A New Machine Learning-based Hybrid Intrusion Detection System and Intelligent Routing Algorithm for MPLS Network

Mohammad Azmi Ridwan[1], Nurul Asyikin Mohamed Radzi[2],
Kaiyisah Hanis Mohd Azmi[3], Fairuz Abdullah[4], Wan Siti Halimatul Munirah Wan Ahmad[5]
Institute of Power Engineering-Department of Electrical and Electronic Engineering-College of Engineering, Universiti Tenaga Nasional (UNITEN), 43000 Kajang, Selangor, Malaysia[1, 2, 3, 4]
Faculty of Engineering, Multimedia University, 63100 Cyberjaya, Malaysia[5]

***Abstract*—Machine Learning (ML) is seen as a promising application that offers autonomous learning and provides optimized solutions to complex problems. The current Multiprotocol Label Switching (MPLS)-based communication system is packed with exponentially increasing applications and different Quality-of-Services (QoS) requirements. As the network is getting complex and congested, it will become challenging to satisfy the QoS requirements in the MPLS network. This study proposes a hybrid ML-based intrusion detection system (ML-IDS) and ML-based intelligent routing algorithm (ML-RA) for MPLS network. The research is divided into three parts, which are (1) dataset development, (2) algorithm development, and (3) algorithm performance evaluation. The dataset development for both algorithms is carried out via simulations in Graphical Network Simulator 3 (GNS3). The datasets are then fed into MATLAB to train ML classifiers and regression models to classify the incoming traffic as normal or attack and predict traffic delays for all available routes, respectively. Only the normal traffic predicted by the ML-IDS algorithm will be allowed to enter the network domain, and the route with the fastest delay predicted by the ML-RA is assigned for routing. The ML-based routing algorithm is compared to the conventional routing algorithm, Routing Information Protocol version 2 (RIPv2). From the performance evaluations, the ML-RA shows 100 percent accuracy in predicting the fastest route in the network. During network congestion, the proposed ML outperforms the RIPv2 in terms of delay and throughput on average by 57.61 percent and 46.57 percent, respectively.**

Keywords*—*Machine learning; intrusion detection system; routing algorithm; quality of service; communication system

I. INTRODUCTION

Multi-protocol label switching (MPLS) routing technique for telecommunication networks was invented in the late 1990s as a more efficient alternative to the traditional Internet Protocol (IP) routing [1]. In contrast to traditional network protocols, which route data packets according to the source-to-destination (S2D) addresses, MPLS routes traffic from one node to another according to predefined labels in the packet header. These labels may contain information related to quality of service (QoS) such as traffic latency, jitter, packet loss and downtime, which allows network traffic to be prioritized according to its importance. One of the most noteworthy advantages of MPLS is its independence from any protocol or transport medium. It supports IP-based, Ethernet-based, asynchronous transfer mode (ATM), and frame relay transmission [1]. Other benefits of MPLS include [1]: 1) providing good QoS performance for latency-insensitive applications such as video and mission-critical data; 2) allowing data and voice applications to coexist on the same network; 3) allows the pre-programming of different types of data with distinct priorities and service classes; and 4) offers network scalability to users.

However, as the number of applications and users grows exponentially, conventional MPLS networks are likely to become more complicated and require stringent QoS regulations to ensure network reliability, delay tolerance, and throughput. Routing assignment (RA) algorithms used in conventional networks are typically based on the shortest path with fixed rules. They may not give the optimal QoS, particularly in a complicated network architecture. While standards and algorithms have been developed to increase the efficiency of the existing networks, it is anticipated that conventional approaches would be unable to meet growing demand while maintaining QoS. Additionally, the networks are facing threats from cyber attackers that take advantage of network vulnerabilities, resulting in extensive network disruption and significant damage to an organization's reputation. These challenges emphasize the critical importance of intelligent routing techniques and network security protection, such as that provided by network intrusion detection systems (IDSs).

An IDS monitors network traffic for signals of hostile activity by building a predictive model that can discriminate between attack and normal network flows. However, despite decades of advancements, existing IDSs continue to face detection accuracy challenges by reducing the false alarm rates and the identification of unknown threats [2]. Additionally, the fixed rules of IDS systems are vulnerable to threats including Denial-of-Service (DoS) and brute force [3]. To safeguard the network from such vulnerabilities, researchers all around the world have created cutting-edge IDS with the integration of machine learning (ML) algorithms. ML is capable of rapidly identifying patterns in a variety of data and solving complex, multi-dimensional problems with little to no human intervention. Since intrusion detection is a

The research and publication of this article was funded by UNITEN BOLD grant J510050002/2021061 and 202210020YCU.

classification problem, ML can be one of the promising candidates for IDS in the network. A Learning-based system is used in ML-based IDS to identify possible attack classes based on the behavior of an incoming packet. These ML-based IDSs offer various advantages over conventional systems, including lower computational loads and greater flexibility, as well as the ability to detect novel attacks and capture the complex features of attack behavior [4].

This research addresses several issues in MPLS networks, including improving network QoS and enhancing network security using ML algorithms. This work proposes a hybrid supervised ML-based IDS and ML-based RA algorithm (herewith will be referred to as ML-IDS and ML-RA, respectively) that is trained using an ML-IDS dataset generated using a simple data extraction method. The proposed ML-IDS is a security intrusion classification algorithm that analyses the incoming traffic patterns and network conditions and then classifies the traffic as legitimate or potentially intrusive. Afterwards, the ML-RA intelligently computes a route that is predicted to provide the best QoS requirements under any network condition. In short, the main contributions of this research work are as follows: 1) we proposed an ML-IDS algorithm that uses a simple data extraction method from the network to train the classifier without compromising the accuracy; 2) we developed an ML-RA algorithm that predicts the QoS parameters and performs path computing for the incoming traffic with various priorities in different traffic conditions; and 3) we introduce the first hybrid ML-IDS and routing algorithm (RA) that enhances network security and QoS.

This paper is organized as follows. Section II provides the literature review. The formulated ML-IDS and ML-RA methodology is presented in Section III. The discussion on the findings in the evaluation of the proposed strategy is presented in Section IV. Finally, Section V refocuses on the purpose of the research and draws conclusions for this study.

II. Literature Review

A. Routing Strategies in the Literature

One of the networking fundamentals responsible for selecting a path for packet transmission is network traffic routing. With proper network routing management, it is possible to achieve a QoS-compliant and cost-effective route, especially through the implementation of ML in network routing. However, ML-based traffic routing is often challenging because of various constraints including complex and dynamic topologies, diverse traffic, and unique QoS requirements. In routing optimization problems, traffic and route matrices can be used to describe the input and output of ML algorithms [5]. To predict or select a path for incoming traffic, ML algorithms must learn the correlation between traffic inputs and link conditions. The recent applications of ML in routing can be divided into five routing objectives, which are discussed as follows:

1) Routing by predicting network parameters: In today's network operations and administration, it is critical to predict network parameters such as path or connection quality, delay, throughput, optical signal to noise ratio (OSNR), and incoming traffic. ML aims to improve overall network performance by learning from past data or the environment. For example, Alvizu et al. [6] trained an artificial neural network (ANN) using a public dataset from Milan to forecast the traffic load and variation and calculate the best resource allocation via dynamic optical routing in software defined networks (SDNs), thereby reducing energy consumption. In contrast, Choudhury et al. [7] introduced a hybrid machine learning (ML) model based on the Gaussian process (GP) method to forecast traffic volume for each traffic engineering tunnel over time, followed by forecasting the optical performance of new wavelengths in a multi-vendor environment.

2) Routing for QoS improvement: By controlling the network's delay, jitter, bandwidth, and packet loss ratio, a good QoS can be attained. However, with the explosion of traffic volume in the network, it can be challenging to fulfil the QoS specifications for each incoming traffic. Due to the network's complexity, conventional algorithms to improve the QoS parameters may be impractical. To meet the QoS requirements, researchers are continuously developing and refining unique solutions, such as ML-based algorithms, to maximize throughput while minimizing latency.

Nakayama et al. [8] proposed a routing scheme using the Markov Chain Monte Carlo algorithm to reduce the worst-case end-to-end delay of all the front-haul flows of the centralized radio access network (C-RAN) and ensure that all flows meet the latency requirements. The proposed solution successfully reduces all flow's latency, demonstrating that the ML-based approach can address the CRAN's queuing delay problem. Additionally, Stampa [9] proposed a deep RL agent that can optimize routing in accordance with a predefined target metric, such as the delay requirement in SDN. The deep RL model automatically adapts to the current traffic conditions and proposes a customized configuration that minimizes network delay. The suggested deep RL agents were able to reliably calculate total traffic intensities, and the average delay is less than the benchmark of 100,000 randomly created routes.

3) Low computation routing scheme: Incorporating ML in the network may lead to high computational load, especially when dealing with high dimensional input features or when using deep learning (DL)-based algorithms. With that, several works proposed a low computational routing scheme. For example, Hendriks et al. [10] proposed Q2-RA, which is hybrid of Q-routing and Multi-Agent Reinforcement Learning (MARL) RAs. In the algorithm, ad-hoc wireless nodes decide on a route by selecting the neighbor with the best Q-value as the next-hop destination. Although this algorithm has an additional modified reward function to meet the QoS criterion, it is comparable to Q-routing. Only training traffic is provided throughout the learning process to obtain the Q-values on the available path until it converged inside a predetermined threshold. The transmission of data traffic then starts once the rate of sending learning traffic has drastically lowered. The suggested Q2-RA performs better than the ad hoc on-demand distance vector method with QoS awareness and is more flexible to network condition changes.

Martin et al. [11] proposed a classifier that was trained using labelled Risk Weighted Assets (RWA) configurations and solved using inductive logic programming (ILP). The classifier can offer online network setup for newly arriving traffic matrices once it has been trained. In response to rapidly changing traffic patterns, it can dynamically adapt and reconfigure the network because of the quick computation of RWA configurations. Instead of calculating ILP for every incoming traffic, the network will remember prior ILP solutions and allocate a path in accordance with the historical data. As compared to the ILP method, this approach reduces computational time by up to 93%.

4) Congestion-control routing: Congestion is one of the main concerns for network providers as it can degrade the overall network performance. Through congestion control, network stability, fair resource allocation, and a reasonable packet loss ratio are all made possible [12]. In conventional routing protocols, previous network abnormalities, such as network congestion, are not learned. As network traffic keeps growing, the network is put under a lot of strain, which creates problems with resource management and allocation that affect traffic QoS. Given that the majority of networks are still using outdated routing systems, this congestion problem is getting more critical [13]. Additionally, routing systems were created for fixed networks that determine the shortest paths using distance vectors or link costs. Eventually, the network will experience excessive traffic load, which will severely degrade network performance. The conventional RAs frequently commit the same routing mistake when this condition recurs, leading to an unmanageable rise in delay and packet error rate. This is where the predictive ML models come in to attempt to overcome the congestion issue.

Due to the inflexibility of route selection in circuit-switched networks, their total routing performance is usually constrained. This issue is highlighted by the least loaded (LL) routing protocol, one of the network's routing protocols. Because of the excessive capacity consumption under conditions of heavy load, this routing protocol may result in subpar performance and overall inefficiency [5]. To assist the LL routing performance, a novel online-based supervised Naive Bayes (NB) classifier is proposed in [14]. The classifier forecasts the likelihood of future circuit blocking between node pairs. When a service is provided or denied, the network snapshot is stored as historical information to determine the best route for new service connections. The proposed solution outperforms the least-load and short-path conventional routing protocols in terms of the minimum number of extra hops, lowest blocking probability, and least amount of network capacity overconsumption.

Another example of congestion control is demonstrated by Tang et al. [13]. The authors proposed a real-time DL-based intelligent network traffic control method based on deep convolutional NN (deep CNN) with uniquely characterized input and output to represent the wireless mesh network backbone. The performance of the proposed scheme is compared with Open Shortest Path First (OSPF), Intermediate System to Intermediate System (IS-IS), and Routing Information Protocol (RIP). The simulation results showed that the DL-based routing scheme is superior to other routing protocols, as 98.7% of congestion cases are avoided.

5) Load-balancing routing: The bursty nature of SDN packet traffic creates a network load imbalance. Yao et al. [15] proposed a pair of ML-aided load balance routing schemes that take queue utilization (QU) into account to address this issue. The aim is to improve load-balance routing by reducing the packet loss ratio and improving the worst throughput. To deal with network congestion caused by a sudden traffic burst, ANN algorithms predict the QU for the next time slot. The predicted value is used to guide intelligent routing decisions. When compared to the shortest path approach, the proposed scheme improves packet loss ratio and throughput while increasing delay by 20%.

The next-generation wireless network (NGWN) is a network service and operation interface that can support multiple standards such as 5G, Wi-Fi, and cognitive radio networks. However, the volume of traffic in the current communication infrastructure is expanding rapidly that the router’s speed may not be sufficient to keep pace. Additionally, the NGWN's real-time load balance request cannot be satisfied and served by using conventional routing schemes that are solely based on standard rules and have limited computing capacity [16]. To anticipate the network queue state, which is one of the measures for making wise routing decisions, Yao et al. [16] suggested a load balancing routing based on NN. The proposed algorithm is compared to shortest path-based algorithms such as Bellman-Ford (BF) and Queue-Utilization BF (QUBF), in terms of throughput and delay. According to the results, the proposed technique reached the highest throughput while incurring a 20% delay over the BF approach. The proposed algorithm also can predict the next-hop path with the smallest buffer and thus improve load balancing.

B. Challenges in Routing

From the literature review, recent related works on ML-based RA have proven to route the traffic effectively. Almost all related works from literature successfully overcome the limitations of conventional routing protocols. Despite ML’s superiority in routing in communication networks, there are still some challenges to consider, discussed as follows.

1) Trade-off between accuracy and computational load: The trade-off between accuracy and computational load in ML-based RA using classical ML and DL is similar to the issue discussed for ML-based IDS. This trade-off must be considered because ML routing decisions need to be swift and in real-time to avoid processing delay. Unlike the proposed ML-RA in the literature which used DL-based classifiers, our work aims to use ML-based regressions including DT-regressions and LR by Gibbs Sampling (LRgs) to predict the delay in all available routes. DT and LRgs are well-known for their simplicity and interpretability, but they are prone to overfitting. To address this issue, we will train, test, and evaluate our proposed ML-RA under various network and traffic conditions.

2) Lack of congestion and link failure scenario for routing: The majority of associated studies in the literature test the effect of their proposed ML-based routing scheme under congested network condition, as evident from the studies in [13], [17]–[22]. Only a few research, such as [23], develop ML-based RA that considered both congestion and link or node failures. Therefore, our proposed ML-RA considers both congestion and link failure scenarios.

3) Traffic modelling for performance evaluation: As the network becomes more complex and congested with traffic of varying priorities, it is critical that the ML-based routing mechanism can deliver traffic while meeting QoS requirements. However, most related works only consider single traffic type for routing, which may not be feasible to resemble real-world traffic with different priorities. There has been little research into ML-based routing for traffic with varying QoS requirements. For example, in [24], the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) traffic are considered, which correspond to voice over Internet Protocol (VoIP) and video traffic. While the authors [10] used three traffic priorities; high, medium and low priority, they did not explicitly specify the traffic types. Our proposed ML-RA will consider three traffic types which are expedited forwarding (EF), assured forwarding (AF), and best-effort (BE) traffic, which correspond to VoIP, close-circuit television (CCTV), and data transfer, respectively. It is expected that our proposed ML-RA can successfully route all traffic within their QoS requirements.

4) Quality datasets for training: When generating traffic in the network simulator, it is crucial that the traffic pattern is not random or static as demonstrated in [9], [11], [14], [17], [18], [22], to preserve the quality of the dataset used to train the ML algorithms. To improve the quality of the dataset, the traffic packets can be modified to resemble legitimate traffic properties, such as the standardized data rates and size. In this work, the ML-RA dataset is constructed using EF, AF, and BE traffic following the standard of VoIP, CCTV, and file transfers. More details on the traffic properties will be elaborated in Section III. In addition, the traffic is also modelled using the typical EF, AF, and BE traffic mixture ratios of 20:40:40 for [25].

Table I summarizes the discussed ML works in routing with their advantages and shortcomings.

TABLE I. SUMMARY OF RECENT ML-BASED RAS WITH THEIR ADVANTAGES AND SHORTCOMINGS

Authors	Routing Objectives	Description	Issues of Conventional Routing Protocol	ML Method	Advantages	Shortcomings
Yao et al. [15]	Load balancing	Proposed a pair of ML-assisted load-balancing RAs that consider QU, to improve load-balance routing by packet loss ratio reduction and improving the worst throughput	High computational complexity for QoS RA	DL	Improved global realignment and more efficient network optimization	• High computational load • Only considers two traffic patterns (Steady and congest)
Yao et al. [16]	Load balancing	Proposed NN-based load-balancing RA to predict network queue status to make intelligent routing decisions	Traditional RAs cannot always serve the NGWN effectively	DL	Enhance the bit error rate, throughput, and delay	• High computational load • Do not consider link failure • Do not consider different traffic priorities
Fadulullah et al. [23]	Predicting network parameters	Proposed a value iteration architecture-based deep RL routing approach, which includes the network node's adjacency matrix as learning parameters. The method can forecast the next node until the destination is reached	High computational cost to address RWA problems	DL	Ensures more stable network performance in the event of network topology changes	• High computational load • Do not consider different traffic priorities
Murudkar et al. [18]	Predicting network parameters	Proposed a User Specific-Optimal Capacity Shortest Path RL routing in 5G networks is to establish the resource-based optimum-capacity shortest route for a user between S2D pairs	Challenging spectrum resource optimization	RL	Quickly determine the shortest route with the highest capacity	• Only constant bit rate traffic in the simulations • Do not consider different traffic priorities • Do not consider node failures condition
Salani et al. [17]	Predicting network parameters	Proposed an integration of RF-based estimation for routing and spectrum assignment for quality of things	Inaccessible of perfect transmission knowledge to train the ML model	RF	Saves up to 30% on spectrum occupation.	• Traffics in the simulations are generated randomly • Do not consider different traffic priorities • Do not consider link failure

Vashishth et al. [22]	Low computation routing scheme	Proposed cascade learning, an ensemble-based ML that combines LR and NN classifiers. Using the ML-based Probabilistic Routing Protocol and the History of Encounters and Transitivity (MLProph) as input, the logistic algorithm will generate two probabilities: delivered or not delivered	Context-free routing protocols suffer from high network overhead ratio and congestion	Ensemble LR and DL	Enhance message delivery probability, network overhead ratio, average hop count, and message drop rate.	• Data generated is based on a flooding-based routing protocol to train the ML algorithms • Do not consider different traffic priorities • Do not consider node failures condition
Vashishth et al. [21]	Low computation routing scheme	Proposed a routing approach based on the Gaussian mixture (GM) model classifier in the Opportunistic Internet of Things (OppIoT) that increases message delivery probability	Fixed S2D path is non-existent, making routing very challenging	GM	Increase message delivery probability, the average hop count, the number of dropped packets, and the network overhead ratio	• Delay is not one of the performance parameters • Do not consider node failures
Hendriks et al. [10]	Low computation routing scheme	Proposed a hybrid of Q-routing and MARL RA. Ad-hoc wireless nodes use the algorithm to make routing decisions by selecting the neighbor with the best Q-value as the next hop	Traditional RAs have a high overhead load to be used in an ad hoc environment	RL	Outperforms well-known ad-hoc RAs in dynamic environments with QoS constraints	• Do not consider node failures or congestion
Li et al. [14]	Congestion-control routing	Proposed an online-based supervised NB classifier for performance improvement. The classifier predicts the likelihood of future circuit blocking and uses the data to select routes for future service connections	Fixed route oriented significantly limits the routing performance and flexibility	NB	Saves ~ 90% of the time for the learning process, significantly speeding up simulation studies.	• Traffic in the simulations is generated randomly • Do not consider different traffic priorities
Tang et al. [13]	Congestion-control routing	Proposed a real-time DL-based intelligent network traffic control based on DCNN for a wireless mesh backbone	OSPF for training the ML-RA, which lacks the necessary intelligence to handle newly occurring situations	DL	Avoid 98.7% of congestion cases	• High computational load • Do not consider different traffic priorities • Do not consider link failure scenario
Pasca et al. [20]	QoS improvement	Proposed an application-aware multipath flow routing framework integrating ML in SDN for traffic classification. The algorithm assigns paths based on QoS requirements of available parameters e.g., bandwidth and delay	Traditional static routing is slow to respond to network changes and slow to converge	NB, DT, Bayesian Network and SVM	Provide better routing configuration effectively reduces the network delay	• Do not consider link failure • Do not improve on delay
Nakayama et al. [8]	QoS improvement	Proposed a routing scheme based on the Markov Chain Monte Carlo algorithm to decrease the worst-case end-to-end delay of all CRAN front-haul flows and ensure that all flows fulfill the latency requirements	The current QoS-aware routing scheme ignores frame-level queuing delay.	Markov Chain Monte Carlo	All flows have a delay that is less than the threshold and meets the requirements.	• Do not consider node failures or congestion • Different class traffic is considered but did not mention explicitly
Stampa et al. [9]	QoS improvement	The authors designed and tested a deep RL agent in SDN that could significantly improve routing based on delay requirements	Limited routing optimization capabilities	Deep RL	Achieves improved delay when compared to the non-DRL agent routing scheme	• Data samples used are 100,000 gravity generated traffic matrix • Do not consider different traffic priorities • Do not consider link failure and congestion
Mao et al. [19]	QoS improvement	Proposed Tensor-based Deep Belief Architectures (TDBA), in which traffic patterns from the edge router are fed into TDBA to build a path to all edge routers	Conventional routing cannot cope with the complex environment	DL	Achieves zero packet loss rate	• High computational load • Do not consider different traffic priorities • Do not consider link failure condition

Alvizu et al. [6]	Predicting network parameters	Trained an ANN for forecasting the traffic load and variation and calculate the best resource allocation in SDN	Over-provisioning yields inefficiency and high operational costs	DL	The proposed scheme yields an optimality gap above 3%	• Training using a public dataset • Do not consider link failure

III. Development of the Hybrid ML-IDS and ML-RA

The ML-IDS dataset development is carried out via Graphical Network Simulator 3 (GNS3) by varying the network inputs. The normal incoming traffic is generated via the OSTINATO traffic generator, while the attack traffic is generated via Low Orbit Ion Cannon (LOIC). The output of the simulations in GNS3 for the ML-IDS algorithm will be the actual normal or attack label, while the output for ML-RA is the actual delay for all available routes between a S2D pair. All inputs and outputs are extracted via Wireshark and tabulated in a CSV file to build the ML-IDS and ML-RA datasets.

For ML-IDS, since it is a classification-based algorithm, the dataset is fed into MATLAB to train several ML classifiers. In contrast, ML-RA is a regression-based algorithm, and the ML-RA dataset is fed to train ML regression models. Both datasets are split into 70% training dataset and 30% testing dataset for performance evaluation during the algorithm development phase.

To further improve the performance of the ML models, the hyperparameters are optimized iteratively in MATLAB until the performance, i.e., error rate, converges to a constant value. Then, all the ML models are further tested using new data. The new data consists of new input features but without the actual output label. It is up to the ML models to provide predictions on the new data. The model which provides the most promising performance such as accuracy, precision, and F-measure are chosen for the proposed ML-IDS and ML-RA. Finally, both ML-IDS and ML-RA are cascaded together to build a new hybrid algorithm to enhance network security and improve network delay and throughput.

The simulation setup, the proposed algorithm, system parameters, and simulation scenarios are discussed in detail as follows.

A. The Simulation Setup

The network environment for the hybrid ML-IDS and ML-RA in the MPLS network system is depicted in Fig. 1. The network consists of eight edge routers, R1, R2, R5, R6, R7, R8, R9, and R10, which can be used as ingress or egress label edge routers (LER). Concurrently, R3 and R4 are normal label switch routers (LSRs) in the MPLS domain. All edge routers are linked to various types of traffic, such as VoIP, CCTV, and file transfers. The network is built in four ring topologies: 1) R1, R2, R3, R4, R5, and R6 form the main ring; 2) R1, R4, R5, R7, and R8 form the second ring and serve as node protection for R4; 3) R2, R3, and R9 form the third ring; and 4) the final ring is made up of R3, R6, and R10. The third and fourth rings protect the links between R2-R3 and R3-R6, respectively. In this network environment, all links are active, and the conventional RA and proposed ML-RA must compute the route for all traffic.

GNS3 is used to build the network for simulation, data collection, and performance analysis. OSTINATO and LOIC are used to generate normal and attack traffic, respectively. All edge routers are connected to either a Virtual PC (VPC), Windows virtual machine (VM), or OSTINATO traffic generator. The VPC is placed in the edge routers for traffic monitoring while the VM is used to mimic an actual Window’s PC in the network for file transfers, generate DoS traffic and for traffic monitoring. Note that generated traffic can be sent using several streams simultaneously using different protocols at different rates.

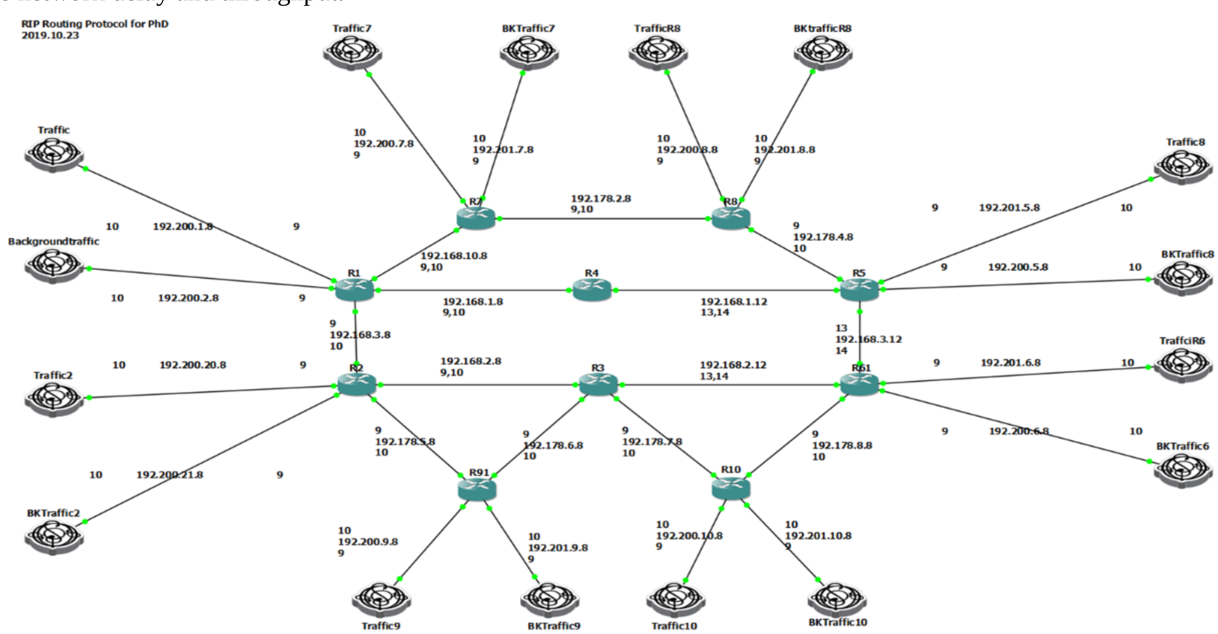

Fig. 1. Network environment in GNS3.

To launch a DoS attack, the LOIC tool is first installed in Window's VM in GNS3. After configuring the virtual Ethernet port of the VM, the IP address of the client or server is entered in the network as the DoS attack target, followed by the DoS method and packet flooding speed. The LOIC will flood the route leading to the targeted client. Once the DoS attack began, all network Virtual Private Clouds (VPCs) lost connectivity to the targeted client, while all clients connected to the route that links to the target client are also affected by the DoS attack. This scenario illustrated the severe damage of a DoS attack in the network domain as the attack affects not only the victim but also the devices linked to it.

B. Hybrid Supervised ML-IDS and ML-RA

This section discusses how ML-IDS and ML-RA are cascaded together to form complete ML-based security and QoS enhancement algorithms. Fig. 2 shows the framework of the proposed hybrid ML-IDS and ML-RA. The ML-IDS is incorporated at every ingress router that is R1, R2, R5, R6, R7, R8, R9, and R10, while the rest of the routers only focus on forwarding the traffic as computed by the ML-RA. ML-IDS is a classifier-based supervised ML that will predict the incoming traffic as normal or attack. In contrast, ML-RA, is a regression-based supervised ML that predicts the delay of all possible routes between the S2D pair.

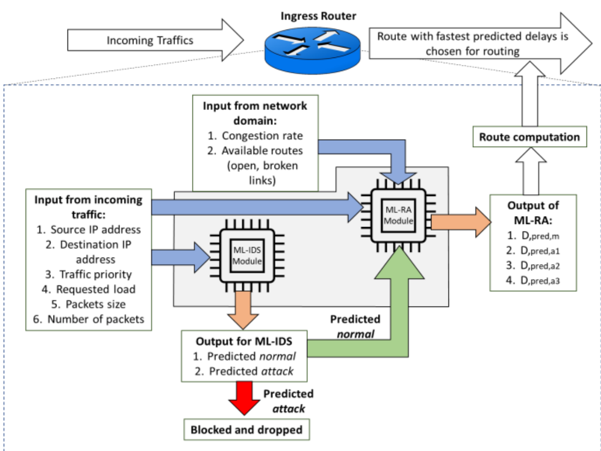

Fig. 2. Framework for the hybrid ML-IDS and ML-RA.

As incoming packets enter the ingress router, network information such as S2D IP address, traffic priority, requested load, packet size, number of packets, and data rates are fed into the ML-IDS. Additional network domain information, such as congestion rate and available routes, is required as inputs for the ML-RA. The output for the ML-IDS will classify the incoming traffic. If it is predicted as attack traffic, the ML-IDS will block and drop the traffic from entering the MPLS network. However, if it is predicted as normal by ML-IDS, the traffic will feed into the ML-RA module for route computation. To build the ML-IDS, the dataset constructed from the data generated using LOIC and OSTINATO is tabulated in a CSV file format. The ML-IDS dataset is later fed into the Classification Learner App (CLA) in MATLAB, which trains model to classify data using supervised ML. This application allows users to import datasets, select features, specify validation schemes, train models, and assess results. Automated training of the ML algorithms allows users to select the models with the best classification model types. These automated training features ease the model development and evaluation process by eliminating the trial-and-error process to choose the best ML classifiers.

After importing the dataset into the CLA, the data is ready to be trained by a series of ML algorithms, and the best performing algorithm will be chosen for the proposed ML-IDS. The overall process to develop the proposed ML-IDS is as shown in Fig. 3. The dataset is split into a training dataset and test dataset with a split ratio of 70:30 [26]. However, the split ratio must be carefully adjusted to avoid over-fitting or under-fitting. A trial-and-error basis is generally adopted until the accuracy saturates.

The dataset of ML-RA is built in GNS3, covering the incoming traffic and network information from the MPLS domain. Simulations are conducted in GNS3 using different traffic parameters in the OSTINATO traffic generator for EF, AF, and BE traffic. For congestion, another OSTINATO traffic generator is run in GNS3 by bursting continuous packets. Broken links are simulated by simply closing the link in GNS3. The available routes in the network are performed by using RIPv2 routing protocol. RIPv2 was chosen due to ease of configurations in the Cisco emulator in GNS3 and because RIPv2 utilizes shortest-path routing scheme regardless of network conditions.

At first, a random S2D pair is chosen. Then, by using trace IP route in the Cisco Command Line Interface (CLI), GNS3 will show the main route computed by the RIPv2. Then, the main route is purposely closed to allow the RIPv2 to recompute to other available routes. The process is repeated until there are no more alternatives for S2D pair. For this research, the alternative routes are already trained by the ML-RA using the RIPv2 routing protocol. The advantage of this method is that ML-RA algorithm no longer needs to manually compute alternative routes in the network when a sudden surge of traffic in the network occurs. The ML-RA algorithm is trained so that it will predict the QoS parameters on all available routes and quickly assign a path for the traffic in any incoming packets and network conditions. For each S2D pair, different EF, AF, and BE traffic configurations, and network conditions are run in the simulations for several iterations to improve the ML-RA training rate. Since the delay and throughput of each iteration are not always precisely constant due to processing delay and limitation of the simulation platform, training it with several iterations will allow the ML-RA to foresee the patterns and trends in the network.

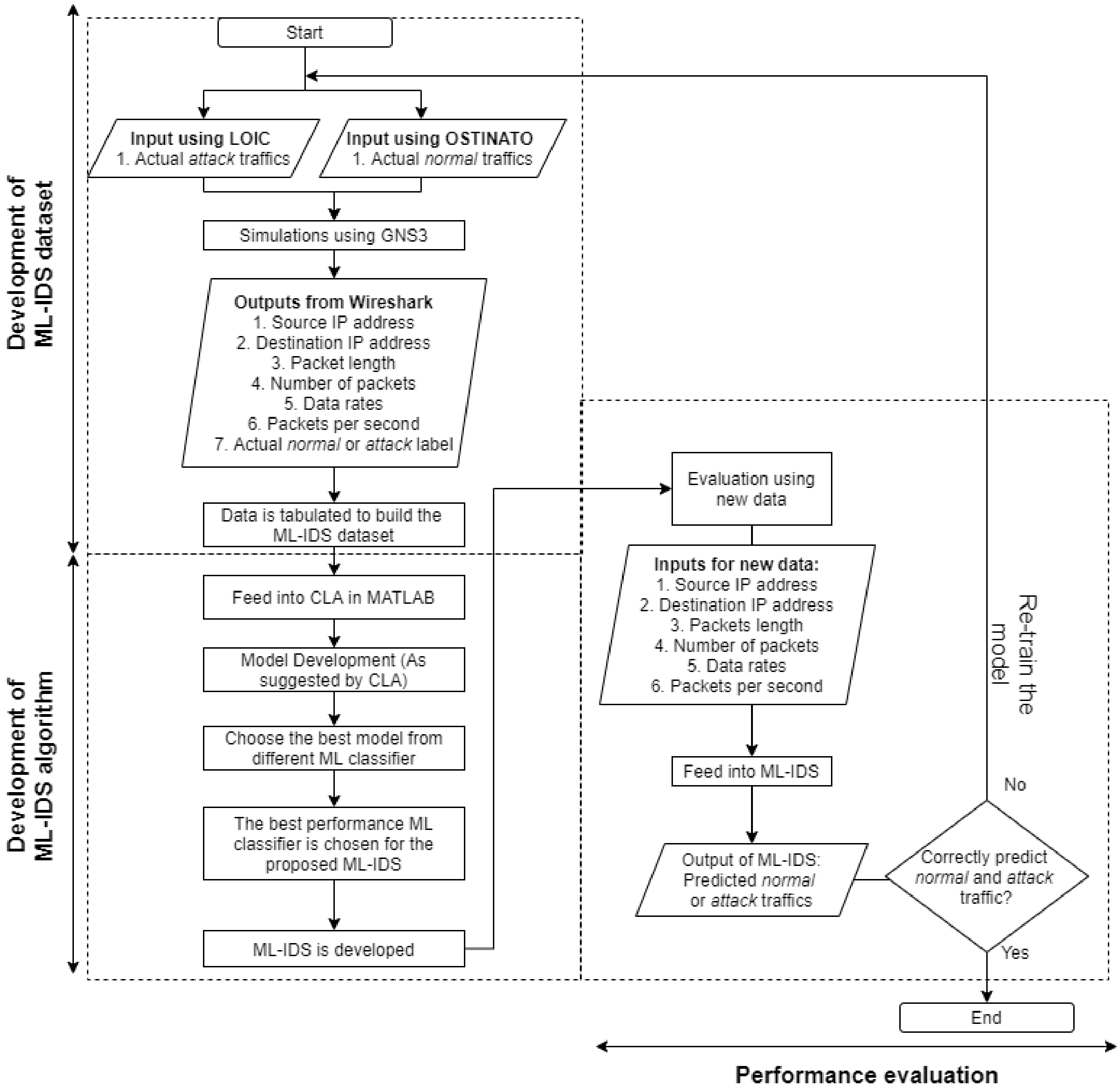

Fig. 3. Flowchart for the development of ML-IDS.

There are a total of 21 features for the ML-RA dataset. The first five input features are the information of the incoming traffic while the rest are on network conditions. To simplify the dataset, network congestion and broken links share the same column, where the value "0" denotes zero congestions, value "20" denotes congestions, while "100" denotes a broken or closed link. The data extracted from the simulations is up to half-a-million of iterations for all possible S2D pairs with different traffic and network conditions.

The ML-RA dataset is then fed into MATLAB's Regression Learner App (RLA) to create a regression model that predicts the delay for each route in the network. The RLA eases the ML development work by suggesting several regression models that fit the ML-RA dataset. For this case, the linear and tree-based regressions algorithms, including medium tree, course tree, and fine tree are suggested by the RLA. All the regression's algorithm is then compared with their prediction speed, training time and RMSE. The best one among all will be chosen as the ML model for the ML-RA. Delay in this context is defined as the total time taken from the moment the first packets enter the MPLS domain to the time the last packets are received at the receiver end. The predicted delay by the ML-RA for the main route, Alternative Route 1 (ALT1), Alternative Route 2 (ALT2), and Alternative Route 3 (ALT3) are denoted as $D_{pred,m}$, $D_{pred,a1}$, $D_{pred,a2}$ and $D_{pred,a3}$, respectively. The predicted delays by the ML-RA will be used for route computation. The flowchart for the route computation is as shown in Fig. 4. The route computation for each traffic will be based on four network conditions, namely, all routes available, one or two routes down, three routes down, and all routes down based on the predicted delay. If the route is not available either due to broken links for node failures, the ML-RA will predict a value of "∞" to the route, indicating that the predicted delay is too high and should be avoided.

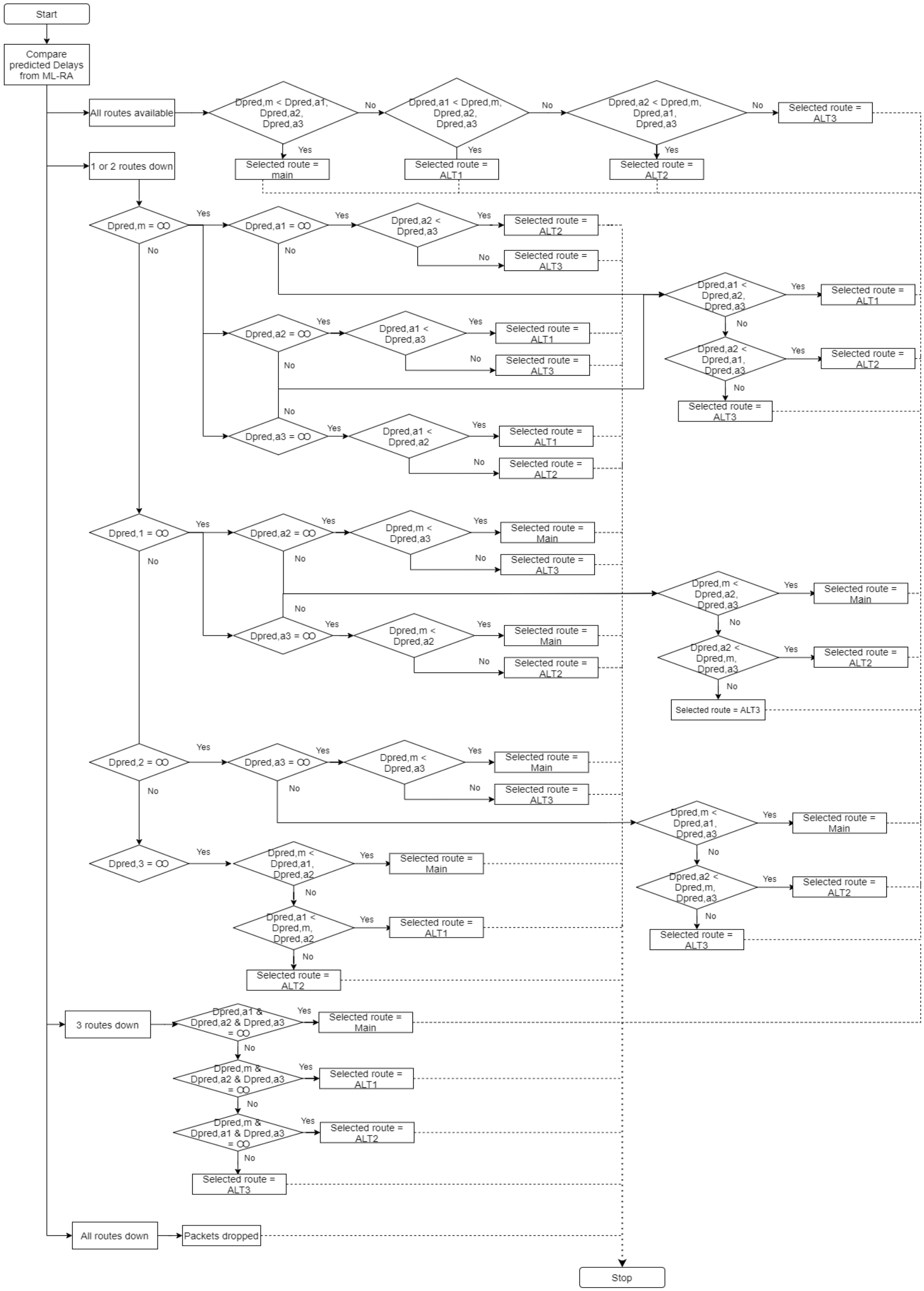

Fig. 4. Route computation flowchart for the proposed ML-based hybrid IDS and intelligent routing algorithm.

As shown in Fig. 4, when all routes are available, the ML-RA module will rank each of the predicted delays from lowest to highest. Then the route with the lowest predicted delays will be chosen for the traffic, starting with EF, followed by AF and BE traffic. In the event where all routes are down, ML-RA will drop all the packets until a route is back to available. One of the advantages of ML-RA is that from the predicted delay alone is already sufficient to understand the network conditions. For instance, when the link is congested, the predicted delay will be high. While, when the link is down or node failure, the predicted delay is "∞". This eases the route computation works to choose the best route for the traffic without processing too much network information.

C. *System Parameters*

The design parameters for ML-IDS are shown in Table II. Note that the traffic generated by both OSTINATO and LOIC traffic generators in GNS3 follows the IEEE 802.3 standard, where the packet length is between 64 to 1520 [27]. The OSTINATO traffic generator is limited to only 20 Mbps for normal traffic, while LOIC generates attack traffic up to 40 Mbps, with up to 40,000 packets per second. However, due to limited computing storage, the number of packets per session is capped at 20,000 packets per session.

The design parameters for the ML-RA are as shown in Table III. The traffic in the network comprises of EF, AF, BE, and congested traffic. The requested load begins at 20% with an increment of 10% up to 100% according to the traffic mixture ratio of 20:40:40 for EF, AF, and BE, respectively. Our finding shows that the maximum network capacity in GNS3 is at 20 Mbps. With that, at 100% requested load, the throughput for EF, AF and BE is 4 Mbps, 8 Mbps, and 8 Mbps, respectively. The packet length for EF traffic is fixed at 160 bytes, following the Cisco bandwidth calculator [28]. Due to the limited computational storage of the GNS3 VM and ease of data extraction, the packet size for AF and BE traffic is fixed at 500 bytes. However, for congestion traffic, the length of the packet is set at random between 60 to 1550 bytes with the number of packets up to 20,000 packets per session using the "random" feature in OSTINATO.

D. *Simulation Scenarios for Performance Study*

Four S2D pairs are chosen for ML-RA validations: R6 to R2, R2 to R5, R10 to R7, and R5 to R3. These S2D pairs were selected because there are one main route and three alternative routes; namely ALT1, ALT2, and ALT3 available and it involves almost the entire link in the network, which is suitable for performance study. It is worth mentioning that there are also other S2D pairs available. However, because the four pairs with all the available routes already involve the entire links and LSRs in the network, it is deemed sufficient.

For ease of explanation on the routing, each of the routes in the network is given a unique LSP ID, as shown in Fig. 5. For instance, the route between R1 to R4 is LSP 1. For routes with more than one hop, for example, the route between R1 to R6 is denoted as LSP 1.2.3, which represents a total of three links. The main and alternative routes computed by the RIPv2 routing protocol for R6 to R2, R2 to R5, R10 to R7, and R5 to R3 are tabulated in Table IV. When the network is not congested, the shortest path offers the best QoS parameters. RIPv2 will always compute the shortest path, which in this case, is the main route between an S2D pair regardless of network congestion. The downside is, when there is a sudden change in the network, RIPv2 may not be able to perform at its peak. When the main route is broken, the RIPv2 will re-compute the next best route, which is the ALT1 and so forth.

To force the traffic to choose another alternative route, traffic engineering is configured via CLI in the Cisco router's Internetwork Operating System. A sample case study is demonstrated in this paper, in which the main route for an S2D pair is congested, with one network route is down. The purpose of this sample study is to investigate how the proposed ML-RA can compute the path for different network conditions. The accuracy of the ML-RA is considered good when it can avoid congested routes and broken links. As a result, it is expected that the ML-RA will offer better QoS performance compared to RIPv2 that only considers the shortest path. The objective of ML-RA is to predict and compute the best routes in the network, regardless of the network conditions. The simulation details in GNS3 for ML-RA performance study are summarized in Table V for R6 to R2, R2 to R5, R10 to R7, and R5 to R3. The performance study focuses on the accuracy of the ML-RA to choose the route with the predicted best QoS. Then, the delay and throughput for the route are computed by the ML-RA and compared with RIPv2.

TABLE II. DESIGN PARAMETERS FOR THE PROPOSED ML-IDS

Design Parameter	Description
Packet length	60 bytes < Packet length < 1520 bytes
Number of packets	Up to 20,000 packets per session
Data rates	Up to 40 Mbps
Packets per second	Up to 40,000 packets per second for DoS attack
ML-based classifier	GBT, RF, DT, DL, FLM, LR, and GLM
Dataset	Proposed ML-IDS dataset and CICIDS-2018
Actual traffic type	The actual traffic from the dataset's label (normal or attack)

TABLE III. DESIGN PARAMETERS FOR THE PROPOSED ML-RA

Design Parameter	Description
Traffic Priority	EF, AF and BE
Requested Load	20%, Δ ± 10% up to 100%
Packet Length	Background traffic (60 bytes to 1550 bytes), EF (160 bytes), and AF and BE (500 bytes)
Number of Packets	Up to 20,000 packets per session
Data Rates	Up to 20 Mbps for 100% load
Congestion Rates	Fixed at 20 Mbps
ML-based Regressions	Linear Model Regressions, Medium Tree Regressions, Fine Tree Regressions, and Course Tree Regressions
Dataset	Proposed ML-RA dataset
Routes	Main route, ALT1, ALT2, ALT3
Delay	Actual delay of the traffic from simulation during the training phase

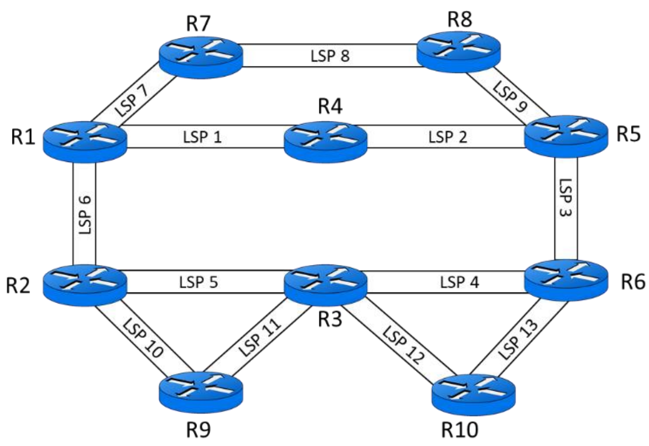

Fig. 5. Network environment with LSP ID.

TABLE IV. MAIN AND ALTERNATIVE ROUTES BY RIPV2 ROUTING PROTOCOL

Source router	Destination router	Main route (LSP)	ALT1 (LSP)	ALT2 (LSP)	ALT3 (LSP)
R6	R2	4.5	13.12.11.10	3.2.1.6	3.9.8.7.6
R2	R5	6.1.2	6.7.8.9	5.4.3	10.11.12.13.3
R10	R7	12.5.6.7	12.11.10.6.7	13.3.9.8	13.3.2.1.7
R5	R3	3.4	3.13.12	2.1.6.5	9.8.7.6.5

IV. RESULTS AND DISCUSSION

This section will discuss the results of the proposed system, along with some of its limitations and the future direction of the hybrid ML-IDS and ML-RA system.

A. Simulation Results

Delay is an essential parameter in evaluating the performance of ML-RA, especially for EF and AF traffic, as they require stringent delay requirements. There is no strict timing delay requirement for BE traffic. However, for comparison purposes, the delay for BE traffic is compared in different network conditions and different routing protocols and measured via the analysis tool in GNS3's Wireshark. Provided that it is within the delay standards, the lower the delay, the better the performance of ML-RA.

The next performance matrix evaluated is throughput, which is defined as the amount of data transferred between a S2D router to show the performance between ML-RA and RIPv2 RA [29]. AF and BE traffic require high amount of throughput compared to the EF traffic. It is also expected that the higher the congestion rate and the number of hop count for selected LSP will lower the throughput performance. Similar to delay, throughput is also measured via the analysis tool in GNS3's Wireshark.

Delay and throughput performance parameters are chosen for benchmarking purposes. This is because one of the problem statements is that the shortest path in the network will not provide the best QoS especially when the network resources are limited. The delay comparison between ML-RA and RIPv2 will show which RA performs the best with different traffic and network conditions. The delay performance parameter is also helpful to evaluate the accuracy of the ML-RA to predict the fastest route in the network.

TABLE V. SIMULATION DETAILS FOR ML-RA PERFORMANCE STUDY

Source router	Destination router	Main route (LSP)
R6	R2	LSP 4.5 is congested and LSP 3 is closed
R2	R5	LSP 6.1.2 is congested and LSP 3 is closed
R10	R7	LSP 12.5.6.7 is congested and LSP 3 is closed
R5	R3	LSP 3.4 is congested and LSP 6 is closed

When the traffic reaches a receiver beyond the standard delay requirements, it is considered as packet loss. This shows that the delay parameter alone is sufficient to describe how the traffic is being forwarded in the network. Conversely, the throughput performance parameter is chosen to verify the capacity rate of the traffic to reach the destination. Having higher throughput proves that the route computed by the RA offers better quality for the incoming traffic. For the performance simulation, a total of four S2D pairs are chosen, which are R6 to R2, R2 to R5, R10 to R7, and R5 to R3 as summarized in Table V.

The sample case studies the performance of ML-RA when the main route is congested, and one of the routes to reach the destination router is down. With that, for all the S2D pairs, there are only two routes available, which are the congested main route and ALT1. Based on the traffic and network conditions, the routes computed by ML-RA for R6 to R2, R2 to R5, R10 to R7, and R5 to R3 are as shown in Table VI. The computed route for this case for all S2D pairs is the ALT1. The result is as expected where ML-RA accurately rerouted the traffic to the normal ALT1 as the main route is congested. As has been explained, since RIPv2 always chooses the shortest path regardless of network congestion, RIPv2 still computed the main route for routing. In terms of delay, it is expected that as the load increases, the delay also increases due to a higher number of packets being forwarded to the destination. The delay comparisons between ML-RA and RIPv2 for all S2D pairs are shown in Fig. 6. The lower the delay of the ML-RA compared to RIPv2, the better the performance of the algorithm.

In Fig. 6, the solid line represents the delay for the route computed by ML-RA, while the dotted line represents the delay for RIPv2. The delay for both routes computed by ML-RA and RIPv2 increase as the offered load for all traffic increases from 20% to 100%. However, the delay for ML-RA is lower compared to RIPv2 for all traffic types. This is because ML-RA rerouted the traffic immediately via the ALT1. The results prove that, although the ALT1 comprises of a higher number of hops, the delays are lower compared to the congested main route with a lesser number of hops. For instance, the delay for R6 to R2, as shown in Fig. 6(a), even though the main route comprises of only one hop, but since the main route is congested, the delay for ALT1 which comprises of three hops contributes to a lower delay.

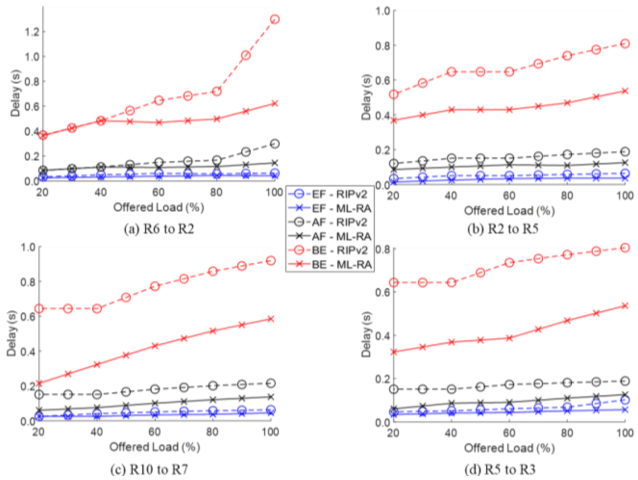

Fig. 6. Delay for the sample case.

TABLE VI. SIMULATION DETAILS FOR ML-RA PERFORMANCE STUDY

Source router	Destination router	Predicted fastest route by ML-RA
R6	R2	LSP 13.12.11.10
R2	R5	LSP 6.7.8.9
R10	R7	LSP 12.11.10.6.7
R5	R3	LSP 3.13.12

However, from 20% to 40% offered load for R6 to R2 S2D pair, the delay for BE traffic for both ML-RA and RIPv2 are almost in-line with each other as shown in Fig. 6(a). This is because the ratio of the number of hops between ALT1 and main route is almost triple, and the traffic from 20% to 40% offered load is too small to notice any delay difference. Nonetheless, beyond 40% offered load, the delay difference is more significant. At 100% offered load, the delay for BE traffic reduces from 1.2974 s to 0.6228 s for the route computed by ML-RA. While in Fig. 6(b) to 6(d), even though the difference in terms of number of hops between the main route and ALT1 is only by one additional hop, there is also a significant delay difference between the route computed by ML-RA and RIPv2. Since RIPv2 only computes the shortest available path in the network, the main route is chosen even though it will result in a higher delay. The delay comparison between ML-RA and RIPv2 for EF traffic for all pairs is not as significant as AF and BE traffic. This is because the data size and data rate for EF traffic are smaller compared to AF and BE.

The throughput is shown in Fig. 7. As opposed to delay, the higher the throughput offered by the ML-RA compared to RIPv2, the better the performance of the algorithm. Since the design parameters for the traffic is set at 20:40:40 traffic mixture ratio for EF, AF, and BE traffic, respectively, both AF and BE should theoretically have the same amount of throughput. However, due to the simulators' computation load, slight throughput fluctuation is expected. The throughput for all four S2D pairs shows a similar trend, which increases with increasing offered load from 20% to 100% for all traffic. However, since ML-RA successfully rerouted the traffic to the ALT1 for all four pairs, the throughput offered by the routes computed by ML-RA is higher compared to RIPv2.

For R6 to R2 S2D, the throughput for AF and BE traffic is almost in-line to each other from 20% to 60% offered load, as shown in Fig. 7. This is because the delay differences for both ML-RA and RIPv2 are not significant enough to produce a higher throughput difference. While for the rest of the S2D pairs where the number of hops difference between the congested main route and ALT1 is just addition by one, the delay difference between ML-RA and RIPv2 is even more significant which correlates to higher throughput difference.

The throughput and delay improvements for all S2D pairs are as shown in Table VII. In terms of throughput, the improvements for EF, AF, and BE traffic are up to 75.0%, 57.1%, and 57.1%, respectively. While in terms of delay, the improvements are up to 50.0%, 44.4%, and 44.4%, respectively. The results prove the superiority of ML-RA to predict and compute the fastest route in the network. For all S2D pairs, ML-RA learned from historical data that for this network condition, the fastest route would be ALT1. As a result, ML-RA intelligently computes ALT1 as the preferred route in the future when given the similar network condition.

From the results, the ML-RA is proven to be able to predict the route with the lowest delay and highest throughput, outperforming the RIPv2. ML-RA avoided the closed and congested routes and accurately computed the other faster alternative routes. Even when all routes are congested, ML-RA accurately predicts the fastest route in the network. Even though the route computed by ML-RA is the same as RIPv2, the goal of predicting the fastest path is met and maintained. Although only four S2D pairs are considered, the main and alternative routes cover the entire network, and almost all the LSRs in the network are involved. Thus, the ML-RA is expected to perform similarly for other S2D pairs.

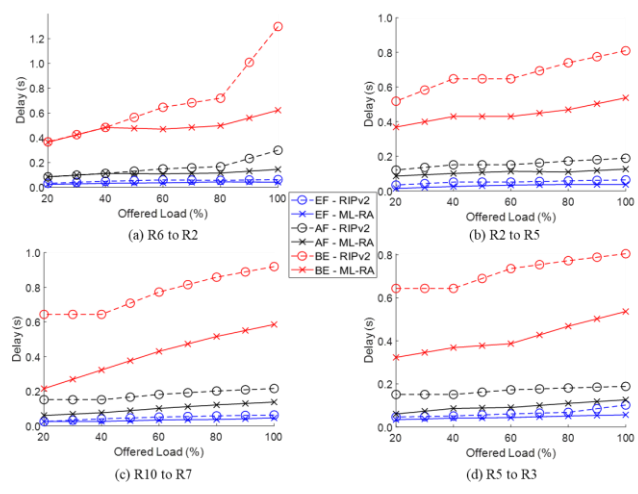

Fig. 7. Throughput for the sample case.

TABLE VII. THROUGHPUT AND DELAY IMPROVEMENT BY ML-RA

Source	Destination	Delay improvement (%)			Throughput improvement (%)		
		EF	*AF*	*BE*	*EF*	*AF*	*BE*
R6	R2	50.0	44.4	44.4	33.3	51.3	52.0
R2	R5	42.8	33.3	33.6	75.0	50.0	50.0
R10	R7	27.7	36.3	36.3	38.4	57.1	57.1
R5	R3	44.4	33.3	33.3	22.2	50.0	50.0

B. Limitations and Future Work of the Hybrid ML-IDS and ML-RA System

From the performance evaluations of the proposed ML-IDS and ML-RA, the system shows 100% accuracy in predicting the fastest route, as well as shows better performance against the RIPv2 in terms of delay and throughput. Despite the great performance of the proposed system, there may be several limitations that can be further improved in future research. For example, the proposed system relies on simulation data generated in GNS3 for dataset development. Note that the dataset was constructed using EF, AF, and BE traffic following the standard of VoIP, CCTV, and file transfers. However, this dataset may not accurately reflect the traffic patterns and network behavior of real-world environments. Thus, it may be advantageous to validate the performance of the system with real-world traffic data to confirm its practicality and effectiveness. Additional research on the influence of the parameters for the dataset on the performance of the system may also be investigated. The system's performance could also be assessed against different datasets in addition to evaluating its performance and scalability on larger and more complicated networks. Moreover, other ML techniques such as deep learning or reinforcement learning could be integrated to improve the accuracy and resilience of the proposed system. It is also advantageous to validate with real-world traffic data to confirm its practicality and effectiveness. Finally, the proposed system's implementation in a real-world environment could be performed to evaluate its applicability and feasibility in practical scenarios.

V. Conclusion

This study develops a hybrid ML-IDS and ML-RA algorithm to enhance the MPLS network's resiliency and improve traffic QoS. The proposed ML-IDS is a classification-based ML algorithm that learns the traffic pattern at the ingress router. Based on historical data, ML-IDS predicts and classifies the incoming traffic as normal or attack. The predicted attack traffic will be denied access to the network domain and discarded. While the predicted normal traffic will be queued according to their priority. This study prioritizes EF, AF, and BE traffic, which correspond to VoIP, CCTV, and data transfers, all having unique delay and bandwidth requirements. The GNS3 network setup is used for simulation and data collection. In particular, the LOIC traffic generator is used in the setup to simulate a DoS attack, while the OSTINATO traffic generator is used to generate the EF, AF, and BE traffic. Another OSTINATO traffic generator is used to burst background traffic in the network to simulate network congestion. The data in the network is collected as datasets. The output label for the ML-IDS and ML-RA datasets is the actual traffic type (normal or attack) and the actual delay for all available routes for all S2D pairs. The datasets are fed into MATLAB, which is then used to train classifiers for the ML-IDS algorithm and regressions for the ML-RA algorithm. For the performance evaluation, the ML-RA algorithm is compared to RIPv2. From the performance evaluations, the ML-RA shows 100% accuracy in predicting the fastest route in the network. During network congestion, the proposed ML outperforms the RIPv2 in terms of delay and throughput on average by 57.61% and 46.57%, respectively.

References

[1] S. Premkumar and V. Saminadan, "Performance evaluation of smart grid communication network using MPLS," Proc. 2017 IEEE Int. Conf. Commun. Signal Process. ICCSP 2017, vol. 2018-Janua, pp. 2116–2120, Feb. 2018, doi: 10.1109/ICCSP.2017.8286779.

[2] G. Andresini et al., "Multi-Channel Deep Feature Learning for Intrusion Detection," IEEE Access, vol. 8, pp. 53346–53359, 2020, doi: 10.1109/ACCESS.2020.2980937.

[3] M. A. Ridwan et al., "Applications of Machine Learning in Networking: A Survey of Current Issues and Future Challenges," IEEE Access, vol. 9, pp. 52523–52556, 2021, doi: 10.1109/ACCESS.2021.3069210.

[4] P. Mishra et al., "A detailed investigation and analysis of using machine learning techniques for intrusion detection," IEEE Commun. Surv. Tutorials, vol. 21, no. 1, pp. 686–728, Jan. 2019, doi: 10.1109/COMST.2018.2847722.

[5] K. Yu et al., "Machine learning driven network routing," 2019 6th Int. Conf. Syst. Informatics, ICSAI 2019, pp. 705–712, Nov. 2019, doi: 10.1109/ICSAI48974.2019.9010507.

[6] R. Alvizu et al., "Matheuristic With Machine-Learning-Based Prediction for Software-Defined Mobile Metro-Core Networks," J. Opt. Commun. Networking, Vol. 9, Issue 9, pp. D19-D30, vol. 9, no. 9, pp. D19–D30, Sep. 2017, doi: 10.1364/JOCN.9.000D19.

[7] G. Choudhury et al., "Two use cases of machine learning for SDN-enabled IP/optical networks: Traffic matrix prediction and optical path performance prediction [Invited]," J. Opt. Commun. Netw., vol. 10, no. 10, pp. D52–D62, Oct. 2018, doi: 10.1364/JOCN.10.000D52.

[8] Y. Nakayama et al., "Low-latency routing for fronthaul network: A Monte Carlo machine learning approach," IEEE Int. Conf. Commun., Jul. 2017, doi: 10.1109/ICC.2017.7996439.

[9] G. Stampa et al., "A Deep-Reinforcement Learning Approach for Software-Defined Networking Routing Optimization," Sep. 2017.

[10] T. Hendriks et al., "Q 2 -Routing : A Qos-aware Q-Routing algorithm for Wireless Ad Hoc Networks," Int. Conf. Wirel. Mob. Comput. Netw. Commun., vol. 2018-Octob, pp. 108–115, Dec. 2018, doi: 10.1109/WIMOB.2018.8589161.

[11] A. Martin et al., "Network resource allocation system for QoE-aware delivery of media services in 5G networks," IEEE Trans. Broadcast., vol. 64, no. 2, pp. 561–574, Jun. 2018, doi: 10.1109/TBC.2018.2828608.

[12] R. Boutaba et al., "A comprehensive survey on machine learning for networking: evolution, applications and research opportunities," J. Internet Serv. Appl., vol. 9, no. 1, pp. 1–99, Jun. 2018, doi: 10.1186/s13174-018-0087-2.

[13] F. Tang et al., "On Removing Routing Protocol from Future Wireless Networks: A Real-time Deep Learning Approach for Intelligent Traffic Control," IEEE Wirel. Commun., vol. 25, no. 1, pp. 154–160, Feb. 2018, doi: 10.1109/MWC.2017.1700244.

[14] L. Li et al., "Naïve Bayes classifier-assisted least loaded routing for circuit-switched networks," IEEE Access, vol. 7, pp. 11854–11867, 2019, doi: 10.1109/ACCESS.2019.2892063.

[15] H. Yao et al., "Machine learning aided load balance routing scheme considering queue utilization," IEEE Trans. Veh. Technol., vol. 68, no. 8, pp. 7987–7999, Aug. 2019, doi: 10.1109/TVT.2019.2921792.

[16] H. Yao et al., "A machine learning approach of load balance routing to support next-generation wireless networks," 2019 15th Int. Wirel. Commun. Mob. Comput. Conf. IWCMC 2019, pp. 1317–1322, Jun. 2019, doi: 10.1109/IWCMC.2019.8766546.

[17] M. Salani et al., "Routing and Spectrum Assignment Integrating Machine-Learning-Based QoT Estimation in Elastic Optical Networks," Proc. - IEEE INFOCOM, vol. 2019-April, pp. 1738–1746, Apr. 2019, doi: 10.1109/INFOCOM.2019.8737413.

[18] C. V. Murudkar and R. D. Gitlin, "Optimal-Capacity, Shortest Path Routing in Self-Organizing 5G Networks using Machine Learning," 2019 IEEE 20th Wirel. Microw. Technol. Conf. WAMICON 2019, Apr. 2019, doi: 10.1109/WAMICON.2019.8765434.

[19] B. Mao et al., “A Tensor Based Deep Learning Technique for Intelligent Packet Routing,” 2017 IEEE Glob. Commun. Conf. GLOBECOM 2017 - Proc., vol. 2018-Janua, pp. 1–6, Jul. 2017, doi: 10.1109/GLOCOM.2017.8254036.

[20] S. T. V. Pasca et al., “AMPS: Application aware multipath flow routing using machine learning in SDN,” 2017 23rd Natl. Conf. Commun. NCC 2017, Oct. 2017, doi: 10.1109/NCC.2017.8077095.

[21] V. Vashishth et al., “GMMR: A Gaussian mixture model based unsupervised machine learning approach for optimal routing in opportunistic IoT networks,” Comput. Commun., vol. 134, pp. 138–148, Jan. 2019, doi: 10.1016/j.comcom.2018.12.001.

[22] V. Vashishth et al., “A Machine Learning Approach Using Classifier Cascades for Optimal Routing in Opportunistic Internet of Things Networks,” Annu. IEEE Commun. Soc. Conf. Sensor, Mesh Ad Hoc Commun. Networks Work., vol. 2019-June, Jun. 2019, doi: 10.1109/SAHCN.2019.8824952.

[23] Z. M. Fadlullah et al., “Value Iteration Architecture Based Deep Learning for Intelligent Routing Exploiting Heterogeneous Computing Platforms,” IEEE Trans. Comput., vol. 68, no. 6, pp. 939–950, Jun. 2019, doi: 10.1109/TC.2018.2874483.

[24] S. Troia et al., “Machine-Learning-Assisted Routing in SDN-Based Optical Networks,” Eur. Conf. Opt. Commun. ECOC, vol. 2018-Septe, Nov. 2018, doi: 10.1109/ECOC.2018.8535437.

[25] N. A. Mohamed Radzi, “Recent Dynamic Bandwidth Allocation Algorithm Methods in Ethernet Passive Optical Network,” Int. J. New Comput. Archit. their Appl., vol. 4, no. 4, pp. 167–176, 2014, doi: 10.17781/P0016.

[26] H. Brink et al., Real-World Machine Learning. Manning, 2016.

[27] LAN/MAN Standards Committee of the IEEE Computer Society, “IEEE Standard for Ethernet,” IEEE Stand. Ethernet, 2018.

[28] “Cisco Community,” CISCO, 2010. https://community.cisco.com/t5/voice-over-ip/bandwidth-calculation/td-p/1649134 (accessed Jun. 28, 2022).

[29] S. U. Masruroh et al., “Performance evaluation of routing protocol RIPv2, OSPF, EIGRP with BGP,” in 2017 International Conference on Innovative and Creative Information Technology (ICITech), 2017, pp. 1–7, doi: 10.1109/INNOCIT.2017.8319134.