Jurnal Mantik, 6 (3) (2022) 2907-2917
Published by:Institute of Computer Science (IOCS)
Jurnal Mantik
Journal homepage: www.iocscience.org/ejournal/index.php/mantik/index

Implementation of nave bayes algorithm for routing recommendations in software defined network (sdn) simulation

Yuggo Afrianto[1], Indriyawati[2], Ritzkal[3], Ade Hendri Hendrawan[4]
[1,2,3,4]Universitas Ibn Khaldun Bogor, Indonesia

ARTICLE INFO

Article history:

Received Aug 26, 2022
Revised Sep 20, 2022
Accepted Okt 07, 2022

Keywords:

Naïve Bayes algorithm
Controller ryu
SDN

ABSTRACT

Routing determines the route or path taken by packets and flows from sender to receiver. Software Define Networking (SDN) is a new concept for designing, building, and managing computer networks by separating controllers and devices. However, the development of a network that is currently increasing makes a computer network require a decision support mechanism, especially in determining the route or path taken by packets and flows from sender to receiver. This study aims to make a routing decision recommendation system using the Naïve Bayes algorithm using a QoS dataset including throughput, packet loss, delay, and jitter with Tiphon standards and QoS requirements for several applications on the network. Openflow protocol, Ryu controller, and simulation-based implementation using mininet were used in this study. The performance results show that the Naive Bayes algorithm can provide routing recommendations for www Browsing, Video Streaming HD, and VoIP Telephony applications with an accuracy of 91% and an average precision of 95%.

Corresponding Author:

Yuggo Afrianto,
Universitas Ibn Khaldun Bogor,
Jl. Sholeh Iskandar, RT.01/RW.10, Kedungbadak, Kec. Tanah Sereal, Kota Bogor, Jawa Barat 16162
Email: yuggo@uika-bogor.ac.id

1. INTRODUCTION

Routing is determining the route or path taken by the packet and flows from the sender to the receiver [1]. With the development of the existing network, there are still many network infrastructures that use traditional networks. The distribution of the control plane is on each device [2]. This makes network development and maintenance take a very long time to configure. Software-Define Networking (SDN) is a new approach concept for designing, building, and managing computer networks by separating controllers in network devices [3] . One of the advantages of SDN is the increase in network performance, which is better than traditional networks. Ryu Controller is an open, software-defined SDN Controller designed to increase network agility by efficiently managing and customizing how paths are handled and for the separation between controllers and devices performed by the OpenFlow protocol [4].

Openflow is the first standard communication interface defined between the control layer and the forwarding layer on SDN [5]. The OpenFlow protocol can be used as a technology from SDN that is useful for separating controllers and devices with the aim of sending data more efficiently [6], with SDN and OpenFlow can create network logic

that is programmed to determine the path (Routing). Recommendation scenarios that can be used in the process of exchanging Data or information, where the path can be adjusted to the type of service, including video streaming, social media, e-commerce, and downloads. Naïve Bayes is a simple algorithm based on Bayes theory which assumes that the attributes of the classification are independent [7] , and the nature of this classification is the idea to get Routing recommendations on computer networks.
Several studies that have been carried out [8] investigated the smart routing problem in SDN, by utilizing a new data flow classification method. Combining various machine learning algorithms, a data flow classification method called MACCA2-RF&RF is proposed to identify data flow categories and obtain QoS requirements. [9] predicting user behavior using an artificial neural network (ANN) shows accurate results in predicting the bandwidth used by users, as well as providing an overview of user behavior patterns in the next few days, whereas for streaming video, a bandwidth of 2.837 KBps is required, social media of 2.837 KBps is required. 996 KBps, e-commerce at 875 KBps, and downloads at 678 KBps. Some of the research above can be a solution for routing problems in SDN networks.

This study uses a nave Bayes algorithm to determine routing recommendations on the network based on the characteristics of the type of data used by the user, which is implemented on a virtualized SDN network using mininet and Ryu controller for static routing mechanisms so that it can create a decision support system in network configuration management that is faster and more effective.

2. METHOD

The research method is a framework for carrying out an action or framework of thinking to develop an idea that is directed and related to the aims and objectives [10] .

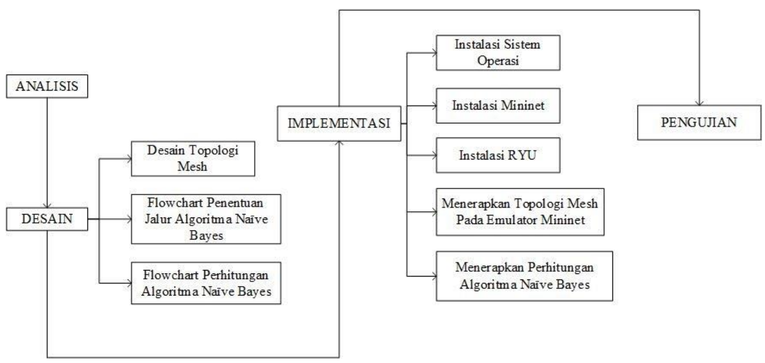

Figure 1. Stages of Research Methods

The analysis phase is carried out to design the system. The analysis is divided into 2, namely the needs analysis carried out for some hardware and software and the analysis of how it works, which is to explain how the system works in a flowchart. The design stage is to discuss the topology description, path determination, and the calculation of the Naïve Bayes algorithm to facilitate the network concept in this study. The implementation stage is to install some software. First, install the Ubuntu Linux operating system for the implementation of the Naïve Bayes algorithm routing calculations to get routing recommendations based on the QoS value indicator metric. The second install the mininet emulator to create a network topology by implementing a mesh topology, and the third install RYU as a logic control plane which will enable the static routing function. The last stage is testing conducted for Routing based on the recommendation results of the calculation of the Naïve Bayes algorithm in static Routing.

2.1 Analysis

The implementation of a routing network simulation system with recommendations for determining paths based on the type of service in a simulation-based SDN network requires an analysis of software requirements and the required QoS metric parameters for several services on the network. The software used in this study are 1). Linux Ubuntu for operating system used in carrying out this research; 2). Mininet A network emulator used to simulate a network of SDN controllers, switches, and hosts; 3). Ryu Controller to run network topology on mininet emulator; 4). Wireshark for network data retrieval. QoS parameter tables based on the TIPHON standard to support the implementation of the Naïve Bayes algorithm as a recommendation for determining paths in SDN network simulations. The TIPHON standard is a standard for assessing QoS parameters issued by the European Telecommunications Standards Institute (ETSI) [11] , with several categories of QoS metric standards shown in Tables 1 Standards of QoS metric requirements for several services based on transmission speed measurements UKE (Office of Electronic Communications) data in the Quality of Service Regulation Manual by the ITU Telecommunication Development Bureau [12] are shown in Table 2.

Table 1. Standardization Delay (Tiphon)

Category Delay	Delay	Troughput	Jitter	Packetloss	Index
Very good	< 150 ms	>2,1 Mbps	0 ms	0%	4
Good	150 s/d 300 ms	700-1200 kbps	0 s/d 75 ms	3%	3
Currently	300 s/d 450 ms	338-700 kbps	75 s/d 125 ms	15%	2
Bad	>450 ms	0-338 kbps	125 s/d 225 ms	25%	1

Table 2. Application Data Transmission Needs (Uke)

Aplication	Throughput not less than	Delay not less than
WWW browsing	1 Mbps	200 ms
Video Streaming HD	6 Mbps	200 ms
Voip Telephony	64 Kbps	150 ms

2.2 Design

The topology design is made for special Routing using the recommendations from the Naïve Bayes algorithm on the SDN network. The topology used is a mesh topology using 4 switches, 2 hosts, and 1 controller. The mesh topology will be applied to the mininet emulator. Link switch 1 is connected to link switch 2, switch 3 is to switch 4, link switch 2 is to switch 1, switch 4 is to switch 3, link switch 4 is to switch 3, switch 2 is to switch 1, switch 3 is to switch 1, switch 4 is to switch 2, switch 1 with host 1, and switch 4 with host 2. Figure 3 flowchart to get the results of the path recommendations as follows.

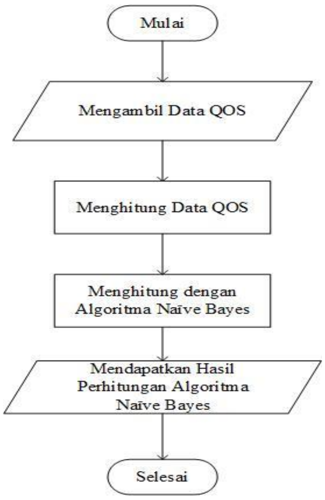

Figure 3. Flowchart of Path Determination of the Naïve Bayes Algorithm

Based on Figure 3 the steps in the calculation of the Naïve Bayes algorithm are first to take QoS data, the data is retrieved using wireshark by capturing network transaction data, internal router data. Second, calculate the QOS data according to the TIPHON standard to get the dataset and its categories. The third is to calculate the Naïve Bayes algorithm and get the results of routing recommendations for several network applications.

2.3 Implementation

1) Operating System Installation

The research process used the operating system installation stage. This study uses the Linux Ubuntu 20.04 operating system [13].

2) Implementation of Mesh Topology on Mininet Emulator

Mininet serves to create an SDN network topology emulator using the Openflow protocol [14]. At this stage, the researcher applies a mesh topology so that it can answer the problems that have been formulated for the computer network routing scenario. Figure 4 execution of the SDN-Openflow network simulation that has been created using the mininet emulator.

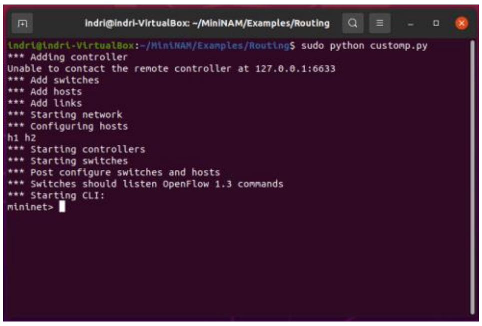

Figure 4. SDN-Openflow Network Simulation Execution

3) RYU Installation

At this stage, the RYU controller research to run static routing using the rest_router module [15], which can be accessed on url:https://github.com/faucetsdn/ryu/blob/master/ryu/app/rest_router.py

3. RESULT AND DISCUSSION

3.1 Nave Bayes Algorithm Routing Recommendations

The calculation of the Naïve Bayes algorithm used to obtain the results of the tracking recommendations can be explained in Figure 5, as follows.

Figure 5. Flowchart of the Calculation of the Naïve Bayes Algorithm

Calculation of the Naïve Bayes algorithm is done by calculating QoS to obtain training and test data. There are 4 QoS indicator metrics used, namely: throughput, packet loss, delay, and jitter. The probability construction of training and test data categories can be calculated using Formula 1, as follows:

$$\text{Category} = \left|\frac{J_1+J_2+J_3+J_4}{4}\right| \qquad (1)$$

Where:
J_1 = throughput quality index value based on TIPHON standardization
J_2 = Packetloss quality index value based on TIPHON standardization
J_3 = Delay quality index value based on TIPHON standard
J_4 = jitter quality index value based on TIPHON standardization
Category value if: 4 = Very Good, 3 = Good, 2 = Medium, 1 Poor.
From the formula above, it is obtained that the dataset construction has been given a category label based on the aggregate of each QoS indicator value, as shown in Tables 3 and 4.

Table 3.Training Data

No.	Troughput (kbps)	Packetloss %	Delay (ms)	Jitter (ms)	Category
1	0,0010	0	30	0,01	Good
2	0,7183	0	80	0,005	Good
3	0,0027	0	20	0,001	Good
4	0,0023	0	20	4,91	Good
5	0,0012	0	20	5,60	Good
6	0,0108	0	700	0,02	Currently
7	0,6178	0,21	40	0,007	Good
8	0,0018	0	20	0,001	Good
9	0,0021	0	40	0,0005	Good
10	0,0032	0	20	2,96	Good

11	0,0025	0	30	2,06	Good
12	0,0010	0	4	1,64	Good
13	0,0011	15	380	220	Bad
14	0,0019	25	500	300	Bad
15	1300	0	30	0	Very good
16	1545	0.03	40	0	Very good
17	1634	0	30	0	Very good
18	0,0013	26	466	300	Bad
19	1450	2	90	0	Very good
20	0,0033	33	370	226	Bad
21	755	26	240	0,02	Good
22	1400	3	466	0,01	Good
23	0,0018	0,21	20	60	Good
24	0,0020	0	225	0,08	Good
25	348	25	268	77	Currently
26	801	25	20	76	Currently
27	1235	18	195	0,001	Good
28	830	4	459	80	Currently

After getting the training data, it is continued by calculating the test data, with the results shown in Table 4.

Table 4. Training Data

Data to	Troughput	Packetloss	Delay	Jitter	Current Category	Category Naive Bayes Predictions
1	1500	0,03	100	0	Very good	?
2	339	3	0	0,007	Good	?
3	1234	0	20	0	Very good	?
4	1295	5	155	0,005	Good	?
5	755	16	185	0,0032	Good	?
6	0,0021	1	387	295	Currently	?
7	117	26	70	77	Currently	?
8	0,0013	5	238	89	Currently	?
9	0,515	25	470	227	Bad	?
10	0,044	16	268	124	Currently	?
11	0,0049	0	240	0,0053	Good	?
12	0,0047	0	151	80	Currently	?

This study uses a split validation technique with a confusion matrix, where the dataset is divided into two parts, 70% (28) of the dataset is used as training data, and 30% (12) of the remaining is used as test data. Get the category classification in table 9 as follows.
P = (K = Very good) = 4/28 = 0.14
P = (K = Good) = 16/28 = 0.57
P = (K = Medium) = 4/28 = 0.14
P = (Poor K) = 4/28 = 0.14
1st Data:
The probability value for each attribute. Calculating attributes is great as follows:
P(Throughput = 1500 | K = Very Good) = 4/4 = 1
P(Packet loss = 0.03 | K = Very Good) = 4/4 = 1
P(Delay = 100 | K = Very Good) = 4/4 = 1
P(Jitter = 0 | K = Very Good) = 4/4 = 1
Counting good attributes include:
P(Throughput = 1500 | K = Good) = 13/16 = 0.81
P(Packet loss = 0.03 | K = Good) = 13/16 = 0.81

P(Delay = 100 | K = Good) = 12/16 = 0.75
P(Jitter = 0 | K = Good) = 0/19 = 0
Calculates the medium attribute as follows:
P(Throughput = 1500 | K = Medium) = 1/4 = 0.25
P(Packetloss = 0.03 | K = Medium) = 1/4 = 0.25
P(Delay = 100 | K = Medium) = 1/4 = 0.25
P(Jitter = 0 | K = Medium) = 0/4 = 0
Counting bad attributes include:
P(Throughput = 1500 | K = Bad) = 4/4 = 1
P(Packet loss = 0.03 | K = Ugly) = 0/4 = 0
P(Delay = 100 | K = Bad) = 0/4 = 0
P(Jitter = 0 | K = Bad) = 0/4 = 0
Calculating the results of very good, good, moderate, and bad attributes are as follows:
(Category = Very Good) = 0.14 x 1 x 1 x 1 x 1 = 0.14
(Category = Good) = 0.57 x 0.81 x 0.81 x 0.75 x 0 = 0
(Category = Medium) = 0.14 x 0.25 x 0.25 x 0.25 x 0 = 0
(Category = Poor) = 0.14 x 1 x 0 x 0 x 0 = 0
It can be concluded that the highest score is in the Very Good category, with a value of 0.14.
The 2nd to 12th data are calculated in the same way as the first data. The results of the calculation process using the Naïve Bayes method on the testing data, totaling 12 data, are shown in Table 5.
Path determination based on the application category used can be constructed based on the context of network neutrality (net neutrality). The relevance of data application performance is also important because it can be used to detect potential degradation of application users. Table 5 illustrates popular applications by non-professional users with the relevance of QoS parameters to the performance of these applications. In table 5, the value of relevance uses a value from 1 to 4, where 1 (less relevant) to 4 (very relevant). This perceived relevance value refers to the ITU-T Rec standard. G.1011, and ITU-T Rec. E.800. Category values can be recalculated using Formula 1.

Table 5.Table Of Nave Bayes Calculation Results

Troughput	Packetloss	Delay	Jitter	Current Category	Prediction Category	Track
1500	0,03	100	0	Very good	Very good	Line 1
339	3	0	0,007	Good	Good	Line 2
1234	0	20	0	Very good	Very good	Line 1
1295	5	155	0,005	Good	Good	Line 2
755	16	185	0,0032	Good	Good	Line 2
0,0021	1	387	295	Currently	Currently	Line 3
117	26	70	77	Currently	Good	Line 2
0,0013	5	238	89	Currently	Currently	Line 3
0,515	25	470	227	Bad	Bad	Line 4
0,044	16	268	124	Currently	Currently	Line 3
0,0049	0	240	0,0053	Good	Good	Line 2
0,0047	0	151	80	Currently	Currently	Line 3

Naive Bayes Test.
From the results obtained in table 5, a confusion matrix table can be formed, as shown in the table 6:

Table 6. Confussion Matrix

		Predict				
		Very Good	Good	Currently	Bad	
Actual	Very good	2	0	0	0	2
	Good	0	4	0	0	4
	Currently	0	1	4	0	5
	Bad	0	0	0	1	1
		2	5	4	1	

Based on table 11, the performance of the use of the Naïve Bayes method can be measured by calculating the accuracy and precision values.
Accuracy = (TP + TN)/Total
= ((2+4) + (4+1)) / 12 = 91%
Precision = Tp/ Total Predict category
Very Good = 2 / 2 = 100%
Good = 4 / 5 = 80 %
Currently = 4 / 4 = 100%
Bad = 1 / 1 = 100%
The average Precision is 95%
Routing testing uses static Routing, according to the recommendations of the Naïve Bayes algorithm. This test is carried out to find out that the network can be connected to each other and to know the path traversed from h1 to h2, which can be explained as follows.

a. Pathway 1 Testing

Line 1 is used for www. browsing media applications, where h1 will pass through switch 1, which has IP addresses 172.16.10.1/24 and 172.16.20.2/24, switch 2 with IP addresses 172.16.20.1/24 and 172.16.30.2/24, and switch 4 with IP addresses are 172.16.30.1/24 and 172.16.40.1/24, and to connect several switches, a gateway or connecting gate is used between switches. The gateway between switch 1 and switch 2 is 172.16.20.1, then the gateway between switch 2 and switch 1 is 172.16.20.2, then the gateway between switch 4 and switch 2 is 172.16.30.2, to connect switch 2 and switch 4 an additional command is used. in the form of a destination with a network of 172.16.40.0/24 and a gateway that is 172.16.30.1. The description of the route is shown in Figure 7.

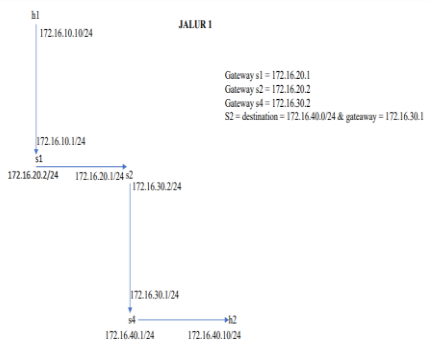

Figure 7. Line 1

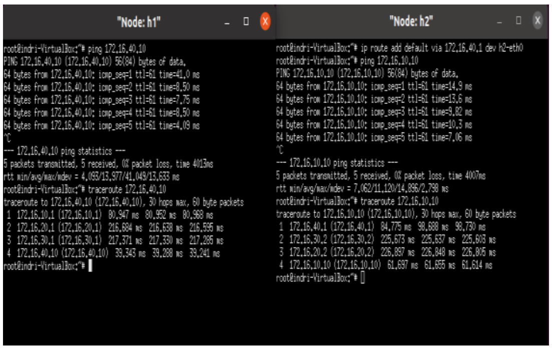

Figure 8. Results of Path 1

b. Pathway Testing 2

Line 2 is used for VoIP Telephony, where h1 will pass through switch 1, which has IP address 172.16.10.1/24 and 172.16.20.2/24, switch 3 with IP address 172.16.20.3/24 and 172.16.30.3/24, and switch 4 with IP addresses, namely 172.16.30.4/24 and 172.16.40.1/24, and to connect several switches, a gateway or connecting gate is used from an inter-switch. The gateway between switch 1 and switch 3 is 172.16.20.3, then for the gateway between switch 3 and switch 1 is 172.16.20.2, then the gateway between switch 4 and switch 3 is 172.16.30.3, to connect switch 3 and switch 4 an additional command is used in the form of a destination with a network of 172.16.40.0/24 and a gateway that is 172.16.30.4. The description of the route is shown in Figure 9.

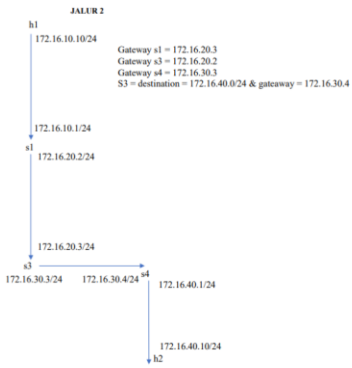

Figure 9. Line 2

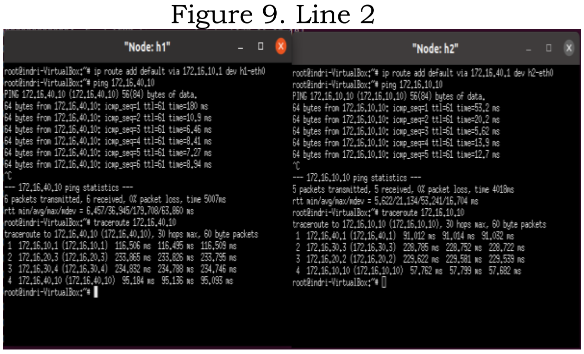

Figure 10. Results of Path 2

c. Path Test 3

Line 3 is used for HD Video Streaming services, where h1 will pass through switch 1 which has an IP address of 172.16.10.1/24 and 172.16.60.1/24, switch 4 with an IP address of 172.16.60.2/24 and 172.16.40.1/24, and To connect several switches, a gateway or connecting gate is used from an inter-switch. The gateway between switch 1 and switch 4 is 172.16.60.2, then the gateway between switch 4 and switch 1 is 172.16.60.1. The description of the route is shown in Figure 11.

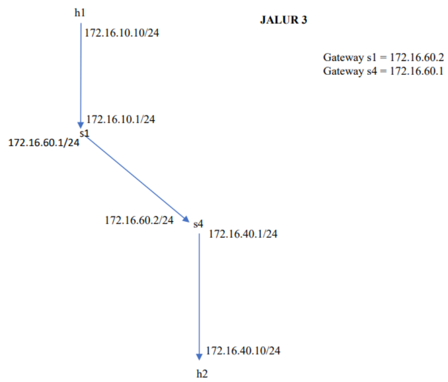

Figure 11. Line 3

3. CONCLUSION

The Naïve Bayes algorithm can provide routing recommendations for the type of application that has been determined based on the throughput, delay, packet loss, and jitter parameter criteria as training and test datasets. The recommendations are set for the WWW browsing application through route 1 with very good QoS category, VoIP Telephony through line 2 with good QoS category, and HD Video Streaming via line 3 with medium category. The results of calculating the accuracy of a prediction category using the Naïve Bayes algorithm are 91%, and the precision results are 95%. The routing simulation can run on an Openflow network using mininet as the host emulator, the switch as the data plane, and Ryu as the control plane.

References

[1] D. Hadiyansyah, W. Yahya, and W. Kurniawan, "Implementasi Penentuan Bobot Link Menggunakan Logika Fuzzy Untuk Pencarian Jalur Terpendek Pada Software Defined Networking," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 9, pp. 2677–2685, 2018.

[2] A. D. Rahmawan, S. Syaifuddin, and D. Risqiwati, "Analisa Performansi Controller Pada Arsitektur Jaringan Software Defined Network (Sdn)," *J. Repos.*, vol. 2, no. 12, p. 1727, 2020, doi: 10.22219/repositor.v2i12.75.

[3] D. Abdillah, Y. Sibaroni, and I. Ummah, "Design and Analysis of Virtual Network Based on Software- Define Networking (SDN)," *e-Proceeding Eng.*, vol. 3, no. 1, pp. 1247–1252, 2016.

[4] Roni Fernando, "Simulasi aringan Software Defined Network Menggunakan protokol Routing OSPF dan RYU Controller," *e-Proceeding Appl. Sci.*, vol. 04, no. 1, p. 2889, 2018.

[5] A. A. Wicaksono, S. N. Hertiana, and D. Irawati, "Analisis Performansi Load Balancing Dengan Metode Pemilihan Jalur Beban Terkecil Pada Sdn (Software Defined Network) Analysis Performance of Load Balancing Using Path With the (Software Defined Network)," *e-Proceeding Eng.*, vol. 6, no. 2, pp. 3441–3450, 2019.

[6] R. M. Negara and R. Tulloh, "Analisis Simulasi Penerapan Algoritma OSPF Menggunakan RouteFlow pada Jaringan Software Defined Network (SDN)," *J. Infotel*, vol. 9, no. 1, pp. 75–83, 2017.

[7] A. Rahmansyah, O. Dewi, P. Andini, T. Hastuti, P. Ningrum, and M. E. Suryana, "Membandingkan Pengaruh Feature Selection Terhadap Algoritma Naïve Bayes dan Support Vector Machine," pp. 1–7, 2018.

[8] W. Sun, Z. Wang, and G. Zhang, "A QoS-guaranteed intelligent routing mechanism in software-defined networks," *Comput. Networks*, vol. 185, p. 107709, 2021, doi: 10.1016/j.comnet.2020.107709.

[9] R. B. Herdian and L. Jasa, "Perilaku Pengguna Pada Jaringan TCP / IP Dengan," vol. 19, no. 1, pp. 73–82, 2020.

[10] M. Penelitian and D. A. N. Penulisan, "Metode penelitian dan penulisan ilmiah 1," *Univ.*

Stuttgart, pp. 1–18, 2006.
[11] P. R. Utami, "Analisis Perbandingan Quality of Service Jaringan Internet Berbasis Wireless Pada Layanan Internet Service Provider (Isp) Indihome Dan First Media," *J. Ilm. Teknol. dan Rekayasa*, vol. 25, no. 2, pp. 125–137, 2020, doi: 10.35760/tr.2020.v25i2.2723.
[12] G. D. Orueta, E. S. C. Ruiz, N. O. Alonso, and M. C. Gil, *Quality of service*. 2016.
[13] J. Informatika, D. A. N. Perancangan, S. Jips, and V. N. Mei, "Implementasi Proxy Server Menggunakan Squid Sebagai Sistem," vol. 4, no. 2, pp. 1–7, 2022.
[14] A. Akbar Ritonga, Ibnu Rasyid Munthe, Masrizal, "Jurnal Mantik Jurnal Mantik," *Mobile-Based Natl. Univ. Online Libr. Appl. Des.*, vol. 3, no. 2, pp. 10–19, 2019, [Online]. Available: http://iocscience.org/ejournal/index.php/mantik/article/view/882/595.
[15] M. K. Rahman, "Aplikasi sederhana Packet-Filter Firewall OpenFlow Ryu Controller," *Osf.Io*, pp. 0–3, [Online]. Available: https://osf.io/chkev/download.