Vietnam Journal of Science and Technology ***63*** *(3) (2025) 576-593*

doi:10.15625/2525-2518/19667

A hybrid multi-constraint lagrangian relaxation based aggregated cost based segment routing in qos aware software defined networks

Kumar Parop Gopal*, M. Sambath

Department of Computer Science and Engineering, Hindustan Institute of Technology and Science, Chennai, Padur, 603 103, India

*Emails: *kpgopal2007@gmail.com, msambath@hindustanuniv.ac.in*

Received: 15 December 2023; Accepted for publication: 4 June 2024

Abstract. The Industrial Internet is a cornerstone of Industry 4.0, emphasizing the need for high reliability and low latency in network control. It enables the integration of diverse IoT sensing and actuation devices for efficient monitoring and management of industrial systems. Software-Defined Networking (SDN), a modern network architecture, offers centralized control by logically separating the control and data planes. This global view facilitates flexible network administration, optimized performance, and enhanced adaptability. However, meeting the increasing demands for efficient and dependable network services in SDNs has made Quality of Service (QoS) assurance a critical challenge. This study introduces a novel approach Hybrid Multi-constraint Lagrangian Relaxation-based Aggregated Cost (HMLR-AC) Segment Routing for QoS-aware routing in SDNs. HMLR-AC combines the strengths of Lagrangian Relaxation (LR) and Aggregated Cost (AC) to enhance routing decisions. Additionally, it incorporates an H-permissible Paths Routing Scheme (HPRS) that effectively routes traffic within cardinality constraints. Utilizing real-time traffic insights and a global network perspective, HMLR-AC dynamically adapts routing strategies. Simulation results demonstrate notable improvements in QoS parameters, proving HMLR-AC as a promising solution for future QoS-driven SDN infrastructures.

Keywords: software defined networking, segment routing, quality of service, H-permissible paths routing scheme, hybrid multi-constraint lagrangian relaxation based aggregated cost.

Classification numbers: 4.6.1, 4.6.4, 4.7.2, 4.9.4.

1. INTRODUCTION

The software-defined network (SDN) has quickly risen to the position of leading networking architecture due to its ability to improve network management while also facilitating new forms of network communication [1]. Division of the control plane from the data plane is a basic feature of SDN design. A logically centralized control plane directs the data plane, therefore maintaining the network state as well. Following control rules, devices forward data

packets in the data plane network [2]. Still, the academic and network sectors have focused especially on this architectural shift. From Google B4, NTT's edge gateways, NTT's community cloud, and Microsoft's public cloud to other contexts, SDNs have shown advantages.

In order to overcome the flexibility and programmability problems in conventional networks, furthermore offering a defined or consistent application programming interface (API), SDN helps the network to include current programme able capabilities [3]. Furthermore, SDNs enable network service providers create a more adaptable, under control, programmable network architecture. These SDN properties enable the network to dynamically modify its operations, therefore enabling the control plane and a global viewpoint of the system architecture. Although SDNs oversee systems in a more Centralized manner, among academics and business professionals they are becoming more and more appealing [4]. This is so because, in all spheres, security now takes front stage, particularly in more modern network systems like cloud and peer-to-peer networks. Though their many advantages, scalability, reliability, controller placement, and latency are only a few of the basic network security concerns that SDNs provide. Other academics also investigated other security flaws and tested them across several network systems [5]. Currently one of the main concerns is the higher danger of security breaches on SDN levels. Since there are no best practices for SND operations, components, and open program ability of networks [6], security concerns include consistency of flow rules, controller vulnerability, legitimacy, malicious applications, Standardized and northbound and southbound communications have higher chance. From its multi-layered architecture, the literature amply illustrates how different SDN levels create different security problems.

1.1. Research of our work

We present a new Hybrid Multi-constraint Lagrangian Relaxation-based Aggregated Cost (HMLR-AC) segment routing system for QoS-aware Software-Defined Networks (SDNs). Using a Lagrangian relaxation technique to maximise path selection, the system combines several QoS requirements like bandwidth, latency, jitter, and packet loss into a single routing architecture. HMLR-AC increases general network performance by dynamically balancing many performance criteria and using segment routing, hence lowering computational complexity. Experimental findings show that the suggested method maintains QoS guarantees over several network topologies and achieves almost optimum routing selections.

1.2. Motivation of this research

This work is driven in response to the growing demand for smart and efficient traffic engineering in QoS-aware Software-Defined Networks (SDNs). For traditional routing systems, constraints on several QoS including latency, bandwidth, and reliability all around are challenging. By balancing several network constraints, the proposed hybrid Multi-constraint Lagrangian Relaxation based Aggregated Cost (HMLR-AC) segment routing system aims to optimize path choice, thereby overcoming this. Combining segment routing with a scalable and reasonably cost optimization framework assures QoS compliance and improves overall network performance in dynamic and complex SDN environments.

1.3. The main contribution of this research

- To develop introduces a unique technique, Hybrid Multi-constraint Lagrangian Relaxation-based Aggregated Cost (HMLR-AC) Segment Routing, to handle QoS-aware routing in SDN.

- To optimise routing decisions, HMLR-AC integrates Lagrangian Relaxation (LR) and Aggregated Cost (AC) techniques. In addition, an H-permissible Paths Routing Scheme (HPRS) is developed for effective path cardinality management.
- Finally, the results show significant improvements over previous methods in terms of QoS satisfaction, scalability, and cost efficiency. HMLR-AC significantly lowers congestion, increases performance, and enhances the user experience.

1.4. Paper organization

The work is arranged as follows. The background and inspiration for QoS-aware SDN as well as the suggested segment routing system based on HMLR-AC are introduced in Section 2. Section 2 covers related activity. The hybrid multi-constraint Lagrangian relaxation (HMLR-AC) approach is presented in Section 3. Performance assessments and experimental findings are found in Section 4 and Section 5 concludes the work and provides future directions.

2. LITERATURE SURVEY

Goswami, *et al*. [7] proposed software-defined real-time networks (SDRTN) to address the two main needs of real-time networks in industrial systems: (a) better resource utilisation for the implementation of quality of service (QoS) regulations and (b) enhanced congestion management for controlled timely delivery. Three main conclusions of this chapter are as follows: A theoretical model combining SDN with RTN is developed in order to offer a step-by-step implementation process for SDRTN; the SDRTN scheme is designed for the policy implementation in RTN; and performance evaluation is demonstrated based on throughput, packet loss, latency, and jitter metrics. Presenting illustrated numerical studies, we show how the SDRTN design influences the performance of RTN-based networks.

David, *et al*. [8] with the advent of virtualisation and cloud computing, software defined networking (SDN) has emerged as a viable solution to support networking in the future. By simplifying software, hardware, and administration techniques, the use of SDN principles across many organisations delivers benefits that save operating costs. This paper examines some of the SDN solutions that have been created recently for datacenters, wide area networks (WANs), and campus networks that were discovered by the Gartner Peer Insights assessments. Additionally, a survey was carried out to see how these ideas have been modified in other organisations.

Urrea, *et al*. [9] in the proposal, six controllers that represent various control plane kinds are identified, and 10 criteria are established for choosing the best controller using the Analytic Hierarchy Process (AHP) technique. In order to provide a comprehensive solution that is not confined to the features of a particular scenario, this article analyses and proposes several essential principles for the deployment of SDNs in IIoT. As a result, it may be applied in a limited number of cases.

Karmous, *et al*. [10] provides an improved Intrusion and Prevention Detection System (IDPS) framework (SDN-ML-IoT) that might help with real-time mitigation and more accurate DDoS attack detection. Using ML on an SDN environment, our SDN-ML-IoT protects smart home IoT devices against DDoS assaults. Our ML technique follows a One-versus-Rest (OvR) strategy based on Random Forest (RF), Logistic Regression (LR), k-Nearest Neighbours (kNN), and Naive Bayes (NB). We then evaluated our efforts in line with other pertinent studies. With performance metrics including accuracy, training time, prediction time, confusion matrix, and Area Under the Receiver Operating Characteristic curve (AUC-ROC), SDN-ML-IoT beats current ML algorithms and related techniques in RF.

Akinsolu, *et al*. [11] based on the literature, these are the most important measurements for SDN operations. When the SDN is subjected to a distributed denial-of- service (DDoS) attack; and normal operating conditions with no SDN incidents; they have been behaviourally analysed in the following typical SDN states: flooding of the user datagram protocol (UDP), transmission control protocol (TCP), and hypertext transfer protocol (HTTP); Univariate and multivariate exploratory data analysis (EDA) is the main method used in behavioural research to explain and demonstrate the fluctuations of the SDN parameters for each one of the repeated conditions. Moreover, inferences on the sensitivity of the SDN parameters to the simulated situations based on linear regression are obtained.

Liu, *et al*. [12] provides a feature engineering and machine learning-based approach for DDoS attack detection in SDNs. Improved binary grey wolf optimisation approach was shown to offer the optimal feature subset after CSE-CIC-IDS2018 data cleaning and normalisation. Training and evaluation of the ideal feature subset following the Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbour (k-NN), Decision Tree, and XGBoost machine learning algorithms selected the best classifier for DDoS attack detection and used in the SDN controller. Analysed over several performance parameters including accuracy, precision, recall, F1, and AUC values, the results suggest that RF performs really well.

Khedr, *et al.* [13] advises the four-modules, SDN-based FMDADM solution for Internet of Things network DoS attack detection and prevention. The proposed FMDADM structure has five tiers and four key components. First module uses a 32-packet window size and anchored on the average drop rate (ADR) creates an early detection method. The second module provides early data plane assault detection using a unique double-check mapping tool known as DCMF. Comprising four processes data preparation, feature extraction, training and testing, and classification the third module, an ML-based detecting software, this solution identifies DDoS assaults using just seven features two chosen and five newly created.

2.1. Research Gap

Research gaps still exist even with major developments in Software-Defined Networks (SDNs). In dynamic and large-scale applications, current SDN designs can have difficulty with security, scalability, and effective resource management. Furthermore, intelligent decision-making systems like artificial intelligence and machine learning are not very often used to improve real-time network adaptation and anomaly detection. Furthermore, challenging smooth deployment over diverse networks is compatibility between several SDN controllers and traditional systems. Complete realisation of the possibilities of SDNs in contemporary network architectures depends on addressing these shortcomings.

2.2. Problem Identification of Existing Systems

- Centralized SDN controllers can become a performance bottleneck in large-scale networks due to increased control overhead and limited processing capabilities.
- The centralized nature of SDN controllers introduces vulnerability failure of the controller can disrupt the entire network's operation.
- Real-time decision-making and flow installations can cause latency, particularly in networks requiring high throughput and low delay, such as IoT and 5G applications.

2.3. **Limitations of the existing systems**

- Segment routing (SR) in QoS-aware SDNs may run into scalability problems when the network gets more general or there are more network nodes. The expense of managing

and disseminating segment routing information across the network may become prominent as the network size grows, affecting performance.

- Since segment routing is a new technology, interoperability among vendor applications may need more than unified. Ensuring consistent behavior and compatibility across different SR applications in a multi-vendor system might be challenging.
- To allocate segment routing information, segment routing relies on underlying protocols. Though, not all routing protocols permit segment routing, which limits utilization options and interoperability with existing network equipment.
- QoS-aware SDNs must exactly describe and measure QoS metrics to create optimal routing decisions. Nevertheless, standardized QoS measurements for numerous network services and applications still need to be enhanced. This can complicate important QoS policies and accomplishing consistent QoS across the network.

3. PROPOSED SYSTEM

This research offers a fresh approach to handle QoS-aware routing in SDN: HMLR-AC Segment Routing. To optimise routing decisions, HMLR-AC combines LR and AC methods. Furthermore, developed is an HPRS for effective path cardinality management. While lowering overall network costs, the approach aims to meet many QoS criteria including bandwidth, latency, and dependability. H-permissible pathways ensure that specific QoS requirements are met, therefore providing better service quality. The approach dynamically changes routing depending on real-time traffic data using the centralised control plane and SDN programme ability. This flexibility promotes wise use of resources as well as even traffic flow. Simulations using conventional network topologies validate the method. In Figure 1 shows the block diagram of the proposed model.

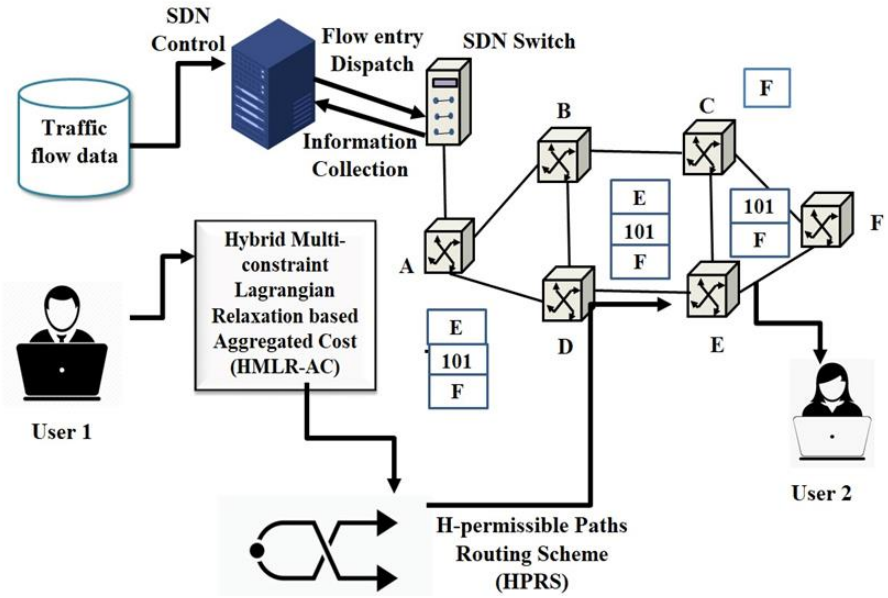

Figure 1. Proposed method of HMLR-AC.

The diagram depicts the suggested framework for optimizing QoS-aware traffic management by combining SDN control with Hybrid Multi-constraint Lagrangian Relaxation-based Aggregated Cost (HMLR-AC) Segment Routing and H-permissible Paths Routing Scheme (HPRS). Initially, the SDN controller collects and analyzes traffic flow statistics, as well as flow entries and network state information. The HMLR-AC technique is then used to determine optimal routing paths that satisfy multiple QoS constraints, including delay, bandwidth, and dependability. The HPRS module improves path selection by restricting the number of hops based on H-permissibility, increasing scalability while decreasing computation overhead. These best pathways are encoded as segment identifiers (SIDs) and routed to SDN switches for processing. Segment-based forwarding routes data packets from User 1 to User 2 via nodes A → D → E → F. Throughout the process, real-time monitoring ensures dynamic response to changing network conditions, allowing for efficient, flexible, and dependable communication across the SDN ecosystem. This architecture successfully balances network cost, performance, and Quality of Service.

3.1. QoS-Aware Routing Optimization with Lagrangian Relaxation

In hybrid SDN environments, QoS-aware routing must consider multiple constraints such as bandwidth availability, end-to-end delay, and link reliability while minimizing overall network costs. Traditional shortest path algorithms fall short in such multi-constrained settings due to their complexity and lack of flexibility. The proposed model formulates the routing as a **multi-constraint optimization problem**, where the objective is to minimize the total cost of a selected path while satisfying QoS constraints. This can be mathematically expressed as:

$$\boldsymbol{Minimize} \sum_{(i,j)\in E} \boldsymbol{c_{ij}}.\boldsymbol{x_{ij}} \tag{1}$$

$$\sum_{(i,j)\in E} \boldsymbol{d_{ij}}.\boldsymbol{x_{ij}} \leq \boldsymbol{D_{max}} \tag{2}$$

$$\sum_{(i,j)\in E} \frac{\boldsymbol{1}}{\boldsymbol{b_{ij}}}.\boldsymbol{x_{ij}} \leq \boldsymbol{B_{inv}} \tag{3}$$

$$\sum_{(i,j)\in E} \boldsymbol{l_{ij}}.\boldsymbol{x_{ij}} \leq \boldsymbol{L_{max}} \tag{4}$$

Where $\boldsymbol{c_{ij}}$ is the cost, $\boldsymbol{d_{ij}}$ is the delay, $\boldsymbol{b_{ij}}$ is the bandwidth, and $\boldsymbol{l_{ij}}$ is the packet loss rate of link $(\boldsymbol{i},\boldsymbol{j})$; $\boldsymbol{x_{ij}} \in \{\boldsymbol{0},\boldsymbol{1}\}$ is a binary variable indicating whether the link is part of the route. To simplify this NP-hard problem, **Lagrangian Relaxation** transforms constraints into penalty terms in the objective function. This results in a relaxed cost function:

$$\boldsymbol{L(x,\lambda)} = \sum_{(i,j)\in E} \boldsymbol{c_{ij}}.\boldsymbol{x_{ij}} + \boldsymbol{\lambda_1}\left(\sum \boldsymbol{d_{ij}}\boldsymbol{x_{ij}} - \boldsymbol{D_{max}}\right) + \boldsymbol{\lambda_2}\left(\sum \frac{\boldsymbol{1}}{\boldsymbol{b_{ij}}}\boldsymbol{x_{ij}} - \boldsymbol{B_{inv}}\right) \tag{5}$$

Here, $\boldsymbol{\lambda_1}$ and $\boldsymbol{\lambda_2}$ are **Varangian multipliers** dynamically adjusted during iterations to guide the algorithm toward constraint satisfaction. By means of effective iterative solvers such as sub gradient or dual ascent techniques, this transformation enables the discovery of near-optimal pathways balancing network cost with QoS guarantees. Real-time traffic data will help SDN controllers to continually improve path choice, hence enhancing performance and reducing congestion in hybrid systems.

3.2. H-permissible Paths Routing Scheme (HPRS)

The H-permissible Paths Routing Scheme (HPRS) aims to address the scalability issue in QoS-aware routing. Building and assessing multiple alternative channels in a hybrid SDN system may result in unnecessarily complex solutions. By requiring a maximum hop count of $\mathbf{H_{max}}$ the HPRS architecture therefore tightly limits path cardinality and addresses problem. Roads exceeding this level are thrown away, thereby simplifying the options for possible route. This reduces computing cost and guarantees sufficient variation of paths to meet the required QoS criteria. One may show the cardinality condition as follows:

$$\boldsymbol{h(p) \leq H_{max}\ \forall_p \in P} \tag{6}$$

where $\mathbf{h(p)}$ is the number of hops in path $\mathbf{p}$, and $\mathbf{P}$ is the set of all potential paths. By restricting the search space to only H-permissible paths, HPRS not only accelerates path computation but also ensures that these paths can still satisfy the bandwidth, delay, and reliability requirements set forth in the routing problem. In addition to reducing the solution space, HPRS incorporates aggregated cost functions to rank and select the best H-permissible paths. Each path $\mathbf{p}$ is evaluated using a weighted composite cost:

$$\mathbf{C(p) = w_1 . delay(p) + w_2 . \frac{1}{bandwidth(p)} + w_3 . loss\_rate(p)} \tag{7}$$

The goal is to find paths with minimal $\mathbf{C(p)}$ while ensuring that all selected paths remain within the hop limit:

$$\mathbf{Minimize \sum_{p \in P_H} C(p)\ \ subject\ to\ \ h(p) \leq H_{max}} \tag{8}$$

here,$\mathbf{P_H \subseteq P}$ represents the subset of H-permissible paths. Finally, to ensure the chosen paths also satisfy the original QoS constraints, the scheme incorporates additional constraints for delay, bandwidth, and loss rate:

$$\mathbf{delay(p) \leq D_{max},\ \ bandwidth(p) \geq B_{min},\ \ loss\_rate(p) \leq L_{max}} \tag{9}$$

using these formulas helps HPRS strike a compromise between scalability and QoS satisfaction. The outcome is a computationally effective, practical routing method for hybrid SDN systems.

3.3. Dynamic Adaptation via SDN Controller

Dynamic Adaptation Made Possible by SDN Controller changes routing decisions depending on network situation by using centralised control powers of SDN. The routing of conventional networks depends on dispersed protocols or stationary settings with delayed reaction to changes. Conversely, the SDN controller maintains a bird's-eye view of the network's architecture traffic patterns, and connection statuses, therefore enabling real-time data collecting and dynamic modifications. By means of this centralisation, the controller may monitor important indicators such link utilisation $\mathbf{u_{ij}}$ available bandwidth $\mathbf{b_{ij}}$ and path delays $\mathbf{d_{ij}}$ and use them to always assess and modify routing strategies. Mathematically, the overall link utilisation may be stated as:

$$\mathbf{U = \sum_{(i,j) \in E} u_{ij}} \tag{10}$$

The controller dynamically computes the shortest or most cost-efficient path $\mathbf{p}$ for each flow by considering the aggregated link costs. If the cost of a path $\mathbf{p}$ is given by$\mathbf{C(p)}$, incorporating metrics such as latency, available bandwidth, and reliability, the controller seeks to minimize:

$$\mathbf{C(p)} = \sum_{(\mathbf{i},\mathbf{j})\in\mathbf{p}} \left(\boldsymbol{\alpha}.\mathbf{d_{ij}} + \boldsymbol{\beta}.\frac{\mathbf{1}}{\mathbf{b_{ij}}} + \boldsymbol{\gamma}.\mathbf{r_{ij}} \right) \quad (11)$$

where $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$ are weighting coefficients, and $\mathbf{r_{ij}}$ represents the link reliability. This equation enables the controller to select paths that balance multiple QoS criteria while maintaining efficiency. When network conditions change such as a link becoming congested or failing the controller can quickly recompute paths. For example, if the utilization $\mathbf{u_{ij}}$ of a link exceeds a threshold $\mathbf{U_{max}}$, the controller reroutes flows to underutilized links. This process can be expressed as:

$$\mathbf{If\ u_{ij} > U_{maz},\ recomput\ C(p)\ and\ select\ a\ new\ path\ p'} \quad (12)$$

The better route options assist to lower the overall traffic congestion by providing fair traffic distribution. Controlling changing traffic loads is also another advantage. Should traffic demand for a given source-destination pair rise, the controller finds unoccupied pathways meeting QoS criteria and adjusts the flow. The basis for this change is a fresh optimization:

$$\mathbf{Minimize\ \sum_{p\in P} C(p)\ \ subject\ to\ Qos\ constraints} \quad (13)$$

where **P** represents the spectrum of all the conceivable routes. Always gathering data over time, the controller modifies path weights and limits depending on real-time performance criteria. This iterative strategy guarantees that the network stays adaptable, therefore allowing its reaction to changing traffic patterns. These methods enable the dynamic adaptive architecture of the SDN controller to increase traffic allocation as well as QoS adherence. Re-optimizing pathways in real time under control from measurements including latency, bandwidth, and dependability generates from the controller a more stable, efficient, and user-friendly network environment. This flexibility gives end customers dependable service delivery and ongoing performance by making the model strong against network disruptions.

Algorithm. 1: HMLR_AC_Segment_Routing ()

Input:

G = (V, E) // Network graph with nodes V and links E

QoS_constraints = {Bandwidth, Delay, PacketLoss}

H_max // Maximum allowed hop count (H-permissible)

α, β, γ // Weight coefficients for cost, delay, and reliability

λ1, λ2 // Lagrangian multipliers

TrafficMatrix // Source-Destination flow demands

Output:

OptimalPathList // List of selected paths satisfying QoS

Begin

Initialize Controller with global network view

For each flow f in TrafficMatrix do

Generate candidate paths P_f from source to destination

Step 1: Apply H-permissible path constraint

H_permissible_Paths = []

For each path p in P_f do

```
        If hop_count(p) ≤ H_max then
          H_permissible_Paths.append(p)
        EndIf
      EndFor
  Step 2: Compute Aggregated Cost using Lagrangian Relaxation
      For each path p in H_permissible_Paths do
        cost_p = 0
        For each link e in p do
          c_e = α * cost(e) + β * delay(e) + γ * (1 - reliability(e))
          penalty = λ1 * max(0, bandwidth_required(f) - bandwidth(e)) +
                λ2 * max(0, delay(e) - delay_threshold)
          cost_p += c_e + penalty
        EndFor
        Assign total_cost[p] = cost_p
      EndFor
  Step 3: Select optimal path
      OptimalPath = path in H_permissible_Paths with minimum total_cost
      OptimalPathList.append(OptimalPath)
  Step 4: Install flow rules via SDN controller
      Controller.installFlowRules(OptimalPath)
    EndFor
  Step 5: Dynamic monitoring and adaptation
    While network is active do
      Monitor real-time metrics: link_utilization, delay, bandwidth, loss
      For each installed path do
        If QoS violation or congestion detected then
          Recompute OptimalPath using updated metrics
          Update flow rules via Controller
        EndIf
      EndFor
    EndWhile
  End
```

More flexible and solvable optimization is made possible by the proposed approach using Lagrangian Relaxation to translate tight QoS requirements into reasonable penalty terms inside the cost function. The H-permissible Path Routing Scheme (H-permissible Path Routing Scheme) filters out paths above a predetermined hop count (H_max), therefore restricting the search space without sacrificing QoS criteria, hence addressing scalability. Weighted and modified by dynamic Lagrangian penalties, the Aggregated Cost Calculation combines

parameters like cost, time, and dependability to guarantee optimal path choice. Moreover, Dynamic Controller Adaptation guarantees ongoing performance and service dependability by enabling the SDN controller to continually monitor network circumstances and fast alter routing pathways in response to congestion or QoS breaches.

3.4 Advantage of proposed model

- The proposed Hybrid Multi-constraint Lagrangian Relaxation-based Aggregated Cost (HMLR-AC) segment routing system offers several key advantages for QoS-aware Software-Defined Networks (SDNs).
- Firstly, Through the use of Lagrangian relaxation, it efficiently manages several QoS constraints, including bandwidth, latency, and packet loss, by combining them into a single optimisation issue. This makes complicated routing decisions more flexible and solvable. Secondly, the integration of the H-permissible Path Routing Scheme (HPRS) significantly reduces computational overhead by limiting path selection to routes within a specified hop count, thereby enhancing scalability. Additionally, the aggregated cost model ensures that selected paths maintain a balance between cost-efficiency and QoS adherence.
- Finally, the dynamic adaptation capability of the SDN controller allows real-time monitoring and path adjustment based on changing network conditions, leading to better resource utilization, reduced congestion, and improved overall network presentation.

4. RESULTS AND DISCUSSION

The simulation setup is conducted in this phase to evaluate the outcomes and the proposed algorithm. Key parameters such as link cost, energy efficiency, link utilization, and path evaluation values are used in the model simulations to monitor network activity across various controllers. Python is used in the Ubuntu environment to implement the simulation software, which is then deployed to the SDN controller. MATLAB is utilized to process and analyze the resulting data. The decision to simulate the HMLR-AC Segment Routing algorithm in QoS-aware SDN using Python and to analyze the results with MATLAB leverages the specific strengths of each environment to optimize the overall workflow. Python is chosen for the simulation phase due to its comprehensive libraries, including NetworkX, NumPy, and SciPy, which facilitate efficient network simulation and optimization. Its ease of use, flexibility, and strong community support make Python ideal for developing and testing complex algorithms like HMLR-AC. Once the simulation generates the necessary data, MATLAB is employed for its advanced analytical tools and superior data visualization capabilities. This combination ensures that development and execution benefit from Python's adaptability, while analysis gains from MATLAB's analytical precision and visualization strengths. All links are configured to operate at 1 Gbit/s for consistency. To conduct the studies, the traffic datasets are either measured or produced depending on the relevant topologies. Real network traffic datasets were collected explicitly for Abilene, Cernet, and Geant by capturing readings within the respective topology every 15 or 5 minutes. Conversely, Zoo's synthetic topologies and traffic datasets are created using the appropriate topologies' gravity models.

The results demonstrate that, among other performance measures, the model we propose, HMLR-AC, outperforms current routing systems in terms of throughput, packet loss ratio, latency, RMSE, network cost, and end-to-end delay. Some of the current systems used in this

article are DQN-based energy and latency-aware routing protocol (DQELR) [16], Deep Q-Network and Traffic Prediction based Routing Optimization (DTPRO) [14], Long Short-Term Memory (LSTM) based Recurrent Neural Network (RNN) [15], and Deep Neural Networks (DNNs) [17]. Below are the performance metrics of the HMLR-AC model we propose.

A Software Defined Network specific dataset [18] was generated by using Mininet Emulator and RYU simulator. Malicious (or DDoS) attack traffic flows are created for five hours, from early 8:00 AM to 1:00 PM, whereas legitimate traffic flows are created for one hour. DDoS attack traffic flows for protocols including Transmission Control Protocol (TCP), SYN assaults, LAND attacks, User Datagram Protocol (UDP), and Internet Control Message Protocol.

4.1. Throughput analysis

Throughput in QoS Aware Software Defined Networks denotes the amount of data or traffic volume that can be successfully transmitted within a given time frame. It relates to the significance of data that can move through a network design that supports Quality of Service (QoS) protocols and uses segment routing. This statistic is essential for analyzing the network's capability to successfully transport data and accomplish the performance values recognized by QoS policies in an SDN environment.

Table 1. Throughput analysis for HMLR-AC method.

Number of odes	DTPRO	LSTM-RNN	DQELR	DNNs	HMLR-AC
100	856.12	1234.67	945.19	1654.23	1854.23
200	874.78	1256.98	956.45	1634.98	1813.56
300	888.12	1345.78	1023.76	1723.87	1876.34
400	903.67	1432.78	1123.67	1734.98	1923.56
500	934.78	1545.34	1169.76	1787.45	1945.34

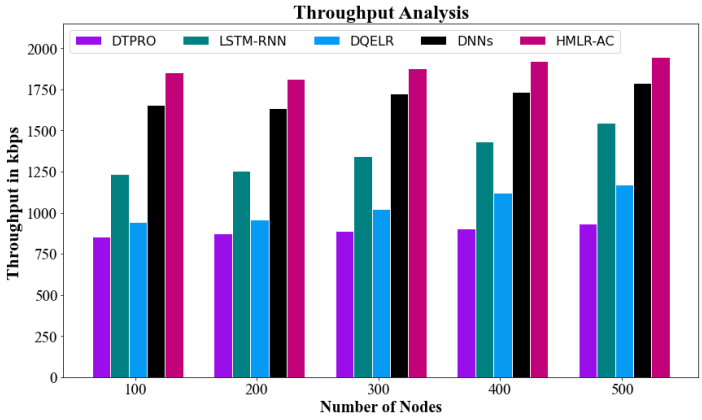

Figure 2. Throughput analysis for HMLR-AC technique.

In Table 1 and Figure 2, the throughput study of the HMLR-AC technology is shown in contrast to the existing methods. The graph founds unequivocally how superior in every way the recommended method is to the alternatives. The HMLR-AC technique, for example, has a throughput of 1854.23 kbps with 100 nodes, associated to the throughputs of the other known approaches, such as DTPRO, LSTM-RNN, DQELR, and DNNs, which are 856.12 kbps,

1234.67 kbps, 945.19 kbps, and 1654.23 kbps, respectively. Similar to this, the DTPRO, LSTM-RNN, DQELR, and DNNs methods have throughputs of 934.78 kbps, 1545.34 kbps, 1169.76 kbps, and 1787.45 kbps, respectively, while the suggested way has 1945.34 kbps with 500 nodes. This establishes the increased performance with greater throughput of the HMLR-AC method.

4.2. End to end delay analysis

End-to-end Quality of Service segment routing is used by software-defined networks (SDNs) to effectively track the time it takes for a packet to travel from its point of origin to its final purpose. Label-based routing strategies are used in segment routing. Being aware of QoS to set up and keeping track of QoS parameters, it uses software-defined networking. To ensure compliance with QoS standards, the end-to-end delay calculation considers transmission, processing, and queuing delays. This method ensures QoS compliance while enabling adequate packet travel time tracking throughout the network.

Table 2. End to end delay analysis for HMLR-AC technique.

Number of odes	DTPRO	LSTM-RNN	DQELR	DNNs	HMLR-AC
100	170.45	146.67	112.56	156.98	92.98
200	172.18	149.13	125.98	160.34	94.41
300	174.87	150.34	134.56	162.76	95.87
400	176.98	151.87	138.67	163.56	96.34
500	181.54	154.44	140.56	167.77	99.14

The end-to-end delay analysis of the HMLR-AC methodology using existing methodologies is designated in Table 2 and Figure 3. The graph demonstrates that the HMLR-AC technique surpasses the other strategies in every way. For instance, with 100 nodes, the HMLR-AC technique has taken only 92.98 ms as its end-to-end delay, while the different existing methods like DTPRO, LSTM-RNN, DQELR, and DNNs have an end-to-end delay of 170.45 ms, 146.67 ms, 112.56 ms, and 156.98 ms, respectively. Similar to this, the DTPRO, LSTM-RNN, DQELR, and DNNs have an end-to-end delay of 181.54 ms, 154.44 ms, 140.56 ms, and 167.77 ms, respectively, while it is 99.14 ms for the HMLR-AC approach with 500 nodes.

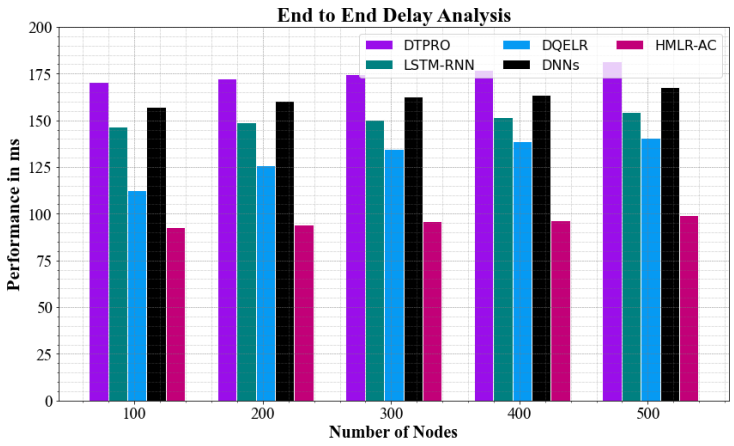

Figure 3. End to end delay analysis for HMLR-AC technique.

4.3. Latency analysis

Latency in Segment Routing in QoS Aware Software Defined Networks is the delay or lag experienced by packets or data flows in the network architecture that integrates SR and QoS algorithms in SDNs. It affects the performance and responsiveness of the network by measuring how long it incomes packets to cross the network and arrive at their destination. In real-time applications, including audio and video streaming, online gaming, and financial transactions, latency must be minimized to ensure data delivery.

Table 3. Latency analysis for HMLR-AC technique.

Number of Nodes	DTPRO	LSTM-RNN	DQELR	DNNs	HMLR-AC
100	81.67	51.45	41.67	31.78	21.56
200	80.85	50.98	42.98	32.98	24.67
300	88.12	56.87	47.93	36.98	25.56
400	90.45	56.66	46.87	33.19	24.98
500	92.19	61.34	47.77	39.77	27.77

Table 3 and Figure 4 compare the recommended HMLR-AC technique's latency to existing methods. The graph shows that the HMLR-AC method has overtaken all other approaches. The recommended HMLR-AC method has a delay of only 21.56 sec with 100 nodes, whereas other present techniques such as DTPRO, LSTM-RNN, DQELR, and DNNs take 81.67 sec, 51.45 sec, 41.67 sec, and 31.78 sec, respectively. Similarly, the suggested HMLR-AC approach delays by 27.77 sec with 500 nodes, while the existing techniques like DTPRO, LSTM-RNN, DQELR, and DNNs have taken 92.19 sec, 61.34 sec, 47.77 sec, and 39.77 sec of latency, correspondingly.

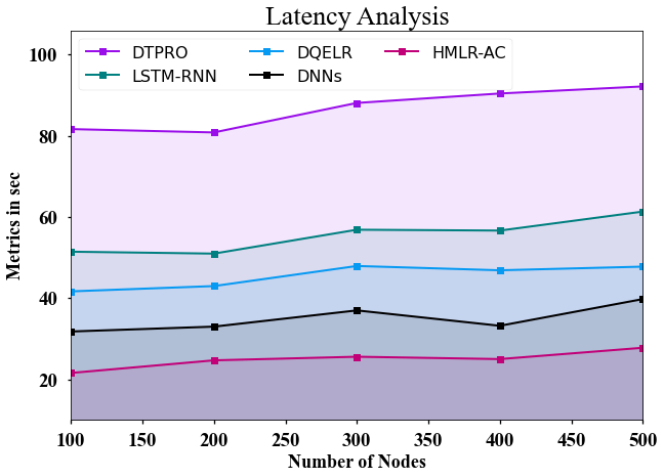

Figure 4. Latency analysis for HMLR-AC technique.

4.4. Packet loss ratio

In software-defined networks with QoS awareness, segment routing calculates the proportion of packets lost during transmission. The ratio of lost packages to all packets sent quantifies the network's dependability. Segment routing in QoS-aware SDN by monitoring and reducing packet loss optimizes performance and service quality.

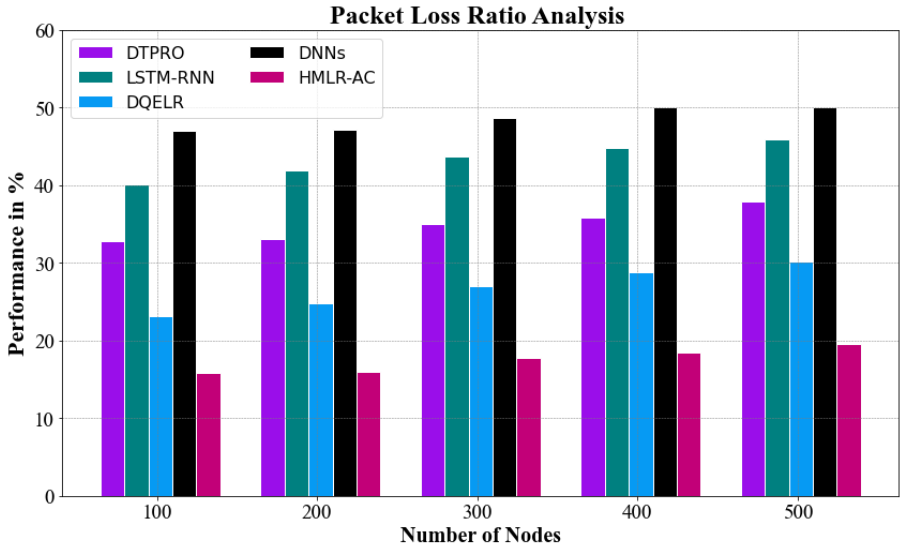

Figure 5. Packet loss ratio analysis for HMLR-AC technique.

In Figure 5 and Table 4, the packet loss ratio analysis of the HMLR-AC methodology is displayed in contrast to other methods. The graph shows how the DL method performs better with reduced PLR. In difference, the PLR values for the DTPRO, LSTM-RNN, DQELR, and DNNs models are 32.87 %, 40.12 %, 23.19 %, and 46.98 %, respectively, whereas the HMLR-AC model has a packet loss ratio of 15.87 % with 100 nodes. The HMLR-AC model, however, has demonstrated optimal performance for various nodes with low packet loss ratio. The PLR for the HMLR-AC model is 19.56 % with 500 nodes, compared to 37.87 %, 45.87 %, 30.12 %, and 50.12 % for the DTPRO, LSTM-RNN, DQELR, and DNNs models, respectively.

Table 4. Packet loss ratio analysis for HMLR-AC technique.

Number of nodes	DTPRO	LSTM-RNN	DQELR	DNNs	HMLR-AC
100	32.87	40.12	23.19	46.98	15.87
200	33.12	41.87	24.87	47.12	15.98
300	34.98	43.76	26.98	48.67	17.77
400	35.87	44.76	28.87	49.99	18.45
500	37.87	45.87	30.12	50.12	19.56

4.5. RMSE analysis

We use the Root Mean Square Error to accurately assess the overall effectiveness of our prediction models. The root of the average sum of squared errors, or RMSE, is the modification among the predictable and actual values.

In Figure 6 and Table 5, an RMSE analysis of the HMLR-AC methodology is displayed in contrast to other methods. The graph shows how the DL method has enhanced the performance with reduced RMSE. In difference, the RMSE values for the DTPRO, LSTM-RNN, DQELR, and DNNs models are 57.77 %, 51.11 %, 46.98 %, and 40.12 %, respectively, whereas the

HMLR-AC model has RMSE of 32.19 % with 100 nodes. However, the HMLR-AC model has shown to work at its best for several nodes with low RMSE values. The RMSE value for the HMLR-AC model is 37.98 % with 500 nodes, compared to 61.11 %, 56.23 %, 50.34 %, and 45.67 % for the DTPRO, LSTM-RNN, DQELR, and DNNs models, correspondingly.

Table 5. RMSE analysis for HMLR-AC technique.

Number of nodes	DTPRO	LSTM-RNN	DQELR	DNNs	HMLR-AC
100	57.77	51.11	46.98	40.12	32.19
200	58.12	52.78	47.78	42.67	34.98
300	59.99	53.98	48.12	43.33	35.12
400	60.23	55.67	49.99	44.98	37.77
500	61.11	56.23	50.34	45.67	37.98

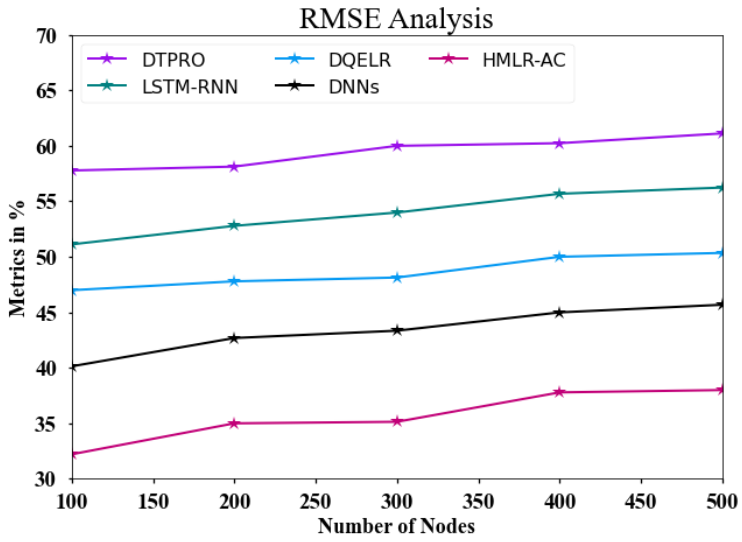

Figure 6. RMSE analysis for HMLR-AC technique.

4.6. Scalability analysis

Table 6. Scalability analysis for HMLR-AC technique.

Number of odes	DTPRO	LSTM-RNN	DQELR	DNNs	HMLR-AC
100	75.12	84.23	83.18	80.12	92.19
200	76.66	85.66	84.55	81.77	93.33
300	77.77	86.12	87.77	81.43	94.44
400	78.12	87.67	89.99	82.22	95.19
500	79.34	88.34	90.13	82.76	97.77

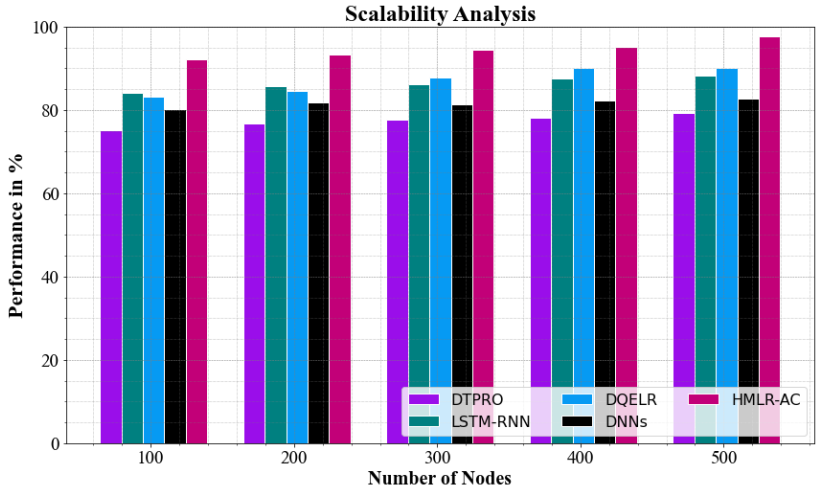

Figure 7. Scalability analysis for HMLR-AC technique.

Scalability means the network architecture can efficiently handle increasing workloads and accommodate a growing number of network nodes and services while maintaining good performance and QoS. It involves dynamically scaling network resources like bandwidth and processing capacity to meet changing needs and traffic patterns, delivering optimal performance and QoS guarantees for diverse forms of network traffic.

In Figure 7 and Table 6, the scalability of the HMLR-AC method is contrasted to other available techniques. The graph shows that the machine learning approach has led to an improved and scalable performance. As an illustration, the DTPRO, LSTM-RNN, DQELR, and DNNs models have scalability of 75.12 %, 84.23 %, 83.18 %, and 80.12 %, correspondingly, whereas the HMLR-AC model has 92.19 % with 100 nodes. The HMLR-AC model, nevertheless, has established its best performance using a dissimilar node. Like to this, the HMLR-AC model scales at 97.77 % under 500 nodes, likened to DTPRO, LSTM-RNN, DQELR, and DNNs models at 79.34 %, 88.34 %, 90.13 %, and 82.76 %, correspondingly.

5. CONCLUSIONS

This study presents a novel method, "Hybrid Multi-constraint Lagrangian Relaxation based Aggregated Cost (HMLR-AC) Segment Routing", to handle the QoS-aware routing problematic in SDNs. The HMLR-AC Segment Routing technique combines the benefits of both LR and AC approaches to progress routing decisions. Moreover, to successfully route traffic flows when path cardinality restrictions are in place, an HPRS is executed. It seeks to decrease entire network costs while meeting many QoS restrictions such as bandwidth, latency, and reliability. It also comprises the idea of H-permissible pathways, which are paths that match the given QoS values, providing high-quality service delivery. The recommended approach decouples the data plane and takes advantage of the centralized control plane. HMLR-AC creates use of the programmability and flexibility of SDNs. It uses a global network perspective and real-time traffic statistics to dynamically change routing decisions in response to changing network circumstances. This permits for more effective resource use and traffic load balancing, which leads to better network performance. Extensive simulations are run using a typical network situation to judge the efficacy of the HMLR-AC Segment Routing system. The significances

demonstrate that our new HMLR-AC model outperforms even the most sophisticated routing systems concerning throughput, end-to-end delay, network cost, scalability, latency, packet loss ratio, and RMSE, between other performance parameters. According to research data, the ideal HMLR-AC has a throughput of 1945.34 kbps, end-to-end delay of 99.14 ms, latency of 27.77 sec, packet loss ratio of 19.56 %, network cost of 47.77, and scalability of 97.77 %. Due to the limited experimental environment, we can expand our research by deploying SDN networks to real-world industrial Internet scenarios. SDN advancements can help transport networks support and deliver high-quality services.

Data Availability: No data were used to support the findings of this study.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

REFERENCES

1. Yungaicela-Naula N. M., Vargas-Rosales C., Pérez-Díaz J. A., and Zareei M. - Towards security automation in software defined networks, Computer Communications **183** (2022) 64-82. https://doi.org/10.1016/j.comcom.2021.11.014
2. Gupta N., Tanwar S., and Badotra S. - Performance analysis of ODL and RYU controllers' against DDoS attack in software defined networks, Cluster Computing **27** (8) (2024) 10899-10919. https://doi.org/10.1007/s10586-024-04535-y
3. Kim J., Seo M., Lee S., Nam J., Yegneswaran V., Porras P., and Shin S. - Enhancing security in SDN: Systematizing attacks and defenses from a penetration perspective, Computer Networks **241** (2024) 110203. https://doi.org/10.1016/j.comnet.2024.110203
4. Ghosh S., Dagiuklas T., Iqbal M., and Wang X. - A cognitive routing framework for reliable communication in IoT for industry 5.0, IEEE Transactions on Industrial Informatics **18** (8) (2022) 5446-5457. https://doi.org/10.1109/TII.2022.3141403
5. Zong L., Memon F. H., Li X., Wang H., and Dev K. - End-to-end transmission control for cross-regional industrial Internet of Things in Industry 5.0, IEEE Transactions on Industrial Informatics **18** (6) (2021) 4215-4223. https://doi.org/10.1109/TII.2021.3133885
6. Kaur K., Mangat V., and Kumar K. - A review on Virtualized Infrastructure Managers with management and orchestration features in NFV architecture, Computer Networks **217** (2022) 109281. https://doi.org/10.1016/j.comnet.2022.109281
7. Goswami B., Hu S., and Feng Y. - Software-defined networking for real-time network systems, In Handbook of Real-Time Computing, Singapore: Springer Nature Singapore, 2022, pp. 935-959. https://doi.org/10.1007/978-981-287-251-7_69
8. David O., Thornley P., and Bagheri M. - Software defined networking (SDN) for campus networks, WAN, and datacenter, International Conference on Smart Applications, Communications and Networking (SmartNets) IEEE, 2023, pp. 1-8. https://doi.org/10.1109/SmartNets58706.2023.10215722
9. Urrea C. and Benítez D. - Software-defined networking solutions, architecture and controllers for the industrial internet of things: A review, Sensors **21** (19) (2021) 6585. https://doi.org/10.3390/s21196585

10. Karmous N., Aoueileyine M. O. E., Abdelkader M., Romdhani L., and Youssef N. - Software-defined-networking-based one-versus-rest strategy for detecting and mitigating distributed denial-of-service attacks in smart home internet of things devices, Sensors **24** (15) (2024) 5022. https://doi.org/10.3390/s24155022
11. Akinsolu M. O., Sangodoyin A. O., and Uyoata U. E. - Behavioral study of software-defined network parameters using exploratory data analysis and regression-based sensitivity analysis, Mathematics **10** (14) (2022) 2536. https://doi.org/10.3390/math10142536
12. Liu Z., Wang Y., Feng F., Liu Y., Li Z., and Shan Y. - A DDoS detection method based on feature engineering and machine learning in software-defined networks, Sensors **23** (13) (2023) 6176. https://doi.org/10.3390/s23136176
13. Khedr W. I., Gouda A. E., and Mohamed E. R. - FMDADM: A multi-layer DDoS attack detection and mitigation framework using machine learning for stateful SDN-based IoT networks, Ieee Access **11** (2023) 28934-28954. https://doi.org/10.1109/ACCESS.2023.3260256
14. Bouzidi E. H., Outtagarts A., Langar R. - Deep reinforcement learning application for network latency management in software defined networks, In 2019 IEEE Global Communications Conference (GLOBECOM), 2019, pp. 1-6. IEEE.http://dx.doi.org/10.1109/GLOBECOM38437.2019.9013221.
15. Azzouni A., Pujolle G. - A long short-term memory recurrent neural network framework for network traffic matrix prediction, (2017). https://doi.org/10.1109/NOMS.2018.8406199.
16. Su Y., Fan R., Fu X., Jin Z. - DQELR: An adaptive deep Q-network-based energy-and latency-aware routing protocol design for underwater acoustic sensor networks, IEEE Access **7** (2019) 9091-9104.https://doi.org/10.1109/ACCESS.2019.2891590
17. Yu C., Lan J., Guo Z., Hu Y. - DROM: Optimizing the routing in software-defined networks with deep reinforcement learning, IEEE Access **6** (2018) 64533-64539. http://dx.doi.org/10.1109/ACCESS.2018.2877686
18. https://www.kaggle.com/datasets/pmpcse/software-defined-network-specific-dataset