Quality-of-Service Performance Comparison: Machine Learning Regression and Classification-Based Predictive Routing Algorithm

M.A.Ridwan[a], N.A.M. Radzi[b], and F. Abdullah[b]

[a]Institute of Power Engineering, Universiti Tenaga Nasional, 43000 Kajang, Selangor, Malaysia
[b]Department of Electrical & Electronics Engineering, College of Engineering, Universiti Tenaga Nasional, 43000 Kajang, Selangor, Malaysia

Article History: *Do not touch during review process(xxxx)*

Abstract: The Internet network has evolved rapidly and by no means of slowing down. The complexity of the network is expected to grow exponentially, and with the high dependencies on Internet applications, there is a need to upgrade the current routing mechanism in the network. The conventional routing protocol that is based on the shortest path is no longer relevant. Recently, Machine Learning (ML) algorithms have become more prevalent in networking due to their ability to solve complex problems intelligently. This work proposes two ML predictive routing algorithms using regression and classification approaches to improve the Quality of Service in the network. Our simulation results show that the proposed regression-based routing improves the delay by up to 52% compared to the classification-based algorithm. Although the regression-based routing achieved better performance compared to the classification approach, it requires more input features to be trained. This work also discusses the pros and cons of both approaches.

Keywords: Machine Learning, Routing, Classification, Regression, Quality of Service

1. Introduction

In recent years, traffic in the network has grown exponentially with the increase of bandwidth and delay-sensitive applications, including voice over Internet Protocol (VoIP), 5G long-term evolutions, and on-demand videos. The traffics that are being forwarded by the existing network are becoming highly complex and challenging to fulfill the Quality-of-Service (QoS) for each network application. Furthermore, the number of users is expected to increase, especially with the emergence of the Internet of Things, high-speed connectivity offered by 5G, and cloud computing. With that, conventional algorithms or protocols are not able to handle the complexity of future network expansion, making it prone to network issues, including network congestion, ineffective resource management, poor QoS, or intrusion.

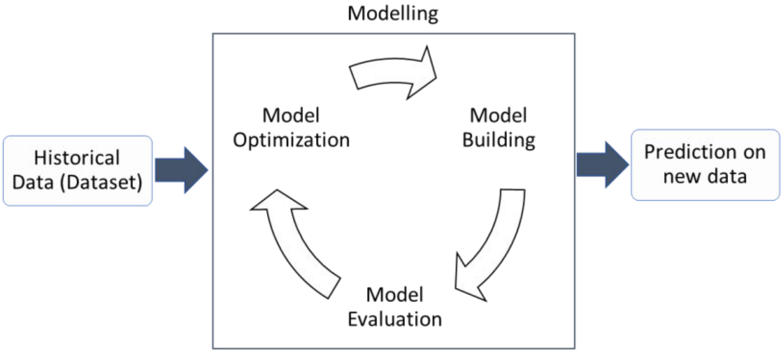
Machine Learning (ML) algorithms have recently captured the attention of researchers due to their superiority in addressing complex problems intelligently(Hooda *et al.* 2018). ML is one of the promising subsets of Artificial Intelligence technology that brings intelligence to various broad sectors. Considering the benefits of ML in solving complex issues, more advanced research is being conducted to ensure that the network subscriber's demand can be addressed(Mammeri 2019). Applications of ML include predicting the traffic performance(Morales *et al.* 2017), predicting the link quality by performing link evaluations using ML such as in(Bote-Lorenzo *et al.* 2018; Dudukovich&Papachristou 2018; Liu *et al.* 2020), performing efficient resource management as in (Shen *et al.* 2020; Tayyaba *et al.* 2020), and performing congestion control(Yuvaraj&Thangaraj 2019).

The basic workflow of ML algorithms is depicted in Fig. 1. To train the ML algorithm, historical data is collected to construct a dataset and fed into the ML platform to train the ML algorithms. Next, the ML platform will build the model, and the accuracy of the model is evaluated. If the accuracy is not promising, further optimization is required. This process is repeated until the accuracy of the algorithm converges. Finally, the trained ML algorithm is further validated on a new set of data to ensure the algorithm is not overfitting to the training dataset. The ML algorithm can be trained with a labeled dataset telling the machine what the right answers are, known as supervised learning. Algorithms such as Decision Tree (DT), Linear Regression (LR), and K-Nearest Neighbor (KNN) use such approach to perform regression or classifications. When dealing with labeled data, both input and desired output are known by the system. The supervised learning approach is commonly used when sufficient historical data are present (Brink *et al.* 2017).

Applications of ML include classification and regressions. The main difference between regression and classification algorithms is that regression algorithms are used to predict a continuous value. For instance, predicting the traffic volumes, network delays, or throughput. While in classification, the ML algorithm categorizes the data into different classes. For instance, predicting the incoming traffic as benign or intrusion traffic types. For regression type ML algorithms, the model will try to find the best fit line to predict the output accurately. In contrast, classification-type algorithms aim to find a best decision boundary, dividing the dataset into different classes.

Motivated by the superiority of ML algorithm to predict an output based on the historical data, this work aims to develop two ML-based approaches that is, regression-based (RgRoute) and classification-based (ClassRoute) algorithm to improve the routing mechanism in the network. Then, performance evaluation is conducted to compare the predictive accuracy and QoS improvement in the network.

Figure.1 Basic workflow of ML algorithm.



2.Related Works

From the literature, several predictive routing algorithms have been proposed to predict the network parameters such as traffic variation, traffic volume, traffic matrix, and bit error rate. The objective of ML is to learn from historical data or the environment and make prediction on the network parameters to improve the efficiency of the entire network system.

A study by (Alvizu*et al.* 2017) use Artificial Neural Network (ANN) for traffic load forecasting. This is to predict in advance the tidal traffic variation and to calculate the best resource allocation to reduce its energy consumption. The effectiveness of the ML-based dynamic routing scheme is proven with the results matched almost entirely the behavior of the network that performs optical routing reconfiguration. The proposed scheme yields an optimality gap above 3%, while the static-based routing scheme reduces the optimality gap below 0.2%.

(Choudhury *et al.* 2018)propose a hybrid ML model to predict the traffic volume for each of the traffic engineering tunnels at future time horizons followed by predicting the optical performance of new wavelengths in a multi-vendor environment. The genetic programming (GP) algorithm is chosen for the task of making a real-time prediction of traffic loads for each traffic engineering. After compiling all the available data for every optical path in the network, the RF model will predict the path performance. The path with the least Optical signal-to-noise ratio value is chosen to route the incoming traffics. The proposed scheme can improve the efficiency and reduce the cost by 9% as compared to non-ML-based scheme. This is due to the predictions made by the ML algorithm to avoid traffic loss by changing the IP layer topology before the traffic surge. Hence, the feasibility and efficiency are improved.

(Azzouni*et al.* 2017)introduced an ANN-based algorithm called NeuRoute to maximize the throughput at minimum cost for Software Defined Radio unicast dynamic routing. The traffic matrix estimator module is proposed to estimate the traffic matrix. Then, the traffic matrix predictor takes the fixed size set of archived traffic matrices and input to predict the traffic matrix at the next cycle. Finally, the traffic routing unit selects the optimal routes based on the predicted traffic matrix. The traffic matrix estimator continuously gathers data from the network and feed the traffic matrix predictor and traffic routing unit.The weights of ANN are then updated to improve the accuracy until it reaches the convergence point. The model successfully picks the near-optimal path learned from the model with an estimated error of 0.05% and execution time within 30 ms compared to the Baseline Heuristic approach that consumes 120 ms of execution time.

(Salani*et al.* 2019)proposed the integration of Random Forest (RF)-based estimation for routing and spectrum assignment for Quality of Transmission in the elastic optical network. All of the known network parameters such as traffic requests, the alternative configuration of routes, and modulation formats are acquired as input for the classifier. The output of the classifiers gives a probability that the light path configuration will satisfy a pre-determined threshold on the bit error rate measured at the receiver. The learning process is iterative, where new information on the adjacent channels are fed into the classifier. Compared with the margined analytical model, the proposed scheme achieves saving in the spectrum occupation up to 30%.

From the literature, the recent predictive routing algorithms exploited the ML algorithm to perform predictions on the network parameters and achieved outstanding performance. However, most works such as in(Alvizu*et* *al.*2017; Choudhury *et al.* 2018; Salani*et al.* 2019)only consider network congestion, while link failure scenarios are neglected in their system. Also, most recent works as presented in(Choudhury *et al.* 2018)only consider one type of traffic type. Since various network applications require different QoS treatment, it is essential to study the performance of an ML-based algorithm using different traffic types. Besides, the data generated in(Salani*et al.* 2019)to train the ML algorithms are primarily based on random traffics.

As opposed to our work, we consider both network congestion and link failure as part of the training of both algorithms. In addition, both algorithms are trained with three traffic types representing high, medium, and low traffic priority using a packets generator that models the actual traffic types. To the best of our knowledge, this is the first work that considers different traffic types with different network conditions to be implemented into the ML-based routing algorithm.

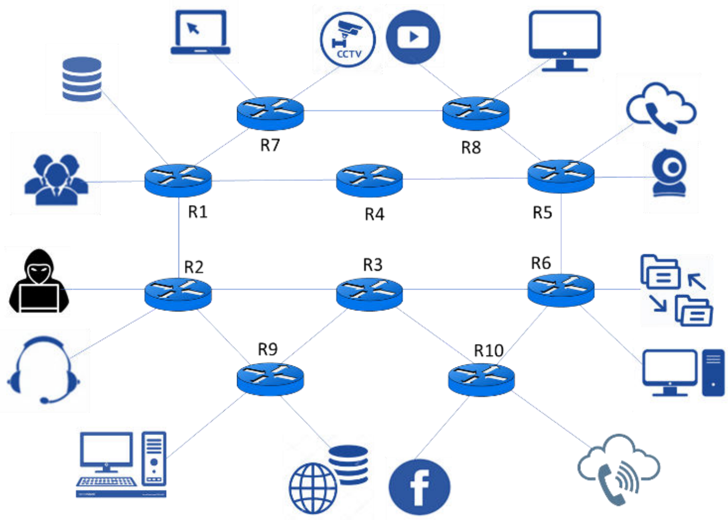
3.Routing Algorithm Development

A. Network Environment

Fig. 2 shows the network environment for the RgRoute and ClassRoute algorithm in the network system. The network comprises of eight edge routers, namely R1, R2, R5, R6, R7, R8, R9, and R10, that can be interchangeable as ingress or egress router. At the same time, R3 and R4 will represent normal LSR in the network domain. All the edge routers are connected to various three traffic types, including Expediated Forwarding (EF), Assured Forwarding (AF), and Best Effort (BE) traffics which corresponds to VoIP, Closed-Circuit TV (CCTV) and file transfers, respectively.

The network is constructed in four ring topologies—the main ring comprises of R1, R2, R3, R4, R5, and R6. The second ring consists of R1, R4, R5, R7, and R8, which act as node protection for R4. The third ring consists of R2, R3, and R9, while the final ring includes R3, R6, and R10. The third and fourth ring act as link protection between R2-R3 and R3-R6, respectively. However, for this network environment, all links are active, and it is up to the routing algorithm to compute the route for all traffic. For this work, as a proof-of-concept, there are total of four available routes available from the source to destination (S2D) router, namely main route, Alternative Route 1 (ALT1), Alternative Route 2 (ALT2), and Alternative Route 3 (ALT3). The purpose of having several alternative routes is to test the intelligence of the proposed ML-based routing algorithm to compute the fastest route. For this network, the main route for all SDR pairs is the shortest path, followed by ALT1, ALT2, and ALT3. For instance, from R1 to R5, the main route is R1-R4-R5. While R1-R7-R8-R5, R1-R2-R3-R6-R5 and R1-R2-R9-R3-R10-R6-R5 is the ALT1, ALT2, and ALT3, respectively for R1 to R5 S2D. The same routing assignments for the other S2D pairs in the network.

Figure.2 Network Environment



The simulation platform used for this work is the Graphical Network Simulator 3 (GNS3) and it is used by network engineers for various applications to emulate, configure, test, and perform troubleshooting in a virtual or real network. Aside from open source, GNS3 offers other advantages over other network simulators, including the ability to test and verify real-world deployment using real hardware emulator from various vendors. Emulation of routers such as the Cisco router is possible in GNS3, where it mimics the hardware of a device and runs the actual images. With that, the routers in the simulator function as it would in the real-world that eases the performance study. The network in Fig. 2 is constructed in GNS3 for simulation, data collection, and performance study.

Each of the edge routers are connected with a traffic generator called OSTINATO. OSTINATO is a packet and network traffic generator with an interactive graphical user interface (GUI) for network automation. The traffics generated can be sent using several streams simultaneously. In the OSTINATO GUI, user can configure different protocols at different rates and packet size.Configuration ofthe packetproperties up to layer 5 of the OSI model is also possible. For this work, the OSTINATO traffic generators are used to generate the VoIP, CCTV, and file transfer traffic in the network in GNS3. Another OSTINATO is used to generate network congestion by forwarding a continuous stream of traffic in the network. To simulate a link failure, the links in the network is purposely closed.

B. Framework of the proposed algorithm

Both RgRoute and ClassRoute algorithms are proposed to be implemented at the ingress router. This is to ensure that the path assignment decisions are already taken care of when the traffic enters the network domain. Besides, this will reduce the computation works on the other routers in the network. The framework for the RgRoute algorithm is as illustrated in Fig. 3. The objective of RgRoute is to predict the delay for all available routes in the network from the S2D pairs. Therefore, the RgRoute algorithm must learn the properties of the incoming traffic and the current network conditions. From the ingress port router, inputs including source and destination Internet Protocol (IP) address, traffic priority which is based on the Differentiated Services Code Point (DSCP), requested load, packets size, number of packets, data rates, and the available routes from S2D pair. In addition, two crucial inputs are considered from the network domain, including congestion rate and available routes that can be classified as open or broken links. With all the information gathered by the RgRoute, the output of the algorithm will be the predicted delays for all of the available routes. The route computation module will then choose the route with the lowest predicted delay and assigned it for incoming traffic. It is expected that the delay for all traffics is reduced as the RgRoute algorithm will compute the fastest route. Since delay is inversely proportional to throughput, the throughput is also expected to improve.

In contrast to RgRoute, the ClassRoute algorithm predicts the incoming traffic by classifying them into EF, AF, and BE traffic, as shown in Fig. 4. Only six inputs are required: source and destination IP address, requested load, packet size, number of packets, and data rates. The output of ClassRoute is the predicted incoming traffics according to their priorities. Since EF and AF are considered delay and throughput-sensitive traffic, they are given priority to utilize the least congested route. Therefore, the predicted EF and AF will be assigned with the fastest, available route in the network. In contrast, BE traffic, which is not delay-sensitive, will be assigned with other available routes regardless of the network conditions.

Figure.3 Network Framework of the proposed RgRoute Module

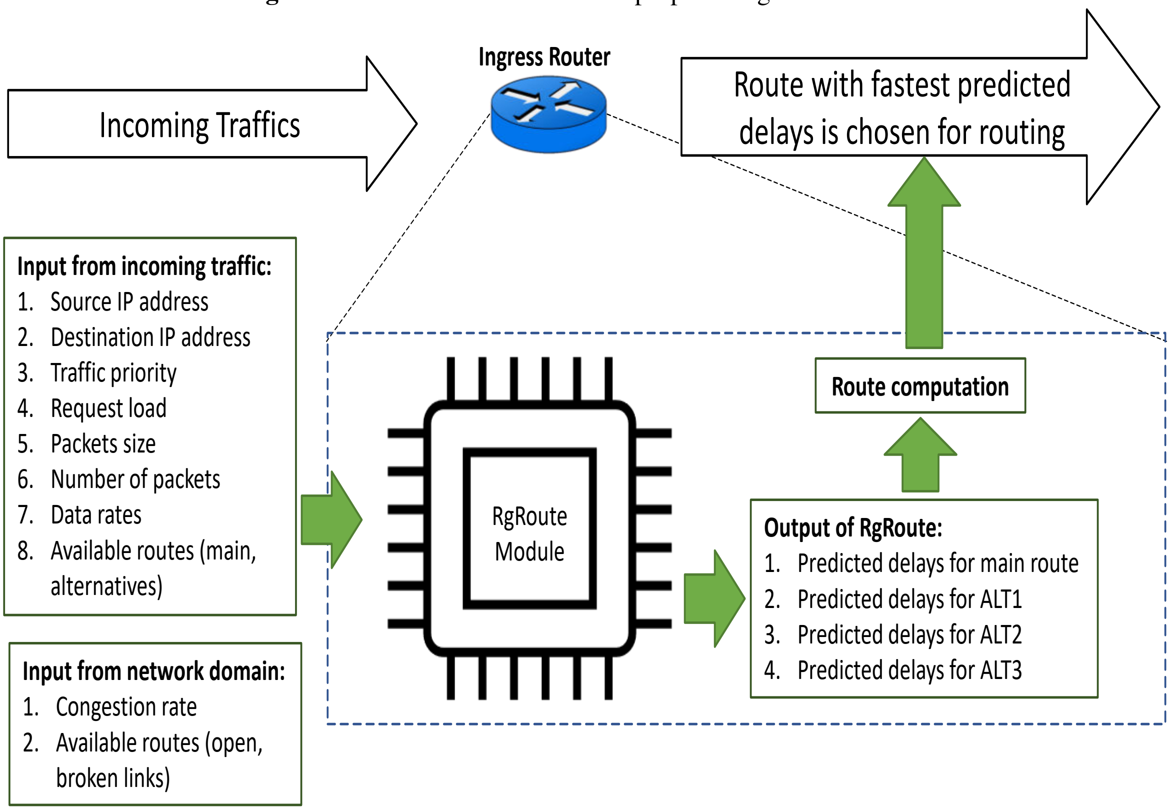
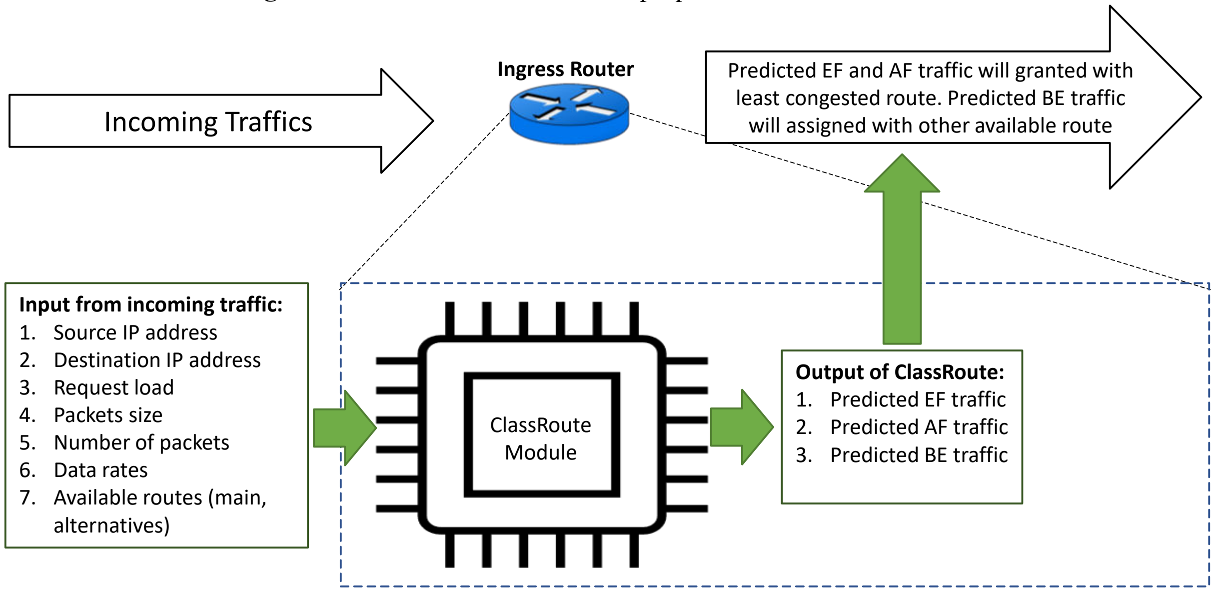


Figure.4 Network Framework of the proposed ClassRoute Module



C. RgRoute and ClassRoute dataset and algorithm development

To train both RgRoute and ClassRoute algorithms, the network is configured to run at different network parameters randomly. During the training phase, the source and destination IP address are set at random with three different traffic priorities namely, VoIP, CCTV and file transfer. The requested load begins at 20% with an increment of 10% up to 100% requested load. Our finding show that the maximum network capacity of OSTINATO in GNS3 is capped at 20 Mbps, with that 100% request load in equivalent to 20 Mbps traffic mixture of all VoIP, CCTV and file transfer traffics respectively. The packet size for VoIP length is fixed at 160 bytes, following the Cisco bandwidth calculator (“Voice Over IP - Per Call Bandwidth Consumption - Cisco” n.d.). While packets size for CCTV and file transfer are fixed at 500 bytes, for simplicity reason. Number of packets are up to 20,000 packets per session. The available routes include main route, ALT1, ALT2, and ALT3, depending on the network and link conditions. The greater the number of broken links, the lesser the available routes. For RgRoute algorithm, another two inputs are required from the network domain, that is, the congestion rate and available routes according to the S2D pairs. To build the RgRoute dataset, the output are the actual delays for all of the available routes. While for ClassRoute dataset, the output are the actual traffic types.

The simulations are run for half a million iterations with different network inputs, and all of the data collected are tabulated to construct both RgRoute and ClassRoute datasets. These datasets are then fed into the ML platform, particularly the Rapid Miner Studio for the ML algorithm development. Using Rapid Miner Studio’s auto model feature, a total of four regression-based ML algorithms are suggested, including Linear Regression (LRg), Fine, Medium, and Coarse DT. The performance comparisons between all four algorithms areevaluated in terms of their root mean square error (RMSE), training time, and prediction speed are compared, and the best performing algorithm is chosen for the RgRoute algorithm. While for the ClassRoute algorithm, six ML-based classifiers are suggested by Rapid Miner Studio,which are Linear Discriminant (LD), Quadratic Discriminant (QD), Ensemble Boosted Trees (EBT), DT, NB, and KNN. For the ClassRoute, the classifiers with the best performance parameters include accuracy, training time, and the most number correctly predicted traffic, are chosen.

4.Result and Discussion

For VoIP traffic, the ITU-Telecommunication G.114 Standardization recommends a maximum latency of 150 ms one-way latency(ITU-T 2009). While for CCTV traffic, the delay allowance is given to 400 ms(Jayant G *et al.* 2011; Uribe-Pérez *et al.* 2017). While BE traffic does not have rigorous timing requirements, however, for benchmarking purposes, the delay allowance is set to 2 seconds(Uribe-Pérez *et al.* 2017). With that, the packets are considered loss if the delays exceeded 150 ms, 400 ms, and 2 s for VoIP, CCTV, and file transfer, respectively. The performance of the proposed algorithms are considered superior when the delays are lower compared to each other.

For performance comparison, four S2D pairs are considered, which are R6-R2, R2-R5, R10-R7, and R6-R9. These S2D pairs are chosen because there are four available routes between the S2D pairs and involves almost all the links in the network. As for the network conditions, for all S2D pairs, both main route and ALT1 are closed, while ALT2 is congested. The delay for both RgRoute and ClassRoute are compared based on the network conditions.

A. Performance comparison between different regression models for RgRoute algorithm

After the RgRoute dataset is fed in RapidMiner Studio, the performance comparison between four regression models, including LRg, Fine, Medium, and Coarse DT, is tabulated in Table 1. Table 1 shows the performance of medium DT and coarse DT with the exact prediction speed, but coarse DT has a slight edge in RMSE and only 0.0838 s difference in training time compared to medium DT. This is expected since coarse DT consists of a few large leaves and has the lowest flexibility compared to the fine and medium DT algorithms. This allows Coarse DT to have better RMSE compared to the rest. For that reason, coarse DT is chosen for the proposed RgRoute.

B. Output of RgRoute algorithm

The output of the RgRoute, which is the predicted delay is as shown in Table 2 and 3. The algorithm is trained so that when the route is closed, the predicted delay is fixed at 1000 s. This is deemed logical as the route computation module will not choose the route with a 1000 s predicted delay. Since the Main Route and ALT1 are closed, RgRoute has accurately predicted the delay for both routes as shown in Table 2. ALT2 and ALT3, on the other hand, are functioning normally. However, since only ALT2 is congested, RgRoute has predicted that the delay for ALT2 is higher than ALT3, as shown in Table 3. For that, the RgRoutehas proven to have good accuracy for this network condition. For the RgRoute algorithm, all traffic, including EF, AF, and BE, are assigned to utilize the predicted fastest route in the network.

Table.1. Performance comparison between different regression algorithms

	RMSE	Training time	Prediction speed (observation/sec)
LRg	0.19105	12.786	420,000
Fine DT	0.10534	4.3137	1,400,000
Medium DT	0.10479	3.8649	1,500,000
Coarse DT	0.10451	3.9487	1,500,000

Table.2. Predicted delays for main route and ALT1

	Predicted Delays (s)	
	Main Route	ALT 1
S2D pairs	EF, AF, BE	EF, AF, BE
R6-R2	1000	1000
R2-R5	1000	1000
R10-R7	1000	1000
R6-R9	1000	1000

Table.3. Predicted delays for main route and ALT2 and ALT3

	Predicted Delays (s)					
	ALT 2			ALT 3		
S2D pairs	EF	AF	BE	EF	AF	BE
R6-R2	0.071	0.299	1.297	0.046	0.150	0.553
R2-R5	0.102	0.216	0.923	0.072	0.137	0.585
R10-R7	0.071	0.189	0.805	0.046	0.151	0.644
R6-R9	0.102	0.151	0.644	0.051	0.137	0.585

C. Performance comparison between different classifiers models for ClassRoute algorithm

For the ClassRoute algorithm, the dataset is also fed into RapidMiner Studio to train several ML-based classifiers, including LD, QD, EBT, DT, NB, and KNN. Since the ClassRoute algorithm is a multiclass classification algorithm, only accuracy and training time are the considered performance parameters. The accuracy and training time for the classifiers are tabulated in Table 4. DT, KNN, and EBT offer the best accuracy. However, LD and QD offer better training time, which corresponds to the lower complexity of the model. DT, on the other hand, ranked third in terms of training time.

To further evaluate the classifiers, the developed ML classifiers in RapidMiner Studio predict all three traffic types with different offered loads from 20% to 100% with 20% step size. The results are tabulated in Table 5, where the green shaded cells are the correctly predicted traffics, while the red cells are the wrongly predicted traffics. Amongst all classifiers, only DT offers the most accurate predicted traffic. Except for BE traffic at 20% offered load, it is falsely classified as AF traffic. This is because at 20%, the BE traffic is probably insignificant enough to have a definite classification difference with AF traffic, which prone to classification error. In addition to that, the rest of the classifiers fail to predict the EF traffics accurately. For that reason, DT is considered for the ClassRoute, and the performance is compared with RgRoute.

When two or more traffics with the same priority and bandwidth requirement arrived at the exact moment, that traffic shares the transmission service proportionally according to assigned weights(Addeo*et al.* 2014). For this work, the assigned weight is the data size. With that, Strict Priority Queue (SPQ) mechanism will grant the traffic with greater data size to be transmitted first, followed by the next smaller data size of the same priority (Addeo*et al.* 2014; Jayant G *et al.* 2011), as illustrated in Fig. 5. Since at 20% offered load, the actual BE traffic is falsely predicted as AF traffics by the ClassRoute algorithm, and that the data size of BE is greater than AF, SPQ will allow the falsely predicted BE as AF traffic to transmit first followed by actual AF traffics. For that, the actual AF traffic will suffer from a higher delay due to queueing.

Figure.5 Illustration of queuing in ClassRoute algorithm at 20 % offered load

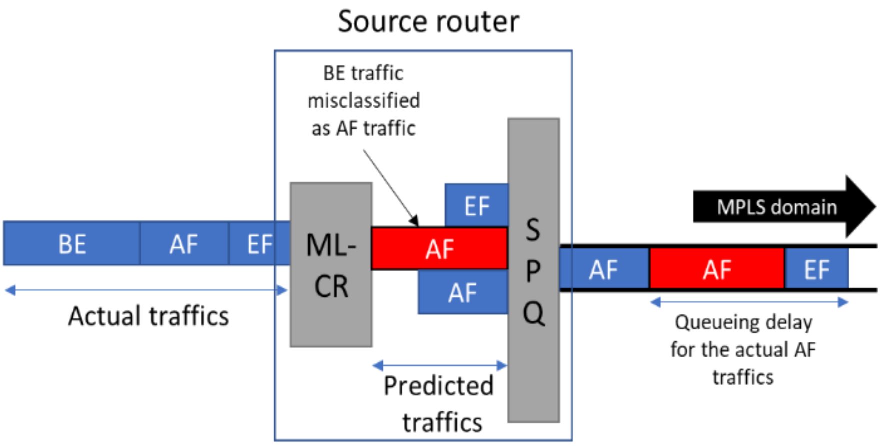


Table.4. Performance comparison for ML classifiers for ClassRoute

	Accuracy	Training time (s)
DT	98.4%	0.553
LD	90.1%	0.327
QD	96.4%	0.458
NB	92.3%	117.060
KNN	98.0%	0.762
EBT	98.5%	7.671

On top of that, ClassRoute computes the route for the traffics as per their priorities. With that, the EF and AF traffics are routed via the fastest route, whereas the BE traffic is routed via any remaining available routes. For performance comparison purposes, the BE traffic is routed via the shortest path between the S2D pair. EF and AF traffic is routed via the computed routes similar toRgRoute since it has accurately chosen the fastest route in the network. The same network condition is used for performance study.

Table.5. Actual versus predicted traffic using different ML classifiers

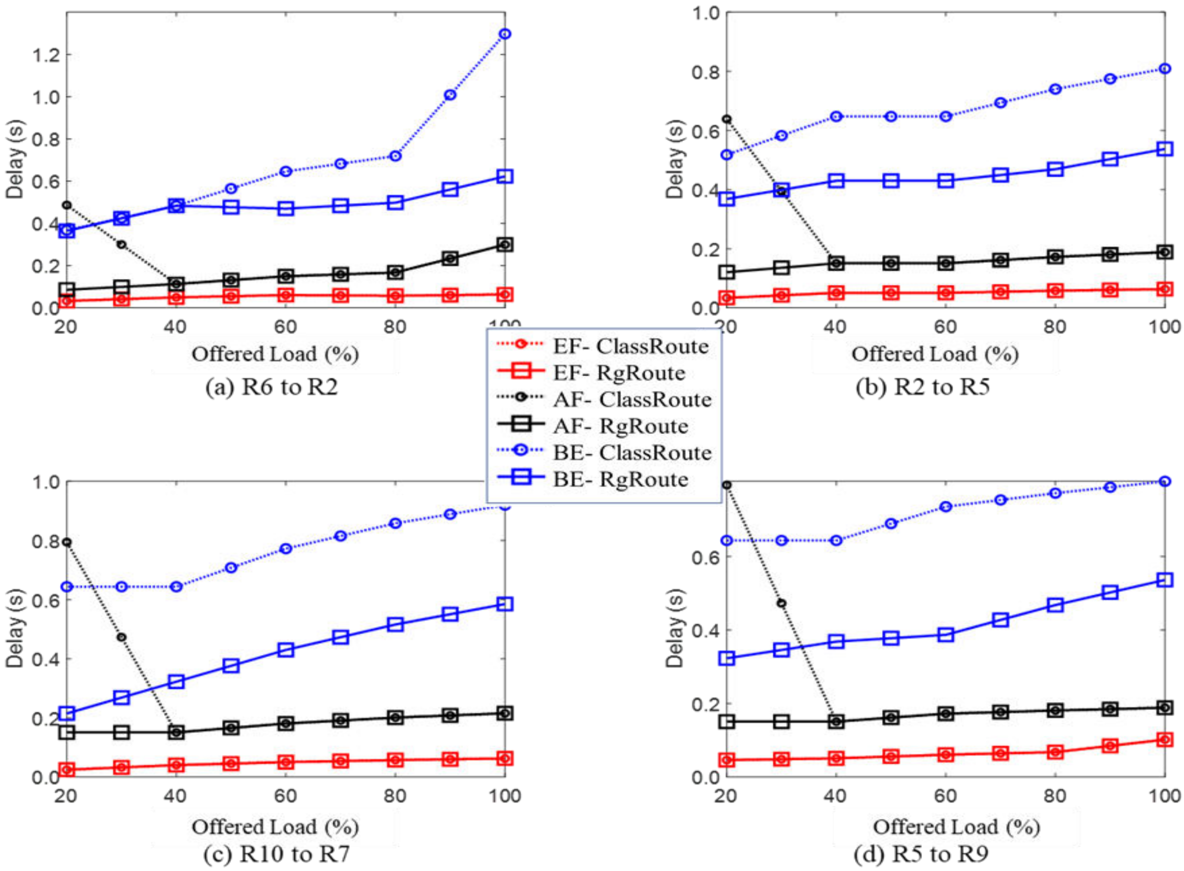
		Predicted Traffic					
Load	**Actual traffic**	**DT**	**LD**	**QD**	**NB**	**KNN**	**EBT**
20%	EF	EF	EF	AF	AF	AF	AF
40%	EF	EF	AF	AF	AF	AF	AF
60%	EF	EF	AF	AF	AF	AF	AF
80%	EF	EF	AF	AF	BE	AF	AF
100%	EF	EF	AF	AF	BE	AF	AF
20%	AF	AF	AF	AF	AF	AF	AF
40%	AF	AF	AF	AF	AF	AF	AF
60%	AF	AF	AF	AF	AF	AF	AF
80%	AF	AF	AF	AF	BE	AF	BE
100%	AF	AF	AF	AF	BE	BE	BE
20%	BE	AF	BE	BE	AF	BE	BE
40%	BE	BE	BE	BE	BE	BE	BE
60%	BE	BE	BE	BE	BE	BE	BE
80%	BE	BE	BE	BE	BE	BE	BE
100%	BE	BE	BE	BE	BE	BE	BE

D. Delay results comparison between RgRoute and ClassRoute

The delay comparison between RgRoute and ClassRoute algorithms for all four S2D pairs is as shown in Fig. 6. Two noticeable observations can be spotted from the result: (1) the delay of BE traffics from 20% to 100% offered load, and (2) the delay of AF traffics at 20% offered load for both RgRoute and ClassRoute algorithm. For BE traffics, for all S2D pairs, the delay for ClassRoute is higher than RgRoute. For instance, BE traffics at 100% offered load for R6 to R2 as shown in Fig 6a. The delay for ClassRoute is 1.2974s, as opposed to RgRoute with only 0.6228s. The delay improvement is 52.0%. For R2 to R5 as shown in Fig. 6b, the delay is 0.7399s, and 0.4690s, for ClassRoute and RgRoute, respectively, which shows an improvement of 33.6%. Then, for R10 to R7 as shown in Fig. 6c, RgRoute offers delay reduction by 0.3343s, which is an improvement of 36.3%. Finally, for R5 to R9 as shown in Fig. 6d, RgRoute outperforms the ClassRoute with a 33.3% delay improvement. The delay difference between RgRoute and ClassRoute is because of the routes computedby both ML-based routing algorithm is different when it comes to traffics with lower priority. RgRoute routes all traffic via the fastest route, while ClassRoute prioritized EF and AF traffic. This is because BE traffic has no stringent delay requirements that require to be routed via the fastest route. RgRoute, on the other hand, routes all traffic via the predicted fastest route in the network. That is the reason why there are noticeable delays between RgRoute and ClassRoute for BE traffics.

As mentioned earlier, ClassRoute misclassified the BE traffics as AF traffics at 20% offered load. Since BE packets size is larger than AF packets, the BE traffics that are wrongly classified as AF traffics are transmitted first by the SPQ. The queuing illustration for ClassRoute at 20% offered load is as depicted in Figure 5. At 20% offered load, the delay of AF traffics is significantly higher than the BE traffics. The queuing delays suffered by the actual AF traffic at 20 % offered load for R6 to R2, R2 to R5, R10 to R7, and R5 to R9 are 0.4009s, 0.5519s, 0.6684s, and 0.6899s, respectively. These queuing delays have caused the actual AF traffics to exceed the maximum delay allowance of CCTV traffics, which is 0.4000s. With that, at 20% offered load, the performance of CCTV may be degraded due to the inevitable queueing delays due to classification error by the ClassRoute.

Figure.6 Delay performance comparison between RgRoute and ClassRoute for EF, AF, and BE traffic



5.Conclusion and Future Works

This work studies the delay comparison performance between regression and classification-based routing algorithms in the network, namely RgRoute and ClassRoute, respectively. ClassRoutealgorithm is prone to classification error; however, it can be advantageous for network providers to predict the incoming traffics beforehand and manage the network resource accordingly. Besides, only six inputs are required to train the ClassRoute algorithm, which may signify lesser computational works. However, to further improve the ClassRoute algorithm, a safety mechanism to overcome the classification error must be implemented to avoid the degradation of delay-sensitive services.

In contrast with RgRoute, the queueing of the traffics is based on the DSCP at the IP header. Then, based on the historical data, the RgRoute will predict the delay of the traffics based on the network and traffics condition. Since all traffics are routed via the fastest predict route, there will be no delay-sensitive traffics mistakenly route via the congested link. While the RgRoute algorithm shows promising delay improvement, it takes eight inputs from the incoming traffic and two inputs from the network domain. This may incur higher computing works due to higher network input to be processed.

The result brings back to the dilemma of implementing an ML-based routing mechanism in the network. The network providers are required to look for the best trade-off between accuracy and computing load. For our future works, we will improve on the RgClass accuracy and cascade together with RgRoute to form a hybrid ML-based routing algorithm for further enhancement of routing capability in the network.

References (APA)

Addeo, C., Cazzaniga, G., Crescentini, R., & Valente, F. (2014). On QoS Mechanism Profiling in MPLS-TP Transport Networks. *Bell Labs Technical Journal*, **18**(4), 3–17.

Alvizu, R., Troia, S., Maier, G., &Pattavina, A. (2017). Matheuristic with machine-learning-based prediction for software-defined mobile metro-core networks. *Journal of Optical Communications and Networking*, **9**(9), D19–D30.

Azzouni, A., Boutaba, R., &Pujolle, G. (2017). NeuRoute : Predictive Dynamic Routing for Software-Defined Networks. *Networking and Internet Architecture, ArXiv*.

Bote-Lorenzo, M. L., Gómez-Sánchez, E., Mediavilla-Pastor, C., & Asensio-Pérez, J. I. (2018). Online machine learning algorithms to predict link quality in community wireless mesh networks. *Computer Networks*, **132**, 68–80.

Brink, H., Richards, J. W., &Fetherolf, M. (2017). *Real-world Machine Learning*, Manning.

Choudhury, G., Lynch, D., Thakur, G., &Tse, S. (2018). Two Use Cases of Machine Learning for SDN-Enabled IP / Optical Networks : Traffic Matrix Prediction and Optical Path Performance Prediction [Invited]. *IEEE/OSA Journal of Optical Communications and Networking*, **10**(10), D52–D62.

Dudukovich, R., &Papachristou, C. (2018). Delay Tolerant Network Routing as a Machine Learning Classification Problem. In *2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, IEEE, pp. 96–103.

Hooda, N., Bawa, S., & Rana, P. S. (2018). MCTOPE Ensemble Machine Learning Framework: A Case Study of Routing Protocol Prediction. In *Proceedings on 2018 IEEE 3rd International Conference on Computing, Communication and Security, ICCCS 2018*, IEEE, pp. 92–99.

ITU-T. (2009). *G.114 Amendment 2: One-way transmission time Amendment 2: New Appendix III – Delay variation on unshared access lines*. Retrieved from https://www.itu.int/rec/T-REC-G.114-200911-I!Amd2/en

Jayant G, D., Eunyong, K., & Marina, T. (2011). Differentiated Services QoS in Smart Grid Communication Networks. *Bell Labs Technical Journal*, **16**(3), 61–81.

Liu, L., Yin, B., Member, S., Zhang, S., & Member, S. (2020). Deep Learning Meets Wireless Network Optimization: Identify Critical Links. In *IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING*, IEEE, pp. 167–180.

Mammeri, Z. (2019). Reinforcement learning based routing in networks: Review and classification of approaches. *IEEE Access*, **7**, 55916–55950.

Morales, F., Ruiz, M., Gifre, L., Contreras, L. M., López, V., & Velasco, L. (2017). Virtual network topology adaptability based on data analytics for traffic prediction. *Journal of Optical Communications and Networking*, **9**(1), 35–45.

Salani, M., Rottondi, C., &Tornatore, M. (2019). Routing and Spectrum Assignment Integrating Machine-Learning-Based QoT Estimation in Elastic Optical Networks. In *Proceedings - IEEE INFOCOM*, pp. 1738–1746.

Shen, Y., Shi, Y., Zhang, J., &Letaief, K. B. (2020). LORM: Learning to Optimize for Resource Management in Wireless Networks with Few Training Samples. *IEEE Transactions on Wireless Communications*, **19**(1), 665–679.

Tayyaba, S. K., Khattak, H. A., Almogren, A., … Guizani, M. (2020). 5G vehicular network resource management for improving radio access through machine learning. *IEEE Access*, **8**, 6792–6800.

Uribe-Pérez, N., Angulo, I., de La Vega, D., Arzuaga, T., Fernández, I., &Arrinda, A. (2017). Smart grid applications for a practical implementation of IP over narrowband power line communications. *Energies*, **10**(11), 1–16.

Voice Over IP - Per Call Bandwidth Consumption - Cisco. (n.d.). Retrieved July 23, 2020, from https://www.cisco.com/c/en/us/support/docs/voice/voice-quality/7934-bwidth-consume.html

Yuvaraj, N., &Thangaraj, P. (2019). Machine learning based adaptive congestion window adjustment for Congestion Aware Routing in Cross Layer Approach Handling of Wireless Mesh Network. *Cluster Computing*, **22**, 9929–9939.