adaptive traffic distribution. However, reduced MLU does not guarantee lower packet loss, particularly during short traffic bursts. Moreover, the reviewed Segment Routing studies report packet-loss outcomes less consistently than utilization metrics, limiting direct comparisons of reliability improvements.

Comparative Performance Evidence. Table 5 summarizes selected findings from representative intelligent traffic engineering studies.



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

Table 5. Selected Network-Performance Evidence from Intelligent Traffic Engineering Studies

Study	Approach	Reported outcome	Principal limitation
Le et al. [26]	Prediction-assisted SR-TE	More than 80% of reference optimal MLU-reduction performance	Prediction and monitoring conditions
Wumian et al. [35]	SARSA-based Segment Routing	MLU reduced from 71.67% to 50%	Experimental conditions and baseline
Tian et al. [8]	PPO-based DRL	MLU reduction up to 33%	Scenario-dependent
Chen et al. [39]	DRL traffic splitting	Throughput improvement up to 9%	Hybrid IP/SR simulation
Xu et al. [41]	Actor-critic DRL-TE	Reduced delay and improved network utility	Generic networking environment
Rusek et al. [27]	GNN-based RouteNet	Network-performance prediction	Prediction is not realized QoS improvement

Critical Evaluation. The reviewed studies demonstrate promising improvements in link utilization, throughput, and delay, although evidence for packet-loss reduction remains less consistent. These outcomes cannot be directly compared across studies using different traffic matrices, topologies, link capacities, transport protocols, and simulation configurations.

Furthermore, improving one metric may adversely affect another. For example, reducing MLU through longer alternative paths may increase propagation delay, while maximizing aggregate throughput may reduce fairness among traffic demands. Therefore, multiobjective optimization and explicit QoS constraints are important for balancing competing requirements.

Reliable comparisons should employ consistent metric definitions, equivalent experimental conditions, competitive baselines, and repeated experiments with appropriate variability reporting. Evaluation should also distinguish fluid-model estimates from packet-level measurements and account for routing changes, computational overhead, forwarding feasibility, and policy-deployment latency.

Overall, intelligent traffic engineering can improve selected network-performance metrics through prediction, adaptive routing, and learning-based resource allocation. However, no single performance indicator establishes comprehensive network improvement. Rigorous assessment requires evaluating multiple QoS dimensions alongside computational efficiency, routing stability, and architectural constraints. These considerations provide the basis for examining computational complexity and training requirements in Section V.D.

D. Computational Complexity and Training Overhead

Computational complexity and training overhead are critical factors in evaluating Artificial Intelligence (AI)-based Traffic Engineering (TE) in Multiprotocol Label Switching (MPLS) and Segment Routing over MPLS (SR-MPLS) networks. Although Machine Learning (ML), Reinforcement Learning (RL), and Deep Reinforcement Learning (DRL) may accelerate routing decisions after training, their efficiency depends on network size, model complexity, training requirements, and operational constraints. Therefore, comparisons with classical optimization must distinguish offline training costs, online inference latency, solver execution time, and routing-policy deployment overhead.

Classical Optimization Complexity. Conventional routing methods exhibit different computational requirements. Dijkstra's Shortest Path First (SPF) algorithm has a time complexity of $O((|V|+|E|)\log|V|)$ when implemented using a binary heap, while Constraint-Based Shortest Path First (CSPF) incorporates additional resource and policy constraints. Linear Programming (LP) admits polynomial-time solution algorithms, whereas Integer Linear Programming (ILP) and Mixed-Integer Linear Programming (MILP) may exhibit exponential worst-case complexity due to discrete routing decisions. Heuristic methods provide alternative trade-offs between execution time and solution quality.

Hartert et al. [50] introduced the Declarative and Expressive Forwarding Optimizer (DEFO), demonstrating that constraint-based Segment Routing optimization can generate effective configurations within seconds under the evaluated conditions. This finding emphasizes that AI methods should be compared with efficiently



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

implemented classical algorithms rather than assuming that conventional optimization is inherently computationally expensive.

ML, RL, and DRL Computational Requirements. Supervised and unsupervised ML introduce costs associated with data preparation, feature extraction, training, and hyperparameter tuning. Although trained models can rapidly predict traffic demands or network conditions, additional routing optimization may still be required.

Tabular RL approaches, including Q-learning and SARSA, face increasing memory and computational requirements as state-action spaces expand. DRL addresses some representation limitations through neural networks but introduces substantial training costs involving repeated interactions, gradient updates, and policy evaluation.

Tian et al. [8] proposed Adpt-SRTE, combining Proximal Policy Optimization (PPO) with action branching for adaptive Segment Routing. Their experiments reported online decision times within tens of milliseconds after offline training lasting up to several hours. These findings demonstrate the trade-off between training investment and online responsiveness, although overall efficiency depends on model reuse, retraining frequency, and deployment overhead.

Bernárdez et al. [25] compared a Multi-Agent Reinforcement Learning (MARL) and Graph Neural Network (GNN)-based approach with DEFO, reporting comparable traffic engineering performance while reducing decision time from minutes in some evaluated scenarios to seconds. However, distributed learning introduces communication and coordination costs that must be considered when assessing system-wide computational efficiency.

Hybrid AI and Mathematical Optimization. Hybrid approaches seek to reduce computational requirements by combining learning-based candidate selection with conventional optimization. Wang et al. [36] proposed Smart Node Selection (SNS), integrating RL with LP to reduce routing-search complexity. Similarly, Wang et al. [42] introduced RS4SR, which reuses historical optimization solutions to accelerate Segment Routing decisions, reporting near-optimal configurations within a two-second computation limit on large networks.

Lu et al. [43] developed Graph-based Reinforcement Learning for Traffic Engineering (GRL-TE), combining GNN-based representations, actor-critic learning, and the Alternating Direction Method of Multipliers (ADMM). In their largest evaluated topology of 1,739 nodes, LP required approximately 10,499 seconds, compared with 1.48 seconds for GRL-TE online inference and refinement after approximately 218 seconds of training. However, LP performed faster on smaller topologies, while GRL-TE achieved lower demand satisfaction in some scenarios. These findings demonstrate that computational advantages are scale-dependent and may involve compromises in optimization quality. Furthermore, the experiments were conducted in a general Software-Defined Networking (SDN) environment rather than a verified MPLS or SR-MPLS implementation.

Total Computational Cost and Operational Overhead. Evaluating AI-based optimization requires considering cumulative computational costs rather than inference latency alone. A simplified cost model is:

$$T_{AI,total}(N) = T_{data} + T_{train} + T_{tune} + \sum_{i=1}^N (T_{monitor,i} + T_{infer,i} + T_{validate,i} + T_{deploy,i}) + T_{retrain}$$

where N denotes the number of routing decisions, and the remaining terms represent data preparation, training, tuning, monitoring, inference, validation, deployment, and cumulative retraining costs. This simplified accounting model assumes additive processing costs; actual elapsed time may differ when operations overlap or execute in parallel.



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

The model highlights that rapid inference does not necessarily guarantee lower total computational cost. AI-based approaches become more attractive when training expenses are distributed across repeated routing decisions, whereas frequent retraining may reduce these advantages.

For MPLS-TE, deployment overhead includes Label Switched Path (LSP) establishment, resource verification, and applicable signaling procedures. SR-MPLS additionally requires Segment Identifier (SID) validation, Maximum SID Depth (MSD) compliance, and SR Policy installation. Consequently, online inference latency must be distinguished from complete decision-to-deployment time.

Comparative Synthesis. Table 6 summarizes the computational characteristics of classical and intelligent traffic engineering approaches.

Table 6. Comparative Computational Characteristics of Classical and AI-Based Traffic Engineering Methods

Approach	Offline requirements	Online requirements	Main limitation
SPF/CSPF	Topology and constraint preparation	Path computation	Limited global optimization
LP	Model formulation	Constrained optimization	Problem size and solver cost
ILP/MILP	Model and constraint specification	Discrete optimization	Combinatorial complexity
Supervised ML	Data preparation and training	Prediction and routing optimization	Prediction errors and retraining
Tabular RL	Exploration and policy learning	Action-value lookup	State-action-space growth
DRL	Extensive training and tuning	Inference and validation	Training cost and generalization
MARL	Multi-agent training	Coordinated inference	Communication and stability
Hybrid AI-optimization	Learning and solver preparation	Inference and optimization refinement	Combined processing overhead

Note: This qualitative comparison summarizes computational characteristics rather than providing an empirical ranking.

Critical Evaluation. The evidence indicates that classical optimization remains competitive for efficiently formulated routing problems, while AI-based methods may accelerate repeated online decisions after training. Hybrid approaches offer additional opportunities to reduce optimization complexity while preserving explicit constraint handling. However, these advantages depend on network scale, computational resources, training frequency, and solution quality.

Rigorous comparisons should report training duration, inference latency, solver execution time, computational resources, retraining costs, and policy-deployment overhead. Experiments should use comparable hardware and network conditions where possible, with differences documented. Evaluation must also consider the trade-off between execution time and routing quality, since faster decisions do not necessarily produce better network performance.

Overall, computational efficiency in intelligent MPLS and SR-MPLS traffic engineering depends on balancing training investment, online responsiveness, optimization quality, and deployment costs. Although ML, RL, and DRL can reduce repeated decision-making overhead, their practical advantages require reproducible, architecture-aware evaluations that measure complete routing-decision and deployment performance rather than isolated inference time.

E. Scalability and Generalization

Scalability and generalization are essential for evaluating Artificial Intelligence (AI)-based Traffic Engineering (TE) in Multiprotocol Label Switching (MPLS) and Segment Routing over MPLS (SR-MPLS) networks. Scalability concerns an algorithm's ability to maintain computational efficiency and routing performance as network size, traffic demands, and optimization complexity increase. Generalization refers to its effectiveness under unseen traffic patterns, network topologies, and failure conditions. Although Machine Learning (ML), Reinforcement Learning (RL), and Deep Reinforcement Learning (DRL) offer promising



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

optimization capabilities, their effectiveness depends on model architecture, state-action representations, training diversity, and forwarding constraints.

Computational and Action-Space Scalability. Classical optimization methods, including Linear Programming (LP), Mixed-Integer Linear Programming (MILP), and heuristics, face increasing computational demands as routing variables and constraints expand. Learning-based methods can reduce repeated optimization through trained policies, but their training complexity and action spaces may also increase substantially. Tabular RL suffers from state-action-space growth, while DRL uses neural-network approximation to improve representation capabilities without eliminating combinatorial routing complexity.

Tian et al. [8] introduced Adpt-SRTE, combining Proximal Policy Optimization (PPO) with action branching to manage large discrete routing-action spaces. Their experiments reported Maximum Link Utilization (MLU) reductions of up to 33% and adaptation across different traffic periods. However, these results do not establish unrestricted scalability or generalization across arbitrary network topologies.

Wang et al. [36] proposed Smart Node Selection (SNS), integrating RL-based candidate selection with mathematical optimization to reduce computational complexity. Similarly, Wang et al. [42] developed RS4SR, which reuses historical routing solutions and flow clustering to accelerate Segment Routing optimization, reporting near-optimal solutions within a two-second computational budget. Although these approaches improve efficiency, candidate restrictions and dependence on historical traffic patterns may reduce solution quality under unfamiliar conditions.

Graph-Based and Multi-Agent Generalization. Graph Neural Networks (GNNs) offer mechanisms for representing network structures independently of fixed-dimensional topology descriptions. Ding et al. [49] developed GROM, combining GNNs and DRL to generate routing policies for previously unseen network topologies without retraining under the evaluated conditions. Experiments involving five real-world topologies demonstrated promising cross-topology generalization. However, these findings concern Software-Defined Networking (SDN) rather than verified MPLS or SR-MPLS forwarding.

Lu et al. [43] introduced GRL-TE, integrating graph-based learning, actor-critic optimization, and the Alternating Direction Method of Multipliers (ADMM). Evaluations across six Wide Area Network (WAN) topologies ranging from 12 to 1,739 nodes reported an average demand-satisfaction rate of 89.36% and substantial computational improvements over LP-based methods on larger networks. Nevertheless, faster optimization may involve solution-quality trade-offs, and testing multiple network sizes does not necessarily demonstrate zero-shot generalization.

Hierarchical Multi-Agent Reinforcement Learning (MARL) provides another scalability mechanism. Luan et al. [46] proposed MATE, combining topology abstraction and coordinated traffic allocation across multiple domains. In a 16-domain topology, the framework achieved an approximation ratio below 1.3 in 97% of evaluated scenarios. However, its effectiveness depends on coordination overhead, topology abstraction, and inter-domain communication requirements.

Traffic Generalization and Failure Robustness. Intelligent routing models may perform well on trained network conditions but deteriorate under unseen traffic demands, topology modifications, or link failures. Historical-solution reuse [42] and prediction-driven optimization [8] can improve routing efficiency, although their effectiveness depends on the similarity between training and deployment conditions.

Bernárdez et al. [25] demonstrated promising GNN-based routing optimization across three real-world network topologies, while Ding et al. [49] explicitly examined routing decisions for unseen topologies. Nevertheless, generalization must distinguish direct deployment without retraining, transfer learning with adaptation, and independent model training. Robustness should also be evaluated under demand surges, link failures, and topology changes using consistent performance metrics.

Architectural Constraints and Forwarding Feasibility. Generalization across abstract network representations does not guarantee deployable MPLS or SR-MPLS routing configurations. Conventional MPLS-TE requires



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

feasible Label Switched Paths (LSPs), bandwidth availability, and applicable signaling procedures. SR-MPLS additionally requires valid Segment Identifier (SID) lists, Maximum SID Depth (MSD) compliance, and SR Policy compatibility, as addressed by the SR Policy architecture in RFC 9256. Consequently, scalability and generalization evaluations must incorporate forwarding feasibility, routing stability, and policy-deployment requirements.

Comparative Synthesis. Table 7 summarizes representative studies addressing scalability and generalization in intelligent traffic engineering.

Table 7. Scalability and Generalization Characteristics of Selected Intelligent Traffic Engineering Approaches

Study	Strategy	Reported evidence	Principal limitation
Bernárdez et al. [25]	MARL and GNN optimization	Three real-world topologies	Zero-shot transfer not established
Tian et al. [8]	PPO with action branching	Adaptive SR-TE across traffic periods	Cross-topology generalization unverified
Wang et al. [42]	Historical-solution reuse	Near-optimal solutions within two seconds	Dependence on historical patterns
Luan et al. [46]	Hierarchical MARL	Evaluation across 16 domains	Coordination overhead
Ding et al. [49]	GNN-based generalization	Five real-world topologies, including unseen networks	Generic SDN architecture
Lu et al. [43]	GNN, actor-critic, and ADMM	Evaluations across 12–1,739 nodes	Solution-quality trade-offs

Note: The table provides a qualitative synthesis of reported findings. Differences in network architectures, experimental configurations, and optimization objectives limit direct comparisons.

Critical Evaluation. Scalability and generalization are related but distinct characteristics. A model may optimize large networks efficiently while requiring retraining for each topology, whereas another may generalize across different networks but incur substantial computational or communication overhead. Therefore, comprehensive evaluation should examine network size, traffic-demand complexity, unseen topologies, distribution shifts, and failure recovery independently.

Reliable experiments should employ competitive classical baselines, independent training and testing datasets, repeated trials, and transparent reporting of computational resources. Relevant indicators include MLU, throughput, delay, training duration, inference latency, memory consumption, communication overhead, routing stability, and constraint satisfaction. Appropriate statistical analyses and sensitivity tests are also necessary to establish the robustness of reported improvements.

Overall, GNN-based representations, hierarchical MARL, action-space decomposition, and learning-assisted mathematical optimization offer promising approaches to improving scalability and generalization. Nevertheless, their operational effectiveness in MPLS and SR-MPLS remains dependent on architectural compatibility, training diversity, constraint enforcement, and rigorous experimental validation. These considerations establish the foundation for examining routing stability and constraint satisfaction in Section V.F.

Routing Stability and Constraint Satisfaction

Routing stability and constraint satisfaction are essential for reliable Artificial Intelligence (AI)-based Traffic Engineering (TE) in Multiprotocol Label Switching (MPLS) and Segment Routing over MPLS (SR-MPLS) networks. Although Machine Learning (ML), Reinforcement Learning (RL), and Deep Reinforcement Learning (DRL) can improve routing adaptability and resource utilization, frequent or infeasible routing changes may introduce oscillations, packet reordering, control-plane overhead, and service disruption. Therefore, routing optimization must balance network-performance improvements with configuration stability and compliance with operational constraints.

Routing Stability and Reconfiguration Costs. Dynamic traffic engineering may trigger frequent path adjustments in response to changing traffic demands or network conditions. Both classical optimization and



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

learning-based approaches can produce unnecessary reconfigurations when responding to minor measurement variations. A stability-aware optimization objective can be formulated as:

$$\min_{x(t)} [\alpha U_{max}(t) + \beta R(t)]$$

subject to

$$X(t) \in X_{feasible}(t)$$

where $U_{max}(t)$ denotes Maximum Link Utilization (MLU), $R(t)$ represents routing-reconfiguration cost, α and β are nonnegative weighting parameters. For traffic-splitting configurations, reconfiguration cost can be expressed as:

$$R(t) = \frac{1}{2} \sum_{k \in K} wk \sum_{p \in P_k} [x_{kp}(t) - x_{kp}(t-1)]$$

where $x_{kp}(t)$ denotes the proportion of demand k allocated to path p , and wk represents the associated reconfiguration weight. The factor $(1/2)$ avoids double-counting redistributed traffic proportions. This formulation balances congestion reduction and routing stability while maintaining explicit feasibility constraints.

Le et al. [26] proposed prediction-assisted multi-time-step Segment Routing optimization to reduce unnecessary path changes and monitoring overhead. Their experiments achieved more than 80% of reference optimal MLU-reduction performance while limiting routing disturbances. However, effectiveness depended on prediction accuracy and traffic variability.

Similarly, Tian et al. [8] developed Adpt-SRTE, combining Proximal Policy Optimization (PPO) with action branching for adaptive Segment Routing. Their results demonstrated improved link utilization with relatively low path-configuration overhead. Nevertheless, routing stability under unexpected failures, topology changes, and transient forwarding conditions requires further validation.

Constraint Satisfaction in MPLS and SR-MPLS. Classical optimization methods, including Linear Programming (LP), Integer Linear Programming (ILP), and Mixed-Integer Linear Programming (MILP), can explicitly enforce link-capacity, flow-conservation, and routing constraints. However, mathematical feasibility does not automatically guarantee successful deployment when models omit forwarding-device limitations or rely on outdated network information.

In conventional MPLS-TE, Constraint-Based Shortest Path First (CSPF) supports resource-aware path computation, while Resource Reservation Protocol–Traffic Engineering (RSVP-TE) provides signaling and resource reservation. RFC 3209 additionally specifies make-before-break rerouting mechanisms that can reduce disruption during Label Switched Path (LSP) transitions.

SR-MPLS introduces further requirements involving Segment Identifier (SID) validity, Maximum SID Depth (MSD), forwarding-platform capabilities, and SR Policy selection. RFC 9256 defines the SR Policy architecture and candidate-path validity procedures, while RFC 8664 specifies Segment Routing extensions to the Path Computation Element Communication Protocol (PCEP). These mechanisms support feasible path computation and control but do not independently guarantee bandwidth availability or optimal network performance.

Moreno et al. [51] demonstrated architecture-aware optimization through ILP formulations and heuristics that account for Segment Routing label-stack restrictions. Their findings emphasize that effective routing configurations must satisfy both optimization objectives and forwarding constraints.

AI-Based Constraint Enforcement and Operational Stability. RL and DRL generate routing decisions through learned policies rather than explicitly solving constrained mathematical models. Although reward penalties can discourage infeasible actions, they cannot guarantee compliance with mandatory constraints. Consequently,



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

practical implementations require feasible-action masking, candidate-path filtering, constrained optimization, or post-decision validation.

Routing stability also depends on accurate network measurements, controller synchronization, and reliable policy deployment. Delayed telemetry or inconsistent configuration updates may produce forwarding behavior that differs from intended routing decisions. Stability safeguards, including hysteresis thresholds, minimum holding times, and controlled policy updates, can reduce unnecessary reconfiguration but may delay responses to genuine congestion or failures.

Comparative Synthesis. Table 8 summarizes the principal stability characteristics and constraint-handling mechanisms of classical and intelligent traffic engineering approaches.

Table 8. Comparative Assessment of Routing Stability and Constraint Satisfaction in Traffic Engineering

Approach	Stability characteristics	Constraint handling	Main limitation
Classical MPLS-TE	Controlled LSP reoptimization	CSPF, RSVP-TE, resource validation	Signaling and reconfiguration overhead
Classical SR-MPLS	Explicit paths and policy selection	SID validation, MSD limits	Platform and state dependencies
Supervised ML-assisted TE	Prediction can reduce reactive changes	Downstream constrained optimization	Forecast errors and model drift
Tabular RL	Feedback-driven adaptive routing	Restricted actions and validation	Exploration and oscillations
DRL	Complex adaptive routing policies	Action masking and feasibility checks	Policy instability and constraint violations
Hybrid AI-optimization	Learned decisions with mathematical refinement	LP/MILP and path validation	Combined computational overhead

Note: The table presents a qualitative comparison rather than an empirical ranking.

Critical Evaluation. Routing stability and constraint satisfaction represent distinct but complementary requirements. A mathematically feasible configuration may cause excessive reconfiguration, while a stable routing policy may become inefficient following traffic surges or network failures. Therefore, evaluating intelligent TE requires measuring routing-change frequency, policy-installation latency, convergence, traffic-redistribution magnitude, transient packet loss, and constraint-violation rates.

Comparative experiments should also assess the proportion of AI-generated routing actions that satisfy forwarding constraints and document rejected, corrected, or successfully deployed decisions. Evaluations using abstract routing models without realistic signaling, device capabilities, or policy-installation behavior provide limited evidence of operational reliability.

Overall, stable and constraint-aware routing is essential for translating intelligent optimization into practical MPLS and SR-MPLS deployments. Classical methods offer explicit constraint-handling mechanisms, while AI approaches provide adaptive decision-making capabilities. Hybrid frameworks combining learning-based optimization with feasibility validation and controlled policy deployment offer a promising direction for improving operational reliability.

F. Fault Tolerance and Resilience

Fault tolerance and resilience are essential for reliable Traffic Engineering (TE) in Multiprotocol Label Switching (MPLS) and Segment Routing over MPLS (SR-MPLS) networks. Fault tolerance concerns maintaining service continuity during component failures, whereas resilience encompasses failure detection, protection, restoration, and adaptation to degraded network conditions. Machine Learning (ML), Reinforcement Learning (RL), and Deep Reinforcement Learning (DRL) offer opportunities for failure-aware routing and resource reallocation. However, their effectiveness depends on recovery time, network-state accuracy, forwarding feasibility, and operational stability. Therefore, improvements in congestion or link utilization under normal conditions do not necessarily demonstrate resilience during failures.



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

Conventional Fault Protection and Recovery. MPLS provides established fault-recovery mechanisms through Fast Reroute (FRR). RFC 4090 specifies one-to-one backup and facility-backup protection for Resource Reservation Protocol–Traffic Engineering (RSVP-TE) Label Switched Paths (LSPs), allowing a Point of Local Repair (PLR) to redirect traffic around failed links or nodes. These mechanisms can support recovery within tens of milliseconds under suitable conditions, although actual performance depends on failure detection, backup-path availability, hardware capabilities, and resource constraints.

In SR-MPLS, Topology Independent Loop-Free Alternate (TI-LFA), specified in RFC 9855, supports local protection through precomputed repair paths. Unlike controller-dependent restoration, TI-LFA can redirect traffic immediately following failure detection without waiting for network-wide reoptimization. However, restored connectivity does not necessarily guarantee optimal traffic distribution or Quality of Service (QoS).

Bidirectional Forwarding Detection (BFD), defined in RFC 5880, supports rapid failure detection. A general service-recovery metric is expressed as:

$$T_{\text{recovery}} = t_{\text{service restored}} - t_{\text{failure}}$$

where t_{failure} denotes the failure occurrence time and $t_{\text{service restored}}$ represents the time at which the

affected service satisfies its defined restoration criterion. Recovery time must be distinguished from failure-detection latency, routing-algorithm execution time, and subsequent traffic reoptimization.

AI-Assisted Failure Recovery. Learning-based techniques can improve post-failure routing by identifying alternative paths and reallocating traffic according to updated network conditions. Ma et al. [52] proposed FRRL, an RL-based recovery framework for hybrid Software-Defined Networking (SDN). The approach uses a Topology Difference Vector (TDV) to represent failed links and generate routing decisions under modified network conditions. Experiments involving single-link and multiple-link failures demonstrated favorable performance against the evaluated baselines. However, the findings concern hybrid SDN rather than verified MPLS or SR-MPLS recovery, and rapid policy computation should not be equated with hardware-level FRR.

Farhan et al. [53] developed RDG-TE, combining Graph Neural Networks (GNNs) and DRL for reliability-aware traffic engineering. The framework incorporates link reliability, utilization, network centrality, and path length into routing optimization. Experimental results reported a 24.19% reduction in disturbed flows, a 35.38% reduction in hop count, and a 40.17% improvement in reliable-route prediction accuracy relative to the evaluated comparison approach. Nevertheless, these improvements do not independently establish service-restoration guarantees or MPLS-specific forwarding compatibility.

Multi-agent optimization provides additional opportunities for resilient routing. Luan et al. [46] introduced MATE, a hierarchical multi-agent DRL framework that coordinates traffic allocation across network domains. Evaluations demonstrated robustness under changing traffic conditions and random link failures. Similarly, Lu et al. [43] evaluated GRL-TE under multiple-link failures in large-scale Wide Area Networks (WANs). In the ASN topology, demand satisfaction decreased from 87.59% without failures to 85.10% under 200 simulated link failures. Although these findings indicate robustness under degraded conditions, demand satisfaction does not measure service-restoration time or guarantee forwarding continuity.

Hybrid Protection and Intelligent Reoptimization. Combining conventional protection mechanisms with AI-based optimization offers a promising resilience architecture. RSVP-TE FRR and TI-LFA can provide rapid local protection, while RL or DRL controllers subsequently optimize traffic distribution across the remaining network resources. This separation enables immediate service restoration followed by congestion reduction and resource reallocation. However, such integration requires reliable telemetry, controller availability, path validation, and coordinated policy deployment. Its effectiveness must be experimentally established rather than inferred from the independent capabilities of protection and optimization mechanisms.



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

Multiple Failures and Architectural Constraints. Simultaneous link failures, node outages, and Shared Risk Link Group (SRLG) failures can invalidate multiple forwarding paths. Learning models trained primarily on single-failure scenarios may perform poorly under compound failures or unexpected topology partitions. Consequently, resilience evaluation should include diverse failure patterns and account for the availability of feasible alternative paths.

In SR-MPLS, RFC 9256 defines SR Policy candidate-path validity and selection mechanisms, supporting fallback to alternative valid candidates when available. Nevertheless, SR Policy selection differs from local FRR and does not guarantee instantaneous restoration. AI-generated replacement paths must also satisfy Segment Identifier (SID) availability, Maximum SID Depth (MSD), and forwarding-device constraints.

Comparative Synthesis. Table 9 summarizes the principal fault-tolerance and resilience mechanisms relevant to classical and intelligent traffic engineering.

Table 9. Comparative Assessment of Fault-Tolerance and Resilience Mechanisms in Traffic Engineering

Approach	Recovery mechanism	Main advantage	Principal limitation
RSVP-TE FRR	Pre-established backup LSPs	Rapid local protection	Resource and signaling requirements
SR-MPLS TI-LFA	Precomputed local repair paths	Fast local recovery	Topology and hardware constraints
Classical reoptimization	Recomputed feasible paths	Explicit constraint handling	Computation and deployment delays
RL/DRL restoration	Failure-aware routing policies	Adaptive post-failure optimization	Training coverage and path validation
Multi-agent optimization	Coordinated traffic reallocation	Distributed adaptation	Communication and coordination overhead
Hybrid protection and AI	Local protection followed by AI optimization	Combines rapid recovery with resource optimization	Integration and operational complexity

Note: This table presents a qualitative comparison of resilience mechanisms rather than an empirical ranking of recovery performance.

Critical Evaluation. Fault tolerance, recovery responsiveness, and post-failure optimization quality represent distinct evaluation dimensions. Conventional FRR mechanisms prioritize immediate forwarding continuity, whereas AI-based approaches primarily support adaptive routing and resource optimization following network changes. A method that reduces congestion after a failure may still experience service interruption, while rapid local protection may maintain connectivity without restoring optimal traffic distribution.

Rigorous evaluation should measure failure-detection time, local protection-switching time, service-restoration time, transient packet loss, throughput degradation, post-failure Maximum Link Utilization (MLU), and subsequent reoptimization latency. Experiments should consider single-link, multiple-link, node, and SRLG failures under realistic forwarding constraints. Packet-level simulation, network emulation, and hardware-based testing can provide stronger evidence of operational resilience than abstract routing optimization alone.

Overall, AI-assisted fault tolerance offers promising capabilities for failure-aware path selection, traffic redistribution, and post-failure resource management. However, the available studies do not demonstrate that learning-based optimization can independently replace established MPLS and SR-MPLS fast-protection mechanisms. Hybrid architectures combining local protection with constraint-aware intelligent reoptimization represent a promising direction, provided their effectiveness is validated through realistic failure scenarios, forwarding constraints, and comprehensive recovery measurements.

G. Experimental Platforms and Simulation Environments

Experimental platforms are essential for evaluating the performance, scalability, and operational feasibility of Artificial Intelligence (AI)-based Traffic Engineering (TE) in Multiprotocol Label Switching (MPLS) and Segment Routing over MPLS (SR-MPLS) networks. Machine Learning (ML), Reinforcement Learning (RL), and Deep Reinforcement Learning (DRL) approaches are commonly evaluated through mathematical simulation, discrete-event simulation, network emulation, and physical testbeds. These environments differ in their ability to



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

represent traffic dynamics, protocol behavior, forwarding constraints, and deployment overhead. Consequently, experimental findings must be interpreted according to the capabilities and limitations of the selected platform.

Mathematical and Algorithmic Simulation. Mathematical simulation represents networks as graphs containing routers, links, capacities, and traffic demands. Environments such as Python and MATLAB, combined with optimization solvers including Gurobi and CPLEX, support routing optimization, traffic allocation, Maximum Link Utilization (MLU) analysis, and comparisons with Linear Programming (LP), Mixed-Integer Linear Programming (MILP), and heuristic baselines.

Tian et al. [8] evaluated Adpt-SRTE using real-world network topologies and historical Traffic Matrices (TMs), while Guo et al. [30] investigated hybrid Segment Routing optimization using Abilene, CERNET, and GEANT topologies. Similarly, Wang et al. [36, 42] examined learning-assisted optimization using representative network demands. Although these approaches provide valuable evidence of computational efficiency and optimization quality, they generally do not establish packet-level forwarding behavior or operational MPLS/SR-MPLS compatibility.

Discrete-Event Network Simulation. Discrete-event simulators represent packet transmission, queuing, transport protocols, and network events. The ns-3 simulator supports packet-level evaluation of throughput, delay, packet loss, and network failures. Xu et al. [41] used ns-3 to evaluate DRL-TE under different network conditions, demonstrating its suitability for studying packet-level performance. However, MPLS and SR-MPLS functionality requires appropriate models or extensions and cannot be assumed from the simulator alone.

OMNeT++, particularly with the INET framework, provides modular protocol and network simulation capabilities. Its MPLS-related models, including Label Distribution Protocol (LDP) and Resource Reservation Protocol-Traffic Engineering (RSVP-TE), support conventional MPLS evaluation where available. However, SR-MPLS functionality depends on the software version and implemented extensions. Both platforms provide greater packet-level detail than abstract optimization models, although simulation accuracy remains dependent on protocol implementation and configuration assumptions.

Network Emulation and Programmable Test Environments. Network emulation uses real networking software to examine forwarding behavior and controller interactions. Mininet provides virtual networking environments for Software-Defined Networking (SDN) experimentation. Zuo et al. [29] employed Mininet to evaluate learning-based path planning. Nevertheless, MPLS signaling and SR-MPLS forwarding require additional compatible implementations.

Container-based platforms such as containerlab, combined with suitable routing software, including FRRouting (FRR), can support architecture-aware experiments involving routing protocols, Segment Identifier (SID) lists, and SR Policies. However, functionality depends on software versions, operating-system capabilities, and supported forwarding mechanisms.

Commercial platforms, including OPNET Modeler and Riverbed Modeler, provide network and application-level simulation capabilities relevant to conventional MPLS Traffic Engineering and Quality of Service (QoS) analysis. Nevertheless, Segment Routing support and AI integration must be verified for the specific software version. Licensing requirements and proprietary models may also limit experimental reproducibility.

Physical Network Testbeds. Physical testbeds offer stronger evidence of operational feasibility by evaluating forwarding devices, programmable controllers, Path Computation Elements (PCEs), and traffic generators. These environments enable measurements of policy-installation latency, throughput, delay, packet loss, recovery time, and forwarding stability. For SR-MPLS, they can additionally verify Maximum SID Depth (MSD), segment-stack processing, policy validity, and device interoperability. However, hardware costs, configuration complexity, and limited scalability restrict their widespread use.

Comparative Synthesis. Table 10 summarizes the principal capabilities and limitations of experimental platforms used in intelligent traffic engineering research.



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

Table 10. Comparative Assessment of Experimental Platforms for Intelligent MPLS and SR-MPLS Traffic Engineering

Platform	Evaluation capabilities	Principal strengths	Main limitations
Mathematical simulation (Python, MATLAB)	Routing optimization, MLU, computational cost	Scalable and configurable	Limited packet-level realism
ns-3	Packet-level QoS, failures, traffic dynamics	Detailed discrete-event simulation	MPLS/SR-MPLS requires suitable models
OMNeT++/INET	Protocol behavior, QoS, MPLS mechanisms	Modular network simulation	Version-dependent SR-MPLS support
OPNET/Riverbed Modeler	Protocol and application simulation	Commercial network-model libraries	Licensing and version restrictions
Mininet	SDN control and virtual forwarding	Real Linux networking components	No automatic MPLS/SR-MPLS support
Containerlab with routing software	Virtual routing, SR Policies, control-plane testing	Architecture-aware emulation	Software and host-resource dependencies
Physical testbeds	Hardware forwarding, interoperability, recovery	Strong operational validation	High cost and limited scalability

Note: Platform capabilities depend on software versions, available protocol models, extensions, and forwarding implementations. The comparison does not imply native support for all MPLS or SR-MPLS features.

Experimental Design and Reproducibility. Reliable evaluation requires representative network topologies, realistic traffic demands, competitive optimization baselines, and transparent experimental configurations. Public resources such as the Internet Topology Zoo [54] and SNDlib provide useful topology and optimization datasets. However, realistic network topologies do not necessarily imply realistic traffic conditions, particularly when demand matrices are synthetically generated.

Comparative experiments should use equivalent topologies, traffic matrices, link capacities, optimization objectives, and forwarding constraints. ML, RL, and DRL studies should document training procedures, hyperparameters, random seeds, convergence criteria, and computational resources. Independent training and testing datasets, repeated experiments, confidence intervals, and sensitivity analyses are important for evaluating robustness and avoiding misleading performance conclusions.

Experimental Fidelity and Deployment Feasibility. Three levels of validation can be distinguished: (1) algorithmic feasibility, assessed through mathematical optimization or abstract simulation; (2) protocol and forwarding feasibility, examined through architecture-aware packet-level simulation or network emulation; and (3) operational feasibility, evaluated through realistic deployments incorporating controller interactions, device capabilities, and policy-installation behavior.

These levels are analytical distinctions rather than a formally validated evidence-quality scale. Importantly, reduced MLU or inference latency in simulation does not automatically demonstrate deployable MPLS or SR-MPLS routing configurations. Architecture-specific validation must therefore consider Label Switched Path (LSP) feasibility, SID availability, MSD limitations, SR Policy requirements, and forwarding stability.

Overall, no single experimental platform captures every aspect of intelligent traffic engineering. Mathematical simulation supports scalable optimization analysis, packet-level simulation evaluates network-performance behavior, and emulation or hardware testbeds provide stronger evidence of forwarding feasibility. Combining these approaches can improve experimental reliability and practical relevance. Future research should emphasize reproducible configurations, realistic traffic workloads, competitive baselines, and transparent reporting of platform limitations.

H. Reproducibility and Statistical Validity

Reproducibility and statistical validity are essential for evaluating Artificial Intelligence (AI)-based Traffic Engineering (TE) in Multiprotocol Label Switching (MPLS) and Segment Routing over MPLS (SR-MPLS) networks. Although Machine Learning (ML), Reinforcement Learning (RL), and Deep Reinforcement Learning



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

(DRL) have demonstrated improvements in routing efficiency and network performance, their reliability depends on experimental transparency, independent repetitions, statistical uncertainty, and fair comparisons. Stochastic training, random exploration, hyperparameter selection, and simulation variability can substantially influence reported results. Consequently, favorable outcomes from individual experiments do not necessarily establish consistent algorithmic superiority.

Experimental Reproducibility and Transparency. Reproducible intelligent TE experiments require detailed documentation of network topologies, link capacities, Traffic Matrices (TMs), routing constraints, simulation environments, model configurations, hyperparameters, random seeds, and computational resources. MPLS and SR-MPLS evaluations additionally require information about Label Switched Path (LSP) configurations, Segment Identifier (SID) assignments, Maximum SID Depth (MSD), and forwarding-policy constraints.

The Association for Computing Machinery (ACM) distinguishes repeatability by the original research team, reproducibility by independent researchers using the original experimental setup, and replicability through independently developed implementations [61]. These distinctions emphasize the importance of accessible experimental artifacts and sufficiently documented procedures.

Experimental Variability and Statistical Uncertainty. RL and DRL outcomes may vary because of random initialization, exploration policies, experience sampling, and changing traffic conditions. Henderson et al. [56] demonstrated the sensitivity of DRL performance to experimental randomness and implementation choices, while Patterson et al. [58] emphasized rigorous experimental design and statistical evaluation. Consequently, intelligent TE studies should conduct independent training runs and distinguish model-training variability from network-simulation randomness.

Agarwal et al. [57] further highlighted the limitations of evaluating reinforcement learning algorithms using few experimental runs. Reliable comparisons should report central performance estimates alongside standard deviations, confidence intervals, or other suitable uncertainty measures. Paired statistical tests or appropriate resampling methods can support comparisons when algorithms are evaluated under identical network conditions. However, statistical analysis must account for dependencies among observations, particularly when multiple measurements originate from the same simulation or traffic scenario.

Practical Significance and Baseline Fairness. Statistical significance alone does not establish operational importance. Performance improvements should be reported using absolute and relative differences, uncertainty estimates, and their practical implications for Maximum Link Utilization (MLU), throughput, delay, packet loss, and computational efficiency.

Fair benchmarking also requires equivalent network topologies, traffic demands, optimization objectives, forwarding constraints, and reasonably comparable computational resources. AI-based approaches should be compared with competitive classical methods, including Constraint-Based Shortest Path First (CSPF), Linear Programming (LP), Mixed-Integer Linear Programming (MILP), and established heuristics. Differences in available network information, training budgets, or baseline configurations must be disclosed to prevent misleading conclusions.

Data Separation and Model Generalization. Proper training, validation, and testing separation is essential for evaluating learning-based routing. Temporally correlated traffic measurements may introduce information leakage when randomly divided into training and testing datasets. Chronological testing provides a more realistic assessment of future traffic prediction, while cross-topology generalization requires evaluating models on networks excluded from training.

RL and DRL studies should also separate training episodes from independent evaluation scenarios and document hyperparameter-search procedures, stopping criteria, and model-selection methods. Ablation studies can identify whether improvements result from the learning algorithm itself or supporting components such as traffic prediction, graph-based representations, candidate-path restrictions, and mathematical optimization.



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

Architectural Reproducibility and Research Artifacts. Reproducibility depends on documenting simulation-platform versions, software dependencies, network configurations, traffic-generation procedures, and computational hardware. In MPLS and SR-MPLS networks, studies must additionally specify signaling assumptions, forwarding capabilities, SID lists, SR Policy configurations, and path-validation mechanisms. Fault-tolerance experiments should report failure-injection methods, detection intervals, backup-path availability, and restoration criteria.

Sharing source code, topology files, traffic datasets, model parameters, experimental scripts, and raw results can facilitate independent verification. The ACM Artifact Review and Badging framework [61] provides relevant principles for evaluating artifact availability and independent validation. Nevertheless, artifact availability alone does not guarantee reproducibility, particularly when experiments depend on proprietary software, restricted datasets, or specialized hardware.

Comparative Synthesis. Table 11 presents proposed methodological criteria for assessing reproducibility and statistical validity in intelligent traffic engineering studies.

Table 11. Proposed Criteria for Assessing Reproducibility and Statistical Validity in Intelligent Traffic Engineering Studies

Assessment dimension	Required evidence	Potential weakness
Network configuration	Topology, capacities, traffic demands, constraints	Incomplete network description
Simulation environment	Platform, version, extensions, configurations	Irreproducible experimental setup
Learning configuration	Model, hyperparameters, rewards, training procedures	Limited algorithm transparency
Experimental repetitions	Independent runs, seeds, testing scenarios	Insufficient repetitions
Statistical reporting	Performance estimates, variability, confidence intervals	Unsupported performance claims
Baseline fairness	Equivalent objectives, inputs, constraints, and budgets	Weak or unequal comparisons
Data separation	Independent training, validation, and testing	Information leakage
Computational transparency	Hardware, training time, inference latency, solver settings	Incomparable execution results
Forwarding feasibility	LSP/SID constraints and policy validation	Unverified deployment compatibility
Artifact availability	Code, datasets, configurations, and results	Limited independent verification

Note: These criteria constitute a proposed methodological appraisal framework rather than completed quality assessments of the reviewed studies.

Critical Evaluation. Reproducibility and statistical validity represent complementary but distinct requirements. A study may provide complete experimental artifacts while reporting insufficient statistical evidence, whereas statistically convincing findings may remain difficult to reproduce because of unavailable data or configurations.

Cross-study comparisons introduce additional limitations because experimental results often differ in network topology, traffic intensity, optimization objectives, simulation platforms, and baseline algorithms. Consequently, reported improvements should not be combined into quantitative rankings unless experimental conditions and outcome definitions are sufficiently comparable. Where quantitative aggregation is inappropriate, structured narrative synthesis should distinguish learning paradigms, forwarding architectures, experimental quality, and reported performance.

The systematic review itself should follow transparent reporting procedures consistent with PRISMA 2020 [10], including documented search strategies, eligibility criteria, study selection, data extraction, and methodological-quality assessment. Supplementary materials should provide study-selection records, extracted evidence, appraisal results, and analysis procedures where feasible.

Overall, reproducibility and statistical validity are indispensable for establishing reliable conclusions about AI-assisted MPLS and SR-MPLS traffic engineering. Future research should emphasize independent experimental repetitions, transparent learning configurations, appropriate uncertainty estimates, competitive baselines, accessible research artifacts, and architecture-specific forwarding validation. These practices are



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

essential for distinguishing robust optimization improvements from findings influenced by experimental randomness, insufficient reporting, or unequal evaluation conditions.

VI. RESEARCH GAPS AND OPEN CHALLENGES

Despite advances in Artificial Intelligence (AI)-assisted Traffic Engineering (TE), significant technical, methodological, and operational challenges remain in Multiprotocol Label Switching (MPLS) and Segment Routing over MPLS (SR-MPLS) networks. Although Machine Learning (ML), Reinforcement Learning (RL), Deep Reinforcement Learning (DRL), and hybrid optimization have demonstrated promising capabilities, their practical effectiveness depends on forwarding compatibility, computational efficiency, scalability, routing stability, and experimental validity. The critical analysis in Section V identifies nine interconnected research challenges involving optimization methodology, network architecture, and experimental evidence. These gaps represent areas requiring further investigation rather than conclusions about their prevalence across the complete literature.

A. Limited Architecture-Specific Validation

A major research challenge is the limited direct validation of intelligent optimization within MPLS and SR-MPLS forwarding architectures. Several approaches have been evaluated using abstract network models or generic Software-Defined Networking (SDN) environments without explicitly considering Label Switched Path (LSP) establishment, Segment Identifier (SID) availability, Maximum SID Depth (MSD), or SR Policy installation.

For example, GROM [49] and GRL-TE [43] demonstrate promising graph-based optimization capabilities in broader networking environments, while Adpt-SRTE [8] and RS4SR [42] provide Segment Routing-focused evidence. However, the precise forwarding implementations must be verified before these results can be classified as direct SR-MPLS validation.

Future research should evaluate whether AI-generated routing decisions satisfy architecture-specific forwarding constraints, control-plane requirements, and device capabilities. Conventional MPLS-TE, SR-MPLS, and Segment Routing over IPv6 (SRv6) should also be distinguished because their forwarding mechanisms and operational limitations differ.

B. Inconsistent Benchmarking Against Classical Optimization

Another challenge concerns comparisons between intelligent routing methods and conventional optimization baselines. Improvements over shortest-path routing or Equal-Cost Multipath (ECMP) do not necessarily establish superiority over advanced mathematical optimization or well-configured heuristics.

Bernárdez et al. [25] addressed this issue by comparing a Multi-Agent Reinforcement Learning (MARL) and Graph Neural Network (GNN)-based framework against the Declarative and Expressive Forwarding Optimizer (DEFO), demonstrating comparable congestion optimization with reduced execution time under the evaluated conditions.

Future evaluations should include competitive Constraint-Based Shortest Path First (CSPF), Linear Programming (LP), Mixed-Integer Linear Programming (MILP), constraint-programming, and heuristic baselines. Comparisons must use equivalent traffic demands, network topologies, optimization objectives, forwarding restrictions, and computational budgets. Solution-quality gaps should also be reported where suitable optimization bounds are available.

C. Scalability and Generalization Across Network Conditions

Scalability and generalization remain important limitations of learning-based traffic engineering. Increasing network size, traffic demands, and routing alternatives can substantially expand optimization complexity. Although DRL addresses some limitations of tabular RL, neural-network approximation does not eliminate the combinatorial nature of routing decisions.



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

Tian et al. [8] employed Proximal Policy Optimization (PPO) with action branching to manage large routing-action spaces, while Ding et al. [49] developed GROM to improve cross-topology generalization. Nevertheless, successful performance under selected conditions does not guarantee reliable operation across arbitrary MPLS or SR-MPLS networks.

Future studies should examine progressively larger topologies, unseen Traffic Matrices (TMs), traffic surges, and failure conditions. Evaluations must distinguish zero-shot generalization, transfer learning, fine-tuning, and complete retraining while assessing optimization quality, computational overhead, and forwarding feasibility.

D. Computational Overhead and End-to-End Operational Efficiency

Rapid inference does not necessarily imply lower total computational cost. Intelligent optimization introduces expenses associated with data preparation, offline training, hyperparameter tuning, retraining, inference, constraint validation, and policy deployment.

Adpt-SRTE [8] demonstrates fast online routing decisions after offline training, while RS4SR [42] accelerates Segment Routing optimization through historical-solution reuse. However, their computational benefits depend on model reuse, changing traffic conditions, and the complete optimization workflow.

A central research challenge is determining when amortized AI training costs become lower than repeated classical optimization. Future evaluations should therefore measure complete decision-to-deployment latency, including monitoring, inference, mathematical refinement, feasibility verification, and forwarding updates. Hardware-normalized comparisons are also necessary to establish meaningful computational advantages.

E. Routing Stability and Constraint-Aware Intelligent Optimization

Intelligent routing policies may produce frequent reconfigurations or infeasible forwarding decisions when optimization objectives do not adequately represent operational requirements. This challenge is particularly important for RL and DRL, where reward maximization does not inherently guarantee compliance with mandatory network constraints.

In SR-MPLS networks, routing decisions must satisfy SID availability, MSD limitations, and SR Policy requirements. RFC 9256 and RFC 8664 provide relevant policy-control and Segment Routing path-computation mechanisms, but they do not independently guarantee the feasibility of AI-generated decisions.

Potential solutions include constrained reinforcement learning, feasible-action masking, mathematical refinement, and independent path validation. Prediction-assisted multi-time-step optimization [26] also demonstrates opportunities to reduce unnecessary routing changes.

Future research should jointly evaluate congestion reduction, routing-change frequency, constraint violations, policy-installation latency, and forwarding stability. The principal challenge is maintaining adaptive routing responsiveness without introducing operational instability.

F. Incomplete Network-State Information and Traffic Uncertainty

Intelligent traffic engineering frequently depends on accurate and timely network-state information. However, operational networks may experience incomplete traffic measurements, delayed telemetry, uncertain demand estimates, and rapidly changing congestion conditions.

Adpt-SRTE [8] combines offline learning with online link-utilization measurements to reduce dependence on complete real-time traffic matrices. Similarly, Le et al. [26] integrate traffic prediction and compressive sensing to reduce monitoring overhead.

Nevertheless, inaccurate predictions or delayed observations may produce unsuitable routing configurations. Future investigations should examine uncertainty-aware prediction, partially observable decision-making, robust optimization, and adaptive monitoring. Experimental evaluations should include missing observations, measurement errors, telemetry delays, and unexpected traffic changes to determine whether intelligent routing remains reliable under incomplete information.



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

G. Fault Tolerance and Resilience Under Unexpected Failures

The resilience of intelligent traffic engineering under unexpected failures remains insufficiently established by the studies discussed in this review. Learning models optimized for normal conditions may perform poorly during link failures, node outages, or simultaneous network disruptions.

Conventional MPLS and Segment Routing architectures provide established protection mechanisms, including RSVP-TE Fast Reroute and Topology Independent Loop-Free Alternate (TI-LFA). AI-based approaches can potentially complement these mechanisms through post-failure path selection and resource reallocation.

A promising research direction is combining rapid local protection with intelligent reoptimization. However, such integration requires complete evaluation of failure detection, protection switching, updated network-state acquisition, routing computation, feasibility validation, and policy deployment.

Future experiments should include single-link, multiple-link, node, and Shared Risk Link Group (SRLG) failures. Relevant measurements include restoration time, transient packet loss, throughput degradation, post-failure link utilization, and the availability of feasible alternative paths.

H. Multiobjective Optimization, Coordination, and Interpretability

Intelligent traffic engineering must balance multiple objectives, including link utilization, throughput, delay, packet loss, routing stability, computational efficiency, and resilience. Improving one metric may degrade another; for example, congestion reduction through longer paths may increase end-to-end delay.

Fixed weighted reward functions may inadequately represent changing service priorities or mandatory operational restrictions. Consequently, intelligent optimization should distinguish hard forwarding constraints from performance objectives that permit trade-offs.

MARL introduces further challenges involving conflicting local decisions, inter-agent communication, and coordination overhead. Hierarchical control and centralized training with decentralized execution offer potential solutions, although network-wide stability requires additional validation.

Interpretability is also important because network operators must understand the reasons for routing changes and their potential operational consequences.

Future research should prioritize constraint-aware multiobjective optimization, adaptive reward design, coordinated multi-agent control, and explainable routing policies that communicate expected benefits, uncertainty, and deployment risks.

I. Experimental Realism, Reproducibility, and Statistical Evidence

Experimental heterogeneity remains a major obstacle to reliable comparisons of intelligent traffic engineering methods. Existing studies employ different simulation platforms, network topologies, traffic matrices, optimization objectives, and baseline algorithms, limiting the comparability of reported improvements.

Mathematical simulations provide useful evidence of optimization quality but may omit packet-level dynamics and forwarding constraints. Network emulation and hardware testbeds offer stronger architectural validation, although their capabilities and scalability vary.

Statistical reliability presents an additional challenge because RL and DRL results can depend on initialization, exploration, hyperparameter selection, and simulation randomness. Experiments based on limited independent repetitions may therefore overstate algorithmic advantages.

Future research should develop standardized, architecture-aware benchmarking frameworks incorporating representative traffic scenarios, competitive classical baselines, explicit forwarding constraints, and consistent performance metrics. Reproducibility should be strengthened through open datasets, documented configurations, source code, independent repetitions, uncertainty estimates, and statistical validation.

Consistent with PRISMA 2020 [10], the systematic review should also transparently document study selection, evidence extraction, methodological appraisal, and synthesis limitations.



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

J. Comparative Synthesis of Research Gaps

Table 12 consolidates the principal research challenges and their recommended research directions, mapped to the research questions.

Table 12. Research Gaps and Open Challenges in AI-Assisted MPLS and SR-MPLS Traffic Engineering

Research gap	Core challenge	Recommended research focus	Related RQs
Architecture-specific validation	Limited evidence of deployable MPLS/SR-MPLS routing	Protocol and forwarding validation	RQ2, RQ5
Benchmark fairness	Uncertain superiority over strong classical methods	Standardized comparisons with CSPF, LP/MILP, and heuristics	RQ3, RQ6
Scalability and generalization	Growing routing complexity and unseen conditions	Scalable representations and cross-topology evaluation	RQ4, RQ7
Computational overhead	Uncertain total cost beyond inference	End-to-end computational and deployment evaluation	RQ4, RQ5
Routing stability and constraints	Infeasible decisions and excessive reconfiguration	Safe learning and constraint-aware optimization	RQ5
Incomplete network information	Delayed and uncertain traffic measurements	Robust learning and uncertainty-aware TE	RQ1, RQ4
Fault tolerance and resilience	Limited evidence of complete failure recovery	Fast protection with intelligent reoptimization	RQ5, RQ7
Multiobjective and coordinated optimization	Conflicting QoS objectives and agent decisions	Multiobjective, coordinated, explainable optimization	RQ1, RQ4, RQ5
Experimental rigor	Heterogeneous and insufficiently reproducible evaluations	Standardized benchmarks, statistical validation, and open artifacts	RQ6, RQ7

Note: The table presents a preliminary analytical classification derived from the reviewed literature. It does not represent a completed frequency analysis or formal quality assessment of all eligible studies.

K. Overall Synthesis

The identified research gaps demonstrate that reliable AI-assisted MPLS and SR-MPLS traffic engineering requires advances in optimization methodology, forwarding architecture, and experimental validation. Improvements in routing efficiency or inference speed provide limited operational value unless the resulting decisions satisfy forwarding constraints, maintain stability, and demonstrate reliable performance under realistic network conditions.

Promising directions include GNN-based generalization, hybrid AI–mathematical optimization, uncertainty-aware routing, constraint-aware learning, and integration with established fault-protection mechanisms. However, their relative benefits must be demonstrated through competitive baselines, architecture-specific experiments, and reproducible statistical evaluation.

The central unresolved question is under which network conditions and operational constraints AI-assisted traffic engineering provides measurable advantages over established optimization methods while maintaining routing feasibility, stability, and service reliability. Addressing this question requires integrated evaluation frameworks connecting intelligent algorithms with programmable network control, constrained path computation, and realistic forwarding behavior.

VII. FUTURE RESEARCH DIRECTIONS

Advancing Artificial Intelligence (AI)-assisted Traffic Engineering (TE) in Multiprotocol Label Switching (MPLS) and Segment Routing over MPLS (SR-MPLS) networks requires addressing challenges involving computational efficiency, scalability, forwarding feasibility, routing stability, and experimental validation. Although Machine Learning (ML), Reinforcement Learning (RL), and Deep Reinforcement Learning (DRL) offer promising capabilities for traffic prediction and adaptive routing, their operational effectiveness remains dependent on network architecture and deployment constraints. Building on the research gaps identified in



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

Section VI, this section outlines priority directions for developing scalable, reliable, and practically deployable intelligent TE systems.

A. Architecture-Aware and Deployment-Oriented Intelligent Traffic Engineering

Future intelligent TE systems should explicitly incorporate MPLS and SR-MPLS forwarding requirements into their optimization processes. Existing studies frequently employ abstract routing models that may overlook Label Switched Path (LSP) establishment, Segment Identifier (SID) availability, Maximum SID Depth (MSD), and policy-installation constraints.

Architecture-aware learning should therefore consider bandwidth availability, signaling requirements, feasible segment lists, and forwarding-device capabilities. Integrating AI with Software-Defined Networking (SDN) controllers and Path Computation Elements (PCEs) offers a practical mechanism for implementing validated routing decisions. RFC 9256 and RFC 8664 provide relevant foundations for SR Policy management and Segment Routing path computation.

Future evaluations should examine complete decision-to-deployment workflows, including path computation, constraint validation, configuration installation, and forwarding verification. The primary objective is to establish whether algorithmic improvements remain effective under realistic MPLS and SR-MPLS operating conditions.

B. Constraint-Aware and Hybrid AI–Mathematical Optimization

Combining learning-based methods with mathematical optimization represents a promising direction for improving routing efficiency while maintaining operational feasibility. RL and DRL can identify adaptive routing decisions, whereas Linear Programming (LP), Mixed-Integer Linear Programming (MILP), and constraint-based algorithms provide explicit mechanisms for enforcing network requirements.

Hybrid frameworks may use ML for traffic prediction, RL or DRL for candidate-path selection, and mathematical programming for solution refinement and feasibility verification. Smart Node Selection (SNS) [36] and RS4SR [42] demonstrate opportunities to accelerate Segment Routing optimization through candidate reduction and historical-solution reuse.

Future research should prioritize safe and constrained RL, incorporating feasible-action masking, constraint-aware learning, and validated fallback mechanisms. Evaluations should measure solution quality, computational efficiency, constraint violations, routing stability, and complete decision-to-deployment latency against competitive classical baselines.

C. Scalable, Generalizable, and Transferable Learning Models

Scalability and generalization remain essential for deploying intelligent TE across different network sizes, traffic distributions, and topologies. Tabular RL suffers from rapidly expanding state–action spaces, while DRL may require extensive training and adaptation when network conditions change.

Graph Neural Networks (GNNs) offer promising capabilities for representing variable network structures and sharing learned information across routers, links, and paths. GROM [49] demonstrates the potential of GNN-based DRL for routing optimization on previously unseen topologies.

Transfer Learning (TL) and continual learning may further reduce retraining costs by reusing acquired knowledge across related network environments. However, differences in topology, traffic characteristics, SID assignments, and forwarding constraints can introduce negative transfer.

Future evaluations should distinguish zero-shot generalization, fine-tuning, and complete retraining while examining unseen traffic matrices, topology changes, network failures, and forwarding compatibility. Relevant measures include adaptation time, optimization quality, inference latency, computational overhead, and constraint satisfaction.



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

D. Uncertainty-Aware Prediction and Adaptive Network-State Estimation

Operational TE often relies on incomplete, delayed, or inaccurate network-state measurements. Traffic matrices may be unavailable in real time, while telemetry delays and unexpected demand fluctuations can reduce routing-decision accuracy.

Future research should investigate uncertainty-aware traffic prediction, robust optimization, and Partially Observable Markov Decision Process (POMDP) formulations. Rather than relying exclusively on predicted traffic values, intelligent controllers could incorporate confidence intervals, probability distributions, or historical observations into routing decisions.

Adaptive monitoring also offers opportunities to reduce telemetry overhead by adjusting measurement frequency according to congestion risk and network-state uncertainty.

Experiments should evaluate intelligent routing under missing measurements, delayed telemetry, prediction errors, and unexpected traffic changes. Performance should be assessed using congestion, throughput, delay, monitoring overhead, and routing stability.

E. Multi-Agent, Hierarchical, and Distributed Intelligent Optimization

Multi-Agent Reinforcement Learning (MARL) and hierarchical optimization offer opportunities to distribute routing decisions across network regions or administrative domains. Such approaches may reduce centralized computational requirements but introduce coordination, communication, and policy-consistency challenges.

Future research should investigate hierarchical architectures combining global traffic engineering objectives with local routing optimization. Centralized Training with Decentralized Execution (CTDE), cooperative rewards, parameter sharing, and graph-based communication may improve coordination among agents.

However, independently optimized routing decisions may create network-wide congestion or violate shared resource constraints. Accordingly, distributed frameworks require coordinated capacity allocation, feasibility verification, and consistent policy deployment.

Evaluations should compare centralized and distributed optimization under equivalent conditions, measuring Maximum Link Utilization (MLU), delay, convergence, communication overhead, computational efficiency, and routing stability.

F. Resilient and Stability-Aware Intelligent Traffic Engineering

Reliable intelligent TE must maintain acceptable performance during network failures and changing traffic conditions. Conventional protection mechanisms, including Resource Reservation Protocol–Traffic Engineering Fast Reroute (RSVP-TE FRR) and Topology Independent Loop-Free Alternate (TI-LFA), support rapid local recovery but may leave traffic distributed inefficiently after failures.

Future research should investigate hybrid resilience architectures combining fast local protection with AI-assisted post-failure reoptimization. Conventional mechanisms would maintain forwarding continuity, while intelligent controllers would subsequently improve traffic allocation across the remaining network resources.

Stability-aware optimization should incorporate routing-change penalties, hysteresis thresholds, minimum holding times, and controlled policy transitions to prevent excessive reconfiguration.

Experimental evaluation should include link failures, node outages, Shared Risk Link Group (SRLG) failures, and correlated disruptions. Key measures include service-restoration time, transient packet loss, post-failure MLU, routing-change frequency, and reoptimization latency.

G. Multiobjective, QoS-Aware, and Explainable Routing Decisions

Future intelligent TE should move beyond single-objective optimization to address competing Quality of Service (QoS) and operational requirements. Minimizing MLU does not necessarily improve end-to-end delay, throughput, packet loss, or service reliability.

Multiobjective optimization should therefore consider resource utilization, latency, reliability, computational efficiency, and routing stability. Promising approaches include Pareto-based optimization, adaptive objective



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

prioritization, and constrained learning that distinguishes mandatory service requirements from flexible performance preferences.

Explainable Artificial Intelligence (XAI) represents an additional opportunity for improving operational transparency. Network operators require interpretable explanations of routing decisions, relevant network constraints, and potential service risks.

Future studies should evaluate explanation fidelity, decision transparency, QoS satisfaction, routing stability, and operator usefulness. Explainability should complement, rather than replace, formal routing-feasibility validation.

H. Standardized Benchmarking, Reproducibility, and Realistic Deployment Validation

Standardized experimental frameworks are essential for establishing reliable comparisons among intelligent and classical TE methods. Existing studies use heterogeneous network topologies, traffic datasets, optimization objectives, and evaluation platforms, limiting the comparability of reported results.

Future benchmarks should incorporate representative MPLS and SR-MPLS topologies, realistic traffic matrices, congestion scenarios, network failures, and competitive classical baselines, including Constraint-Based Shortest Path First (CSPF), LP/MILP, and established heuristics.

Multi-level validation should combine mathematical simulation for optimization analysis, packet-level simulation for QoS evaluation, network emulation for forwarding verification, and physical testbeds for operational testing. SR-MPLS evaluations should explicitly address SID-stack processing, MSD limitations, and SR Policy installation.

Reproducibility requires transparent reporting of datasets, software versions, hyperparameters, random seeds, computational resources, and experimental configurations. Repeated trials and appropriate uncertainty estimates should support statistical comparisons.

Evaluation should also consider total computational costs, including offline training, online inference, validation, policy deployment, and retraining. Open, architecture-aware benchmarks would strengthen independent verification and improve the reliability of comparative conclusions.

I. Integrated Research Agenda and Evaluation Priorities

The proposed directions emphasize the need to integrate learning algorithms, mathematical optimization, programmable network control, and rigorous experimental evaluation. Progress should focus not only on improving optimization accuracy but also on ensuring that routing decisions are computationally efficient, operationally feasible, stable, and reproducible.

Table 13. Proposed Future Research Directions and Evaluation Priorities for AI-Assisted MPLS and SR-MPLS Traffic Engineering

Research direction	Proposed investigation	Principal evaluation measures	Related RQs
Architecture-aware intelligent TE	Integrate AI with MPLS/SR-MPLS forwarding mechanisms	Feasibility, constraint violations, installation latency	RQ2, RQ5
Safe hybrid optimization	Combine RL/DRL with mathematical constraint enforcement	MLU, optimality gap, validation overhead	RQ3, RQ5
Scalable and transferable learning	Investigate GNNs, TL, and adaptive representations	Generalization, adaptation time, scalability	RQ4, RQ7
Uncertainty-aware TE	Address incomplete telemetry and prediction uncertainty	QoS robustness, monitoring overhead	RQ1, RQ4
Hierarchical and multi-agent TE	Coordinate distributed routing optimization	Communication overhead, convergence, global performance	RQ2, RQ4
Resilient intelligent routing	Combine local protection with AI reoptimization	Recovery time, packet loss, post-failure utilization	RQ5, RQ7



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

Multiobjective and explainable TE	Balance QoS objectives with interpretable decisions	QoS satisfaction, stability, explanation fidelity	RQ3, RQ5
Reproducible and realistic evaluation	Develop common benchmarks and deployment validation	Statistical reliability, reproducibility, forwarding fidelity	RQ6, RQ7

Note: The table presents a proposed research agenda derived from the identified challenges. It does not represent experimentally validated recommendations or a quantitative ranking.

An integrated research approach should follow three complementary priorities: developing intelligent algorithms capable of adapting to changing network conditions; integrating these algorithms with architecture-aware control systems that enforce forwarding constraints and routing stability; and validating the resulting systems through reproducible experiments using competitive classical optimization baselines.

A promising direction is a closed-loop, constraint-aware intelligent TE framework in which network telemetry supports traffic prediction and learning-based optimization, candidate paths undergo independent feasibility validation, and approved configurations are deployed through programmable controllers or PCEs. Continuous monitoring would assess the effectiveness of installed policies, while fallback mechanisms would prevent invalid or unstable routing changes. This represents a proposed research framework rather than an implemented or empirically validated system.

Overall, future research should prioritize scalable, constraint-aware, resilient, and operationally feasible intelligent traffic engineering. Integrating the adaptability of ML, RL, and DRL with mathematical optimization and programmable MPLS/SR-MPLS control offers a promising direction. However, meaningful progress requires rigorous evaluation of optimization quality, generalization, computational overhead, forwarding feasibility, routing stability, resilience, and reproducibility. These advances are essential for establishing the conditions under which AI-based traffic engineering provides measurable advantages over established optimization methods.

VIII. LIMITATIONS OF THE REVIEW

This review provides a structured examination of Artificial Intelligence (AI)-assisted Traffic Engineering (TE) in Multiprotocol Label Switching (MPLS) and Segment Routing over MPLS (SR-MPLS) networks, focusing on Machine Learning (ML), Reinforcement Learning (RL), Deep Reinforcement Learning (DRL), and hybrid optimization. However, several methodological and technical limitations must be considered when interpreting the literature and its implications for practical network optimization.

First, literature-search coverage may limit the comprehensiveness of the review. The proposed search strategy covers IEEE Xplore, Scopus, Web of Science Core Collection, ACM Digital Library, and ScienceDirect for publications from 1 January 2015 to 18 October 2026. Although these databases provide substantial research coverage, relevant studies may appear in other repositories, technical reports, or industry publications. Differences in indexing and search terminology may also affect study identification. Furthermore, restricting the review period may exclude foundational contributions, which are retained as theoretical context rather than primary empirical evidence.

Second, eligibility criteria and architectural heterogeneity constrain the scope of evidence synthesis. The review prioritizes peer-reviewed journal articles and full conference papers addressing intelligent optimization in MPLS or relevant Segment Routing environments. Studies conducted exclusively in generic Internet Protocol (IP), Software-Defined Networking (SDN), or Segment Routing over IPv6 (SRv6) environments are treated as contextual evidence unless they satisfy the predefined architectural criteria. This distinction is necessary because forwarding mechanisms, Label Switched Path (LSP) requirements, Segment Identifier (SID) availability, Maximum SID Depth (MSD), and policy-installation procedures differ across architectures. Consequently, findings from abstract routing models cannot automatically be generalized to operational MPLS or SR-MPLS networks.



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

Third, experimental heterogeneity limits direct performance comparisons. The reviewed studies employ different network topologies, Traffic Matrices (TMs), link capacities, optimization objectives, forwarding constraints, and experimental platforms. Performance indicators, including Maximum Link Utilization (MLU), throughput, delay, packet loss, and computational time, may also be measured differently. Moreover, improvements relative to shortest-path routing do not necessarily establish superiority over competitive Linear Programming (LP), Mixed-Integer Linear Programming (MILP), constraint-programming, or heuristic methods. These differences restrict direct cross-study comparisons and require context-specific interpretation.

Fourth, quantitative synthesis may be limited by inconsistent reporting and experimental designs. The proposed review therefore emphasizes structured narrative synthesis informed by Synthesis Without Meta-analysis (SWiM) principles [12]. Formal meta-analysis would require sufficiently comparable outcome definitions, experimental conditions, baseline methods, and statistically independent observations. However, some studies report only relative improvements without underlying measurements, variability estimates, or sufficient methodological details. Consequently, qualitative synthesis may be more appropriate, although it limits the precision of conclusions concerning relative algorithmic effectiveness.

Fifth, methodological quality and reproducibility affect the reliability of reported findings. ML, RL, and DRL performance can be sensitive to random initialization, exploration strategies, hyperparameter settings, training duration, and simulation assumptions. Incomplete reporting of experimental configurations, independent repetitions, uncertainty estimates, or source code may prevent reliable verification. Furthermore, insufficient separation of training and testing data can introduce information leakage and overestimate model generalization. Although the proposed methodological appraisal framework addresses these concerns, it cannot compensate for missing information in primary publications.

Sixth, simulation-based evidence limits conclusions about operational deployment. Mathematical simulation and packet-level experimentation provide valuable insights into optimization quality and network performance but may omit controller latency, protocol signaling, forwarding-state synchronization, policy-installation overhead, and hardware limitations. Even network emulation may not fully reproduce production-network behavior. Therefore, improvements in MLU, throughput, or inference latency should not be interpreted as proof of deployment readiness without architecture-specific validation.

Seventh, scalability, generalization, and resilience findings remain dependent on experimental conditions. Performance on large networks does not necessarily establish generalization to unseen topologies or traffic distributions. Similarly, favorable optimization results following simulated failures do not independently demonstrate rapid service restoration or forwarding reliability. Differences in network size, training procedures, failure scenarios, and forwarding capabilities restrict comparisons among RL, DRL, multi-agent, and transfer-learning approaches.

Eighth, methodological classification introduces interpretive uncertainty. Some studies combine supervised prediction, unsupervised clustering, RL, DRL, and mathematical optimization within a single framework. Assigning these studies to one primary category may obscure the contributions of individual components. Although primary and secondary classifications can improve consistency, overlapping algorithmic characteristics and incomplete methodological descriptions may introduce subjectivity.

Ninth, publication and reporting biases may influence the available evidence. Studies reporting favorable optimization outcomes may be more likely to appear in the published literature, while unsuccessful experiments, constraint violations, and negative findings may be underreported. Selective reporting of performance indicators can further exaggerate perceived algorithmic benefits. Restricting the primary synthesis to peer-reviewed publications does not eliminate these potential biases.

Finally, the review's conclusions depend on completion of its systematic procedures. The proposed methodology follows PRISMA 2020 [10] and PRISMA-S [11] principles for literature searching, eligibility screening, data extraction, methodological appraisal, and evidence synthesis. However, the presentation of a



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

review protocol does not establish that all systematic procedures have been completed. Until database searches, duplicate removal, screening, full-text assessment, and study-level extraction are fully documented, conclusions based on selected publications should be regarded as preliminary. Claims concerning the prevalence of research gaps, methodological weaknesses, or comparative effectiveness must be supported by the completed evidence dataset.

Overall, these limitations emphasize the need for cautious interpretation of AI-assisted MPLS and SR-MPLS traffic engineering research. Although existing studies demonstrate promising optimization capabilities, conclusions regarding algorithmic superiority, scalability, resilience, and deployment feasibility remain dependent on architectural relevance, experimental rigor, and reproducibility. Future evidence syntheses should prioritize broader literature coverage, consistent classification criteria, standardized benchmarking, architecture-specific validation, and transparent statistical reporting. These improvements would strengthen the evidence needed to determine when intelligent optimization offers reliable advantages over established traffic engineering approaches.

CONCLUSION

The increasing complexity and variability of network traffic have intensified the need for adaptive Traffic Engineering (TE) in Multiprotocol Label Switching (MPLS) and Segment Routing over MPLS (SR-MPLS) networks. Artificial Intelligence (AI), particularly Machine Learning (ML), Reinforcement Learning (RL), and Deep Reinforcement Learning (DRL), offers promising capabilities for traffic prediction, routing optimization, congestion management, and resource allocation. This paper has developed a systematic review framework integrating intelligent optimization techniques, forwarding architectures, and experimental evaluation criteria to examine their potential benefits and operational limitations.

The analysis indicates that ML, RL, and DRL provide complementary capabilities rather than universally superior optimization solutions. Supervised learning supports traffic prediction and network-performance estimation, while unsupervised learning facilitates traffic characterization and anomaly identification. RL enables adaptive routing through interaction-based learning, whereas DRL extends these capabilities to more complex network states and routing decisions. However, their effectiveness depends on training quality, reward design, computational requirements, generalization, and compatibility with network forwarding mechanisms.

A central finding is that AI-based approaches must be evaluated against competitive classical optimization methods, including Constraint-Based Shortest Path First (CSPF), Linear Programming (LP), Mixed-Integer Linear Programming (MILP), and established heuristics. Although selected studies report improvements in Maximum Link Utilization (MLU), throughput, delay, and routing responsiveness, these outcomes depend on experimental conditions and baseline selection. Hybrid AI-mathematical optimization offers a promising alternative by combining learning-based adaptability with explicit constraint handling and solution validation.

The review further emphasizes that algorithmic improvements do not automatically establish operational feasibility. Intelligent routing decisions must satisfy Label Switched Path (LSP) requirements, Segment Identifier (SID) availability, Maximum SID Depth (MSD), SR Policy constraints, and forwarding-device capabilities. Software-Defined Networking (SDN) controllers and Path Computation Elements (PCEs) provide mechanisms for implementing intelligent path-control decisions, but practical effectiveness also depends on monitoring accuracy, deployment latency, routing stability, and fault tolerance.

Important research challenges remain concerning computational scalability, generalization across unseen network conditions, incomplete telemetry, training overhead, and resilience under network failures. Graph Neural Networks (GNNs), multi-agent learning, transfer learning, uncertainty-aware optimization, and constrained DRL represent promising directions. However, their effectiveness requires rigorous evaluation using realistic network environments, competitive optimization baselines, reproducible experiments, and appropriate statistical validation.



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

The present analysis is limited by the defined literature scope, architectural heterogeneity, and differences in experimental methodologies. Furthermore, the observations are based on selected studies and should be interpreted as a preliminary critical synthesis until the planned PRISMA 2020-informed search, screening, data extraction, and methodological appraisal procedures are completed and documented.

Overall, the most promising direction for intelligent MPLS and SR-MPLS traffic engineering is the integration of adaptive learning, mathematical optimization, and programmable network control rather than the replacement of classical methods. Future advances should demonstrate measurable improvements in network performance and computational efficiency while ensuring forwarding feasibility, routing stability, resilience, and reproducibility. By connecting algorithmic effectiveness with architectural and operational requirements, this review provides a foundation for developing reliable, scalable, and practically deployable AI-assisted traffic engineering systems.

REFERENCES

- [1]. A. Farrel, Ed., "Overview and Principles of Internet Traffic Engineering," RFC 9522, Internet Engineering Task Force, Jan. 2024. doi: 10.17487/RFC9522.
- [2]. E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture," RFC 3031, Internet Engineering Task Force, Jan. 2001. doi: 10.17487/RFC3031.
- [3]. D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels," RFC 3209, Internet Engineering Task Force, Dec. 2001. doi: 10.17487/RFC3209.
- [4]. C. Filsfils, Ed., S. Previdi, Ed., L. Ginsberg, B. Decraene, S. Litkowski, and R. Shakir, "Segment Routing Architecture," RFC 8402, Internet Engineering Task Force, Jul. 2018. doi: 10.17487/RFC8402.
- [5]. A. Bashandy, Ed., C. Filsfils, Ed., S. Previdi, B. Decraene, S. Litkowski, and R. Shakir, "Segment Routing with the MPLS Data Plane," RFC 8660, Internet Engineering Task Force, Dec. 2019. doi: 10.17487/RFC8660.
- [6]. C. Filsfils, K. Talaulikar, Ed., D. Voyer, A. Bogdanov, and P. Mattes, "Segment Routing Policy Architecture," RFC 9256, Internet Engineering Task Force, Jul. 2022. doi: 10.17487/RFC9256.
- [7]. Y. Xiao, J. Liu, J. Wu, and N. Ansari, "Leveraging Deep Reinforcement Learning for Traffic Engineering: A Survey," IEEE Communications Surveys & Tutorials, vol. 23, no. 4, pp. 2064–2097, 2021. doi: 10.1109/COMST.2021.3102580.
- [8]. Y. Tian, Z. Wang, X. Yin, X. Shi, J. Yang, and H. Zhang, "Adaptive traffic engineering with segment routing through deep reinforcement learning," Computer Networks, vol. 267, Art. no. 111356, 2025. doi: 10.1016/j.comnet.2025.111356.
- [9]. N. W. Hassan, M. I. Khalil, and H. M. Abbas, "AI meets SR: A survey on enhancing segment-routing performance with artificial intelligence," Telecommunication Systems, vol. 89, Art. no. 117, 2026. doi: 10.1007/s11235-026-01493-1.
- [10]. M. J. Page et al., "The PRISMA 2020 statement: An updated guideline for reporting systematic reviews," BMJ, vol. 372, Art. no. n71, 2021. doi: 10.1136/bmj.n71.
- [11]. M. L. Rethlefsen et al., "PRISMA-S: An extension to the PRISMA statement for reporting literature searches in systematic reviews," Systematic Reviews, vol. 10, Art. no. 39, 2021. doi: 10.1186/s13643-020-01542-z.
- [12]. M. Campbell et al., "Synthesis without meta-analysis (SWiM) in systematic reviews: Reporting guideline," BMJ, vol. 368, Art. no. 16890, 2020. doi: 10.1136/bmj.16890.
- [13]. B. Fortz and M. Thorup, "Internet traffic engineering by optimizing OSPF weights," in Proc. IEEE INFOCOM 2000, 2000, pp. 519–528. doi: 10.1109/INFCOM.2000.832225.
- [14]. D. Katz, K. Kompella, and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2," RFC 3630, Internet Engineering Task Force, 2003. doi: 10.17487/RFC3630.
- [15]. Open Networking Foundation, "Software-Defined Networking: The New Norm for Networks," ONF White Paper, Apr. 13, 2012. [Online]. Available: <https://opennetworking.org/wp-content/uploads/2011/09/wp-sdn-newnorm.pdf>



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

- [16]. A. Farrel, J.-P. Vasseur, and J. Ash, "A Path Computation Element (PCE)-Based Architecture," RFC 4655, Internet Engineering Task Force, Aug. 2006. doi: 10.17487/RFC4655.
- [17]. J.-P. Vasseur, Ed., and J.-L. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)," RFC 5440, Internet Engineering Task Force, Mar. 2009. doi: 10.17487/RFC5440.
- [18]. E. Crabbe, I. Minei, J. Medved, and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE," RFC 8231, Internet Engineering Task Force, Sep. 2017. doi: 10.17487/RFC8231.
- [19]. E. Crabbe, I. Minei, S. Sivabalan, and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for PCE-Initiated LSP Setup in a Stateful PCE Model," RFC 8281, Internet Engineering Task Force, Dec. 2017. doi: 10.17487/RFC8281.
- [20]. S. Sivabalan, C. Filsfils, J. Tantsura, W. Henderickx, and J. Hardwick, "Path Computation Element Communication Protocol (PCEP) Extensions for Segment Routing," RFC 8664, Internet Engineering Task Force, Dec. 2019. doi: 10.17487/RFC8664.
- [21]. K. Talaulikar, Ed., "Distribution of Link-State and Traffic Engineering Information Using BGP," RFC 9552, Internet Engineering Task Force, Dec. 2023. doi: 10.17487/RFC9552.
- [22]. R. Enns, M. Bjorklund, J. Schoenwaelder, and A. Bierman, Eds., "Network Configuration Protocol (NETCONF)," RFC 6241, Internet Engineering Task Force, Jun. 2011. doi: 10.17487/RFC6241.
- [23]. A. Bierman, M. Bjorklund, and K. Watsen, "RESTCONF Protocol," RFC 8040, Internet Engineering Task Force, Jan. 2017. doi: 10.17487/RFC8040.
- [24]. A. Valadarsky, M. Schapira, D. Shahaf, and A. Tamar, "Learning to route," in Proc. 16th ACM Workshop on Hot Topics in Networks (HotNets-XVI), 2017, pp. 185–191. doi: 10.1145/3152434.3152441.
- [25]. G. Bernárdez, J. Suárez-Varela, A. López, B. Wu, S. Xiao, X. Cheng, P. Barlet-Ros, and A. Cabellos-Aparicio, "Is machine learning ready for traffic engineering optimization?," in Proc. IEEE 29th Int. Conf. on Network Protocols (ICNP), 2021, pp. 1–11. doi: 10.1109/ICNP52444.2021.9651930.
- [26]. V. A. Le, Y. Ji, H. H. Tran, P. Le Nguyen, and J. C. S. Lui, "Achieving multi-time-step segment routing via traffic prediction and compressive sensing techniques," IEEE Transactions on Network and Service Management, vol. 21, no. 2, pp. 1534–1549, 2024. doi: 10.1109/TNSM.2023.3338622.
- [27]. K. Rusek, J. Suárez-Varela, P. Almasan, P. Barlet-Ros, and A. Cabellos-Aparicio, "RouteNet: Leveraging graph neural networks for network modeling and optimization in SDN," IEEE Journal on Selected Areas in Communications, vol. 38, no. 10, pp. 2260–2270, 2020. doi: 10.1109/JSAC.2020.3000405.
- [28]. M. I. Salman and B. Wang, "Near-optimal responsive traffic engineering in software defined networks based on deep learning," Future Generation Computer Systems, vol. 135, pp. 172–180, 2022. doi: 10.1016/j.future.2022.04.036.
- [29]. Y. Zuo, Y. Wu, G. Min, and L. Cui, "Learning-based network path planning for traffic engineering," Future Generation Computer Systems, vol. 92, pp. 59–67, 2019. doi: 10.1016/j.future.2018.09.043.
- [30]. Y. Guo, K. Huang, C. Hu, J. Yao, and S. Zhou, "Traffic engineering in dynamic hybrid segment routing networks," Computers, Materials & Continua, vol. 68, no. 1, pp. 655–670, 2021. doi: 10.32604/cmc.2021.016364.
- [31]. A. O. Salau and M. M. Beyene, "Software defined networking based network traffic classification using machine learning techniques," Scientific Reports, vol. 14, Art. no. 20060, 2024. doi: 10.1038/s41598-024-70983-6.
- [32]. L. F. Carvalho, S. Barbon Jr., L. S. Mendes, and M. L. Proença Jr., "Unsupervised learning clustering and self-organized agents applied to help network management," Expert Systems with Applications, vol. 54, pp. 29–47, 2016. doi: 10.1016/j.eswa.2016.01.032.
- [33]. J. Mazel, P. Casas, R. Fontugne, K. Fukuda, and P. Owezarski, "Hunting attacks in the dark: Clustering and correlation analysis for unsupervised anomaly detection," International Journal of Network Management, vol. 25, no. 5, pp. 283–305, 2015. doi: 10.1002/nem.1903.
- [34]. R. H. Hwang, M. C. Peng, C. W. Huang, P. C. Lin, and V. L. Nguyen, "An unsupervised deep learning model for early network traffic anomaly detection," IEEE Access, vol. 8, pp. 30387–30399, 2020. doi: 10.1109/ACCESS.2020.2973023.
- [35]. W. Wang, S. Saha, A. Haque, and G. Sidebottom, "RLSR: A reinforcement learning-based segment routing for QoS-aware software-defined network," IEEE Transactions on Machine Learning in Communications and Networking, vol. 4, pp. 813–829, 2026. doi: 10.1109/TMLCN.2026.3693163.



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

- [36]. L. Wang, L. Lu, M. Wang, Z. Li, H. Yang, S. Zhu, and Y. Zhang, "SNS: Smart node selection for scalable traffic engineering in segment routing networks," *IEEE Transactions on Network and Service Management*, vol. 22, no. 1, pp. 92–106, 2025. doi: 10.1109/TNSM.2024.3424928.
- [37]. C.-H. Ke, Y.-H. Tu, and Y.-W. Ma, "A reinforcement learning approach for widest path routing in software-defined networks," *ICT Express*, vol. 9, no. 5, pp. 882–889, 2023. doi: 10.1016/j.icte.2022.10.007.
- [38]. J. A. Boyan and M. L. Littman, "Packet routing in dynamically changing networks: A reinforcement learning approach," in *Advances in Neural Information Processing Systems*, vol. 6 (NIPS 1993), pp. 671–678. [Online]. Available: https://papers.nips.cc/paper_files/paper/1993/hash/4ea06fbc83cdd0a06020c35d50e1e89a-Abstract.html
- [39]. B. Chen, P. Sun, P. Zhang, J. Lan, Y. Bu, and J. Shen, "Traffic engineering based on deep reinforcement learning in hybrid IP/SR network," *China Communications*, vol. 18, no. 10, pp. 204–213, 2021. doi: 10.23919/JCC.2021.10.015.
- [40]. Y. Tian, Z. Wang, X. Yin, X. Shi, Y. Guo, H. Geng, and J. Yang, "Traffic engineering in partially deployed segment routing over IPv6 network with deep reinforcement learning," *IEEE/ACM Transactions on Networking*, vol. 28, no. 4, pp. 1573–1586, 2020. doi: 10.1109/TNET.2020.2987866.
- [41]. Z. Xu, J. Tang, J. Meng, W. Zhang, Y. Wang, C. H. Liu, and D. Yang, "Experience-driven networking: A deep reinforcement learning based approach," in *Proc. IEEE INFOCOM*, 2018, pp. 1871–1879. doi: 10.1109/INFOCOM.2018.8485853.
- [42]. L. Wang, M. Wang, C. Lin, and Y. Zhang, "Accelerating traffic engineering optimization for segment routing: A recommendation perspective," *Computer Networks*, vol. 264, Art. no. 111224, 2025. doi: 10.1016/j.comnet.2025.111224.
- [43]. J. Lu, C. Tang, W. Ma, and W. Xing, "Graph-based reinforcement learning for software-defined networking traffic engineering," *Journal of King Saud University - Computer and Information Sciences*, vol. 37, Art. no. 119, 2025. doi: 10.1007/s44443-025-00133-z.
- [44]. L. Wang, M. Wang, and Y. Zhang, "A safe training approach for deep reinforcement learning-based traffic engineering," in *Proc. IEEE Int. Conf. on Communications (ICC)*, 2022, pp. 1450–1455. doi: 10.1109/ICC45855.2022.9838944.
- [45]. Y. Guo, B. Lin, Q. Tang, Y. Ma, H. Luo, H. Tian, and K. Chen, "Distributed traffic engineering in hybrid software defined networks: A multi-agent reinforcement learning framework," *IEEE Transactions on Network and Service Management*, vol. 21, no. 6, pp. 6759–6769, 2024. doi: 10.1109/TNSM.2024.3454282.
- [46]. Z. Luan, Q. Li, Y. Jiang, J. Duan, R. Zheng, D. Chen, and S. Liu, "MATE: When multi-agent deep reinforcement learning meets traffic engineering in multi-domain networks," *Computer Networks*, vol. 247, Art. no. 110399, 2024. doi: 10.1016/j.comnet.2024.110399.
- [47]. J. Cao, Z. Wu, Q. Tang, and X. Li, "Segment routing optimization algorithm fusing deep reinforcement learning and load centrality theory," *Journal of National University of Defense Technology*, vol. 47, no. 6, pp. 46–59, 2025. doi: 10.11887/j.issn.1001-2486.25050027.
- [48]. M. K. Awad, M. K. Mohammed, and E. M. Alotaibi, "A knowledge transfer-enabled reinforcement learning-based approach for services routing in software-defined networks," *IEEE Transactions on Network and Service Management*, vol. 23, pp. 5258–5279, 2026. doi: 10.1109/TNSM.2026.3702330.
- [49]. M. Ding, Y. Guo, Z. Huang, B. Lin, and H. Luo, "GROM: A generalized routing optimization method with graph neural network and deep reinforcement learning," *Journal of Network and Computer Applications*, vol. 229, Art. no. 103927, 2024. doi: 10.1016/j.jnca.2024.103927.
- [50]. R. Hartert, S. Vissicchio, P. Schaus, O. Bonaventure, C. Filsfil, T. Telkamp, and P. François, "A declarative and expressive approach to control forwarding paths in carrier-grade networks," in *Proc. ACM SIGCOMM*, 2015, pp. 15–28. doi: 10.1145/2785956.2787495.
- [51]. E. Moreno, A. Beghelli, and F. Cugini, "Traffic engineering in segment routing networks," *Computer Networks*, vol. 114, pp. 23–31, 2017. doi: 10.1016/j.comnet.2017.01.006.
- [52]. Y. Ma, Y. Guo, R. Yang, and H. Luo, "FRRL: A reinforcement learning approach for link failure recovery in a hybrid SDN," *Journal of Network and Computer Applications*, vol. 234, Art. no. 104054, 2025. doi: 10.1016/j.jnca.2024.104054.
- [53]. M. Farhan, N. Shah, L. Wang, G.-M. Muntean, and H. H. Song, "RDG-TE: Link reliability-aware DRL-GNN-based traffic engineering in SDN," *Expert Systems with Applications*, vol. 265, Art. no. 125963, 2025. doi: 10.1016/j.eswa.2024.125963.



International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

Mustafa ElGili Mustafa, Volume 15, Issue 10, October 2026, pg. 20-73

(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)

ISSN: 2320-088X

Impact Factor: 7.056

- [54]. S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The Internet Topology Zoo," IEEE Journal on Selected Areas in Communications, vol. 29, no. 9, pp. 1765–1775, 2011. doi: 10.1109/JSAC.2011.111002.
- [55]. B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: Rapid prototyping for software-defined networks," in Proc. 9th ACM Workshop on Hot Topics in Networks (HotNets-IX), 2010, Art. no. 19. doi: 10.1145/1868447.1868466.
- [56]. R. Agarwal, M. Schwarzer, P. S. Castro, A. C. Courville, and M. G. Bellemare, "Deep reinforcement learning at the edge of the statistical precipice," in Advances in Neural Information Processing Systems (NeurIPS), vol. 34, pp. 29304–29320, 2021. [Online]. Available: <https://proceedings.neurips.cc/paper/2021/hash/f514cec81cb148559cf475e7426eed5e-Abstract.html>
- [57]. A. Patterson, S. Neumann, M. White, and A. White, "Empirical design in reinforcement learning," Journal of Machine Learning Research, vol. 25, no. 318, pp. 1–63, 2024. [Online]. Available: <https://jmlr.org/papers/v25/23-0183.html>
- [58]. Association for Computing Machinery, "Artifact Review and Badging," ver. 1.1, Aug. 24, 2020. [Online]. Available: <https://www.acm.org/publications/policies/artifact-review-and-badging-current>