

# Low-rate distributed denial of service attacks detection in software defined network-enabled internet of things using machine learning combined with feature importance

Muhammad Abizar, Muhammad Ferry Septian Ihzanor Syahputra, Ahmad Rizky Habibullah,  
Christian Sri Kusuma Aditya, Fauzi Dwi Setiawan Sumadi

Department of Informatics, Faculty of Engineering, Universitas Muhammadiyah Malang, Malang, Indonesia

## Article Info

### Article history:

Received Oct 9, 2022

Revised Apr 15, 2023

Accepted May 7, 2023

### Keywords:

Feature importance

Internet of things

Low-rate distributed denial of service

Machine learning

Software defined network

## ABSTRACT

One of the main challenges in developing the internet of things (IoT) is the existence of availability problems originated from the low-rate distributed denial of service attacks (LRDDoS). The complexity of IoT makes the LRDDoS hard to detect because the attack flow is performed similarly to the regular traffic. Integration of software defined IoT (SDN-Enabled IoT) is considered an alternative solution for overcoming the specified problem through a single detection point using machine learning approaches. The controller has a resource limitation for implementing the classification process. Therefore, this paper extends the usage of Feature Importance to reduce the data complexity during the model generation process and choose an appropriate feature for generating an efficient classification model. The research results show that the Gaussian Naïve Bayes (GNB) produced the most effective outcome. GNB performed better than the other algorithms because the feature reduction only selected the independent feature, which had no relation to the other features.

*This is an open access article under the **CC BY-SA** license.*



## Corresponding Author:

Fauzi Dwi Setiawan Sumadi

Department of Informatics, Universitas Muhammadiyah Malang

246 Raya Tlogomas Street, Malang 65144, East Java, Indonesia

Email: fauzisumadi@umm.ac.id

## 1. INTRODUCTION

The internet of things (IoT) is a concept where various smart devices are connected via the Internet to collect and transfer data or information [1]. The advancement of IoT is accompanied by efforts to modernize the global communication infrastructure that revolutionizes many aspects of life, enabling system interconnection with intelligent communication [2]. Examples of IoT implementations include medical devices, medical care, driverless vehicles, industrial robots, and smart city infrastructure with remote interaction models [1]. The rapid development of IoT will increase the number of smart devices connected to public networks, raising problems of complexity and security [2]. Even though IoT devices are growing, IoT networks are vulnerable to availability attacks, such as denial of service (DoS) and distributed denial of service (DDoS). Such attacks can quickly attack devices connected to an IoT network.

Moreover, the use of botnets can increase the volume of DDoS attacks, which can tamper the IoT services. In addition, traditional security mechanisms tend to be unsuitable for being implemented because IoT devices have less memory, processing capacity, and power. Due to its resource-limited characteristics, IoT tends to have more vulnerabilities that attackers can easily exploit [2]. This raises concerns about the security risks of IoT networks caused by the large-scale incorporation of smart devices. Due to the rapid development

of IoT, there are more and more efforts made by attackers to find loopholes to infiltrate the IoT network. Low-rate distributed denial of service attack (LRDDoS) is a serious threat to IoT infrastructure networks, among many attacks. LRDDoS attacks present an ongoing threat to almost every internet service as they attack server resources and can also potentially bring down the network. In addition, the main challenge with detecting LRDDoS attacks is the complexity of the attacking pattern. Massive traffic analysis will significantly consume the use of computing resources and even increase the risk of memory overflow.

IoT devices integrated with software defined network (SDN) [3], namely SDN-Enabled IoT, can significantly reduce the amount of computing overhead and provide additional security [4]. IoT aims to distribute data, and SDN provides services for network management by separating the control and data plane. However, because of this separation, the controller becomes a vulnerable target for cyber security attacks. Among all of the possible attacks, the availability threat may direct its attack to the controller by overwhelming the node using flooding, namely DDoS. In response, the controller will process every unwanted packet from the attackers. If the controller crashes, the entire network will collapse [5]. DDoS attacks are categorized into Flood and Shrew according to their characteristics and attack speed. Among them, Flood attacks are divided into high-rate (DDoS attacks with massive delivery rates) and low-rate (which are included in Flood attacks, but the transmission speed is less than 1,000bps). Their division is based on the packet transmission speed [6]. In the SDN-Enabled IoT network, attacks will occur at several levels, such as HRDDoS attacks in the control plane and LRDDoS attacks in the data plane. Attackers launch high-rate DDoS attacks at the SDN control layer by sending large amounts of useless data to weaken controllers and network resources.

A controller running out of resources will cause the entire SDN network to crash. However, HRDDoS attacks on controllers have traffic characteristics that are easy to identify, which can be pointed out by the significant rise in traffic amount in a short period [7]. In contrast, the LRDDoS are hard to detect because it has the same characteristics as regular traffic. So, the general DDoS attack detection mechanism (statistics) is ineffective in detecting LRDDoS because deep packet inspection (DPI) should be performed in order to retrieve the detailed information on the packet's header [8]. Unlike HRDDoS attacks, LRDDoS generates very little attack traffic and is stealthy. With a slow and inconspicuous process, LRDDoS allows the target system's performance to decrease gradually until it completely fails [9]. Low-Rate DDoS attacks are in the form of periodic pulses, where the attacks sent are concentrated. The average attack traffic is small but carried out repeatedly so that it can reduce the quality of service [6]. LRDDoS has the same characteristics as a normal network in the data center: low delay, diversity, and synchronization [10] so LRDDoS will not be easily detected if its characteristics match normal traffic. Low-rate DDoS attacks target the data layer with small attack traffic levels. Attackers can take advantage of it to launch LRDDoS attacks that hide in normal data streams and are difficult to detect with traditional methods.

Several studies have been conducted to detect LRDDoS attacks on SDN and SDN-Enabled IoT networks. Altamemi *et al.* [11] proposed a method for classifying DDoS attacks which include either high-rate or low-rate attacks based on real-time traffic datasets using machine learning method (Gaussian Naïve Bayes (GNB), logistic regression (LR), and decision tree (DT)). The research outcomes showed that DT could produce better accuracy than the other algorithms by gaining 99.9%. However, this paper did not use the appropriate dataset extracted using OpenFlow protocol in order to provide better data classification. Wani and Revathi [12] proposed a ransomware detection system in an IoT environment integrated with SDN, namely IoTSDN-RAN. The classification was performed by inspecting the constrained application protocol (CoAP) packet received by the controller using a combination of GNB and principal component analysis (PCA). The results indicated that the proposed method could predict ransomware traffic, proven by the accuracy pointed at 97.91%. John and Nagappasetty [13] investigated the detection scheme for detecting a Slowloris attack with slow bandwidth traffic aimed to simultaneously open a hypertext transfer protocol (HTTP) connection between the attacker and the targeted server. The authors utilized a statistical approach by extracting the flow statistic provided by OpenFlow. However, the results indicated that the statistical approach did not detect the attack as faster as the Machine Learning approach, proven by the detection time pointed at 260s. Research conducted by Azmi and Sumadi *et al.* [14] aims to detect LRDDoS using the support vector machine (SVM) combined with feature importance using logistic regression (LR) [15], [16]. Feature importance is useful for sorting the features contained in the OpenFlow protocol to ease the controller's classification process. The best accuracy is found in SVM with Linear Kernel, with accuracy reaching 100%. However, in terms of training time, linear SVM takes about 23.6 seconds, while SVM with kernel radial basis function (RBF) is much faster, which is only 1.5 but with lower accuracy results, and the average accuracy only gets 74.3%.

Cheng *et al.* [7] researched machine learning to detect LRDDoS attacks on SDN-Enabled IoT networks. In this study, the researchers tried to overcome one of the LRDDoS, shrewattack. The features used in this study are taken from features extracted from the OpenFlow protocol and are divided into 2, namely stateless and stateful. These researchers used several algorithms: SVM, the multinomial Naive Bayes algorithm (NB), random forest (RF), and K-nearest neighbors (KNN). The dataset used is 204,888 packets containing synchronize transmission control protocol (TCP SYN) packets, repeated TCP transmissions other than normal.

The number of normal data packets is 48,509, including hypertext transfer protocol secure (HTTPS), HTTP, internet control message protocol (ICMP), and message queuing telemetry transport (MQTT). The RF algorithm obtains the highest accuracy value with an accuracy rate of 97% and has the best effect on the switch.

Maslan *et al.* [17] conducted a similar study by combining linear regression models (ANOVA) in the feature reduction process to increase the effectiveness of the classification process using machine learning. In addition, the dataset used in this study is the result of extraction in a test bed environment and has not used the SDN architecture. From the results obtained, RF is the best algorithm in the classification process, with an accuracy value of 98.70%. Khempetch and Wuttidittachotti [18] employed the deep learning method for detecting DDoS, specifically using deep neural network (DNN) and long short-term memory (LSTM). The results indicated that the algorithm could successfully classify the attack, proven by the accuracy value pointed at 99.97% on average. Huraj *et al.* [19] stated that IoT integrated with manufacturing processes could potentially threaten DDoS attacks. Researchers describe case studies of IoT device applications and show the vulnerabilities of these devices. In addition, the researcher proposes to use sample Flow (sFlow) to detect and protect against DDoS attacks during production using machine learning.

In a study by Pande *et al.* [20] DDoS detection was carried out using machine learning techniques, and the algorithm used for model training was RF, resulting in an accuracy value of 99.76%. Alashhab *et al.* [21] found that machine learning is the proposed most effective LRDDoS detection mechanism in addition to other detection techniques. The researchers divided the LRDDoS detection mechanism categories based on machine learning into classification-based and deep learning-based. Wang *et al.* [22] in their research, explained that DDoS attacks are not only centered on the data plane but also in the control plane, causing fluctuations in the number of flows. In this study, the researchers built a DDoS attack with a separate SDN architecture and a new model to define the attack flexibly. The detection model used by the researcher is supervised learning. At the testing stage, the models that produce the highest accuracy values are decision tree (DT), KNN, and bagging tree (BT), with values above 90%. However, the sample used in this study is still lacking to get better accuracy results because it only uses one feature.

Based on previous research, it can be concluded that machine learning is an effective method of detecting LRDDoS attacks. However, no authors provided a thorough analysis of performing the LRDDoS detection using minimal resources in an IoT environment and maintained its datasets to conform with the OpenFlow standard. In this study, the solution proposed by the author to deal with LRDDoS attacks is an integration of SDN-Enabled IoT with machine learning combined with Feature Importance. Machine learning has the function of creating models that are used in the classification process by the controller. The model generation process is combined with three feature importance methods, namely LR, random forest classifier (RFC), and random forest regression (RFR), to reduce the number of features so that the load received by the controller will be reduced because the resources used are only the relevant features. The model goes through a training process using eight different algorithms, including SVM with linear kernel and RBF, RF, DT, multi-layer perceptron (MLP), GNB, AdaBoost (ADB), and KNN. Each model used in the classification process will produce accuracy, precision, recall, F1-score, and classification-loss values from each algorithm. The contribution given in this research is performing LRDDoS detection utilizing several supervised algorithms combined with three different Feature Importance methods for computational reduction in the classification process and adjusting the dataset of LRDDoS with the OpenFlow protocol based on the port statistic. Adjusting the dataset will also significantly improve the accuracy of the detection mechanism since the features were easily extracted on the controller. In addition, this study also compares which algorithm is the most appropriate for detecting LRDDoS attacks from each Feature Importance method.

## 2. RESEARCH METHOD

### 2.1. Emulation's topology and scenario

In this study, the test was operated on an Ubuntu 20.04 LTS computer with a specification of Intel® Core™ i5-10400 CPU @ 2.90 GHz, 8 GB of RAM, and 240 GB of SSD. The SDN-Enabled IoT network topology was emulated by the Mininet emulator [23]. Based on Figure 1, the components used in the network architecture in this topology consisted of 7 Open vSwitch (OvS) [24], [25], 1 RYU Controller [26], and 8 Hosts. The applied topology was a tree with configuration variables of depth=3 and fanout=2. In the topology that had been developed, h1 acted as an attacker, and h6 acted as a victim and a CoAP server [27] with a logical address of 10.0.0.6:5683. As an attacker, h1 overwhelmed the topology using the TCPReplay tool [28] 3 times with different packet transmission speeds, consisting of 20, 50, and 70 packets per second (PPS).

The attack carried out by h1 was sent via a \*.pcap file containing dummy packets. In each of these packets, the IP and MAC source addresses were composed of values that were randomly generated in as many as 39,994 packets using the CoAP (POST) protocol. Packet header information that went to OvS was processed according to the rules defined by the controller. If there was no matching header, the packet was detected as a

new packet and would be processed by the controller directly for network learning purposes. Because the data sent by the attacker was composed of random source addresses, it could indirectly interfere with the controller's performance. If the controller could not withstand the load, this attack could collapse the SDN-Enabled IoT network.

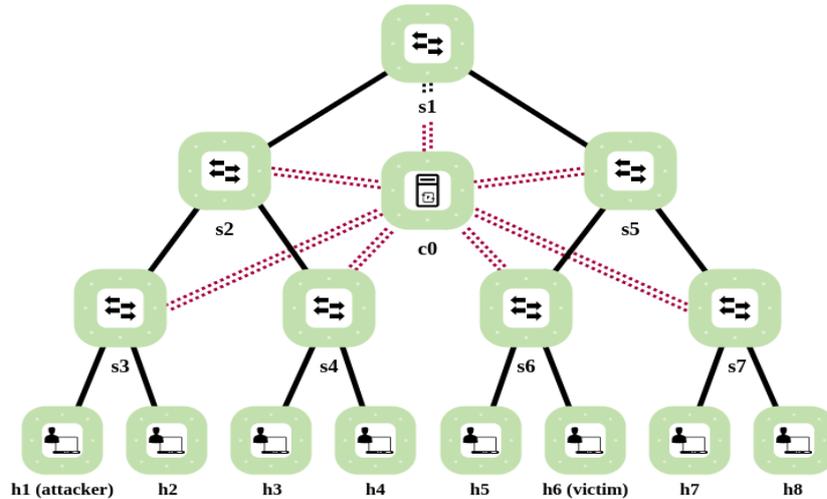


Figure 1. Emulation topology

## 2.2. LRDDoS dataset

The dataset used in this study utilized the OpenFlow protocol to investigate the impact of LRDDoS attacks [29]. The data was generated by crafting a CoAP packet using Scapy and transmitting both normal and LRDDoS packets using TCPReplay. The controller extracted the receiving packet using the OpenFlow protocol described in Table 1. All available hosts on the network perform normal traffics, which is directed to communicate using a CoAP POST message to the server (h6). In contrast, the LRDDoS packets contain dummy packets composed of randomly generated source addresses. This data was divided into two parts: a training dataset of 160,006 packets and a test data of 39,994 packets. If totaled, the total dataset was 200,000 packets. The composition in the dataset had a 1:1 ratio for LRDDoS and normal packets.

The packets were extracted using the OpenFlow feature, including standard headers on IPv4, UDP/TCP protocols, and OvS port usage statistics. The total number of features contained in the OpenFlow protocol was 21. This number could still be excessive, burdening the controller in the classification process. In order to reduce this burden, it was necessary to simplify the number of features used, using feature selection based on the coefficient score of feature importance. The Feature Importance method used in this study includes LR, RFC, and RFR. The results of the LR and RFC processing obtained eight features, while RFR only used two features, which could improve the training model's performance and reduce the workload on the controller. The results of the calculation of feature importance can be seen in Table 1.

Some features marked "-" were not used in the model generation process because they were equal to 0. Features taken from feature importance only had a value greater than 0 or less than 0. Features with a coefficient value of 0 would not affect the evaluation variables originating from the classification process based on the generated model. The comparison between the techniques used in LR and RF on the Feature Coefficient was that the LR method was calculated with all features as input in the model, while the RFC and RFR calculated the coefficient separately for each feature [30].

## 2.3. Model generation and classification process

Figure 2 shows a system block diagram that includes the feature reduction process without eliminating information that is considered essential or relevant based on the value of the coefficient score inputted into each feature to be predicted. This feature reduction process used three Feature Importance methods, LR, RFC, and RFR, followed by model generation, which employed eight different algorithms. All features in the training set would have a coefficient value, and their relevance to the classification process was assessed. In Table 1, the relevant features are shown with a positive or negative value, while those with a value of 0 are removed because they have no significant impact on the classification process. The selected features from each Feature Importance would later be used in the training stage of the classification model with the SVM Linear, SVM RBF, RF, DT, MLP, GNB, ADB, and KNN. This stage generated a classification model used by the SDN-

Enabled IoT controller application to detect attacks from incoming packets. The classification process performed by the controller will be faster because it uses fewer resources by selecting the most relevant features based on the coefficient score. Therefore, the controller did not thoroughly extract all of the 21 features. The use of Feature Importance also prevented a decrease in the quality of the model. After the model was completed, the model was used as a classification model on the SDN-Enabled IoT controller. The model was added to the simple\_switch\_13 application that already existed on the RYU controller.

Table 1. Feature list

Features Name	Features Origin	Coefficient Score		
		Logistic Regression	Random Forest Classifier	Random Forest Regression
datapath_id	OFPT_PACKET_IN	-	-	-
version	IPv4's Header	-	-	-
header_length	IPv4's Header	-	-	-
tos	IPv4's Header	-	-	-
total_length	IPv4's Header	-1.61720	0.05396	-
flags	IPv4's Header	6.76580	0.35920	0.59000
offset	IPv4's Header	-	-	-
TTL	IPv4's Header	-	-	-
proto	IPv4's Header	-	-	-
csum	IPv4's Header	-0.00195	0.00062	-
srcp_ip	IPv4's Header	1.85064	0.30681	0.41000
dst_ip	IPv4's Header	-	-	-
src_port	UDP's/TCP's Header	-0.26961	0.07073	-
dst_port	UDP's/TCP's Header	-	-	-
port_no	OFPPortStatsReply	-0.08737	0.00001	-
rx_bytes_ave	OFPPortStatsReply (rx_bytes/rx_packets)	3.04460	0.17514	-
rx_error_ave	OFPPortStatsReply (rx_bytes/rx_packets)	-	-	-
rx_dropped_ave	OFPPortStatsReply (rx_bytes/rx_packets)	-	-	-
tx_bytes_ave	OFPPortStatsReply (rx_bytes/rx_packets)	0.08789	0.03354	-
tx_error_ave	OFPPortStatsReply (rx_bytes/rx_packets)	-	-	-
tx_dropped_ave	OFPPortStatsReply (rx_bytes/rx_packets)	-	-	-

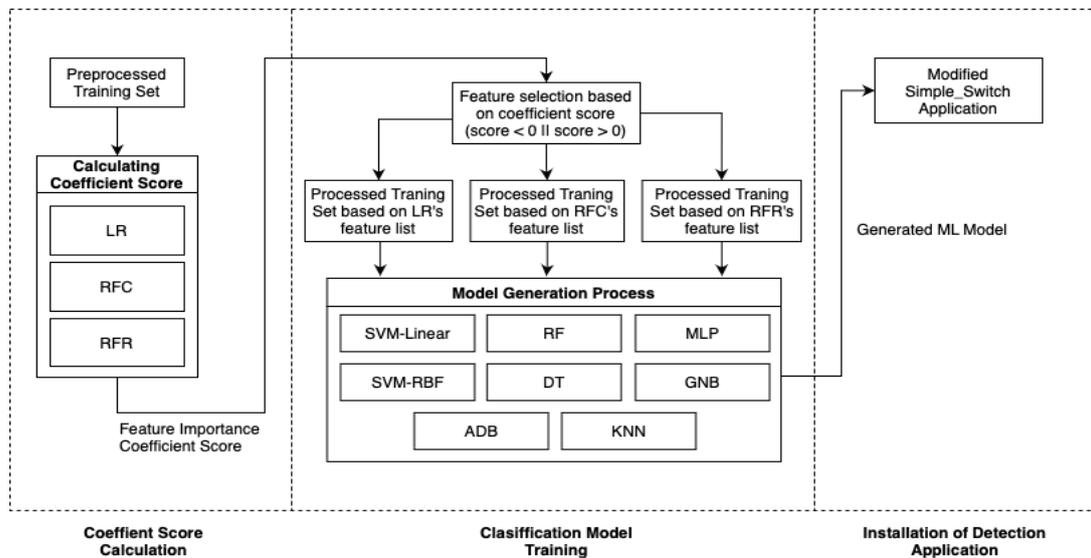


Figure 2. System block diagram

The detail of the classification process is described in Figure 3. The process started with the attacker sending a testing set that contained normal and LRDDoS packets using TCPReplay. Because the components of the attack packet were constructed randomly (source MAC and IP address), the packet was sent to the

controller for learning. The model that was formed from 8 algorithms and 3 Feature Importance functioned to classify packets into the LRDDoS type or normal packets. The classification results were stored in a file for measuring the level of effectiveness using the accuracy, precision, recall, and F1-score. The data was compared with the original class in the testing set of each classified packet. In the classification process, some data was not successfully classified because the link on the OvS was overwhelmed. This condition could be measured by calculating the classification loss value from the total of all successfully categorized packets.

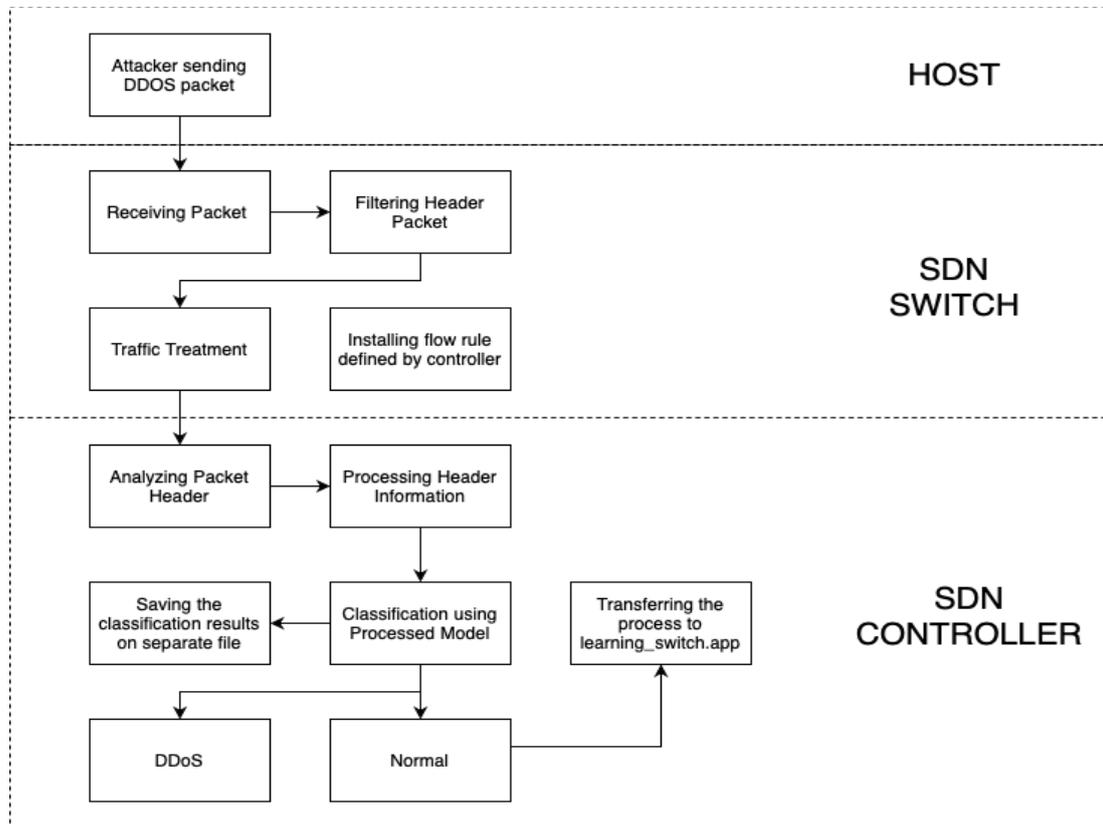


Figure 3. Classification process in SDN-enabled IoT

### 3. RESULTS AND DISCUSSION

#### 3.1. Feature importance and reduction

A large number of features can impact the controller because the greater the resource, the greater the burden the controller receives. Therefore, it is necessary to have a feature reduction process to reduce the number of features that will be used in the classification process by selecting the most relevant features and removing features that will not be too useful for the model to be trained and can even reduce the quality of the model. The feature selection process in this study applied three different Feature Importance methods, namely LR, RFC, and RFR. In this study, the dataset used has a total of 21 features. With the Feature Importance method, only certain features will be used in the classification process to reduce the overload received by the controller.

The feature importance score can be calculated for problems involving the prediction of numerical values called regression and problems involving the prediction of class labels called classification. The Feature Importance method selects features based on the results of a positive and negative coefficient score. This coefficient score can provide a fundamental basis for the feature score that is considered essential. Features with a coefficient score of 0 will be removed to prevent poor model quality. The coefficient score is obtained from the results of entering the score into the input feature for a predictive model that indicates the relative importance of each feature when making predictions. As shown in Table 2, after selecting features using Feature Importance, only eight features have a coefficient score of 0 out of a total of 21 features selected using the Feature Importance LR and RFC methods, while for RFR, there are only two relevant features. In terms of the selected features, the feature affects the prediction model because it has a coefficient score other than 0, which indicates that the feature plays an essential role in the classification process.

Table 2. Selected features based on feature importance methods

Features Name	Features Origin	Coefficient Score		
		Logistic Regression	Random Forest Classifier	Random Forest Regression
total_length	IPv4's Header	-1.61720	0.05396	-
flags	IPv4's Header	6.76580	0.35920	0.59000
csum	IPv4's Header	-0.00195	0.00062	-
srp_ip	IPv4's Header	1.85064	0.30681	0.41000
src_port	UDP's/TCP's Header	-0.26961	0.07073	-
port_no	OFPPortStatsReply	-0.08737	0.00001	-
rx_bytes_ave	OFPPortStatsReply (rx_bytes/rx_packets)	3.04460	0.17514	-
tx_bytes_ave	OFPPortStatsReply (rx_bytes/rx_packets)	0.08789	0.03354	-

### 3.2. Training result in SDN-enabled IoT

Based on Table 3, in the classification model training process, it can be seen that the LR, among several other models, has perfect results with an average value of close to 100% for accuracy, precision, recall, and F1-score. The GNB model was superior because it was also considered the fastest in performing the training than the other models. The RF model produced the worst results, with an accuracy ratio of 89%, 79% for recall, and 88% for the F1-score. In the process of learning the model, the GNB model took time faster than the RF. The GNB model required about 0.031 seconds to train data, while the RF consumed 0.211 seconds. It can be seen in Table 4 that GNB also obtained perfect results of accuracy, precision, recall, and F1-score with a difference in the training data time of about 0.022 seconds, while the RFC model also produced the lowest results among other models.

In the RFR training test, the GNB model became the best and fastest among the previous two methods, as illustrated in Table 5. The GNB model obtained an average result of 100% for accuracy, precision, recall, and F1-score, with the training time data getting 0.013 seconds. In comparison, the RFC and DTC models produced a 10% lower accuracy difference, 20% lower recall, and an 11% lower F1-score. Regarding the time training, there was a difference between RFC and DTC. DTC performed faster within 0.017 seconds, while RFC consumed 0.158 seconds for training data. Based on the results of these tests, it can be concluded that the GNB model is the most effective and fastest model for the three coefficient score calculation methods because it works very well for large amounts of data and does not require a long time in the data training process.

Table 3. LR's training results

Algorithm	Accuracy %	Precision %	Recall %	F1-Score %	Training Time (s)
SVM-LINEAR	100	100	100	100	22.462
SVM-RBF	99	100	100	100	3.541
RF	89	100	79	88	0.211
DT	90	100	80	89	0.050
MLP	100	100	100	100	5.354
GNB	100	100	100	100	0.031
ADB	100	100	100	100	0.064
KNN	100	100	100	100	0.168

Table 4. RFC's training results

Algorithm	Accuracy %	Precision %	Recall %	F1-Score %	Training Time (s)
SVM-LINEAR	100	100	100	100	21.971
SVM-RBF	99	100	100	100	3.746
RF	80	86	80	79	0.233
DT	100	100	100	100	0.043
MLP	100	100	100	100	4.243
GNB	100	100	100	100	0.022
ADB	100	100	100	100	0.052
KNN	99	100	100	100	0.159

### 3.3. Classification result in SDN-enabled IoT network

Table 6 is the result of research emulated in SDN-Enabled IoT using the LR feature, which is accumulated from three packet sending rates, including 20, 50, and 70 pps. The highest scores on the accuracy, precision, recall, and F1-score originated from SVM Linear, DTC, MLP, GNB, and ADB, with an overall score of 100%. Other models, namely SVM, RBF, and KNN, get the lowest results. Table 7 shows the classification results using the RFC model with the same attack data delivery speed (20, 50, and 70 pps). The table shows how the impact of classification loss. The accuracy will increase if the loss value is high because fewer data

are processed compared to the overall testing set. In the delivery range of 70 pps, the classification loss value produced results above 50%, which increased the accuracy value for all classification algorithms. The highest accuracy, precision, recall, and F1-score values were found in SVM Linear, DTC, GNB, and ADB, which were pointed at 100% overall. In contrast, the lowest value was generated by SVM RBF, MLP, and KNN.

Table 5. RFR's training results

Algorithm	Accuracy %	Precision %	Recall %	F1-Score %	Training Time (s)
SVM-LINEAR	100	100	100	100	0.131
SVM-RBF	100	100	100	100	0.491
RF	90	100	80	89	0.158
DT	90	100	80	89	0.017
MLP	100	100	100	100	3.888
GNB	100	100	100	100	0.013
ADB	100	100	100	100	0.024
KNN	100	100	100	100	0.048

Table 6. The results of LR in SDN-enabled IoT

Algorithm	Packet Rate (pps)	Accuracy %	Precision %	Recall %	F1-Score %	Classification Loss %
SVM LINEAR	20	100	100	100	100	4.03560
	50	100	100	100	100	4.06060
	70	100	100	100	100	4.15562
SVM RBF	20	51.879	25.939	50.0	34.158	4.03560
	50	51.879	25.940	50.0	34.158	4.06060
	70	51.868	25.934	50.0	34.153	4.15562
RF	20	90.360	92.165	89.983	90.183	4.03560
	50	90.360	92.165	90.983	90.183	4.06060
	70	90.353	92.165	89.978	90.177	4.15562
DT	20	100	100	100	100	4.03560
	50	100	100	100	100	4.06060
	70	100	100	100	100	4.15562
MLP	20	100	100	100	100	4.03560
	50	100	100	100	100	4.06060
	70	100	100	100	100	4.15562
GNB	20	100	100	100	100	4.03560
	50	100	100	100	100	4.06060
	70	100	100	100	100	4.15562
ADB	20	100	100	100	100	4.03560
	50	100	100	100	100	4.06060
	70	100	100	100	100	4.15562
KNN	20	51.879	25.939	50.0	34.158	4.03560
	50	51.879	25.940	50.0	34.158	4.06060
	70	868	25.934	50.0	34.153	4.15562

Table 7. The results of RFC in SDN-enabled IoT

Algorithm	Packet Rate (pps)	Accuracy %	Precision %	Recall %	F1-Score %	Classification Loss %
SVM LINEAR	20	100	100	100	100	4.03310
	50	100	100	100	100	4.20313
	70	100	100	100	100	51.21518
SVM RBF	20	51.880	25.940	50.0	34.158	4.03310
	50	51.933	25.966	50.0	34.181	4.20313
	70	76.890	38.445	50.0	43.468	51.21518
RF	20	80.768	86.468	79.996	79.672	4.03310
	50	80.803	86.506	80.031	79.724	4.20313
	70	94.311	96.555	87.691	91	51.21518
DT	20	100	100	100	100	4.03310
	50	100	100	100	100	4.20313
	70	100	100	100	100	51.21518
MLP	20	51.880	25.940	50.0	34.158	4.03310
	50	51.933	25.966	50.0	34.181	4.20313
	70	76.890	38.445	50.0	43.468	51.21518
GNB	20	51.880	25.940	50.0	34.158	4.03310
	50	100	100	100	100	4.20313
	70	100	100	100	100	51.21518
ADB	20	100	100	100	100	4.03310
	50	100	100	100	100	4.20313
	70	100	100	100	100	51.21518
KNN	20	51.880	25.940	50.0	34.158	4.03310
	50	51.933	25.966	50.0	34.181	4.20313
	70	76.890	38.445	50.0	43.468	51.21518

Another SDN-enabled IoT research result is RFR, as seen in Table 8. Only RF and DT have different values. This was because RFR only selected two features. The variable values of accuracy, recall, and F1-score in other models (SVM Linear, SVM RBF, MLP, GNB, ADB, and KNN) had the same overall value of 100%. From the three Feature Importance models, it could be concluded that SVM Linear, GNB, and ADB were the best algorithms because they had accuracy, precision, and recall reaching an average of 100% despite the classification loss was different for each delivery speed. The classification loss variable arose because the controller experienced overlapping data reception so that the testing set was not sent or the incoming data was received more than once. Receiving the same packet repeatedly would cause the classification value to increase because the number of classified data was less than the total test data sent. This pattern could happen because the emulator on mininet-IoT was unstable.

Table 8. The results of RFR in SDN-enabled IoT

Algorithm	Packet Rate (pps)	Accuracy %	Precision %	Recall %	F1 %	Classification Loss %
SVM LINEAR	20	100	100	100	100	4.17562
	50	100	100	100	100	4.06310
	70	100	100	100	100	4.34565
SVM RBF	20	100	100	100	100	4.17562
	50	100	100	100	100	4.06310
	70	100	100	100	100	4.34565
RF	20	90.345	91.146	89.983	90.172	4.17562
	50	90.359	92.165	89.982	90.182	4.06310
	70	90.354	92.165	89.971	90.176	4.34565
DT	20	90.345	91.146	89.983	90.172	4.17562
	50	90.359	92.165	89.982	90.182	4.06310
	70	90.354	92.165	89.971	90.176	4.34565
MLP	20	100	100	100	100	4.17562
	50	100	100	100	100	4.06310
	70	100	100	100	100	4.34565
GNB	20	100	100	100	100	4.17562
	50	100	100	100	100	4.06310
	70	100	100	100	100	4.34565
ADB	20	100	100	100	100	4.17562
	50	100	100	100	100	4.06310
	70	100	100	100	100	4.34565
KNN	20	100	100	100	100	4.17562
	50	100	100	100	100	4.06310
	70	100	100	100	100	4.34565

#### 4. CONCLUSION

Feature Importance allows us to understand the relationship of features with target variables, as well as understand which features are relevant and which are not for the model to be built. In addition, when conducting model training, the coefficient score becomes the basis for selecting features to reduce the model's dimensions and save resources to be used. This clearly can improve the performance of the model and controller in carrying out the classification process. Based on the analysis of the test results that have been carried out, the GNB algorithm is the best model for the classification process against LRDDoS attacks because it obtains a fast training time value and also the results of accuracy, recall, precision, and F1-score values in the range of 100% during the model training process. In the Feature Importance method, LR, RFC, and RFR each have a training time of about 0.031 seconds, 0.022 seconds, and 0.013 seconds, respectively. Three models have dominant results in the classification test with SDN-Enabled IoT, including ADB, SVM Linear, and GNB. However, compared to the ADB and SVM Linear models, although they both produce perfect results, if we analyze it in comparative testing without and with SDN-Enabled IoT, the GNB model is superior in all aspects. This is possible because the selected feature has independent properties from other features. In addition, the amount of data that is processed after processing the feature selection also has an impact on reducing the complexity of the data used in the classification process. In future research, the author plans to develop a dataset model that is more effective in handling availability cases while at the same time incorporating statistical techniques in the attack detection module.

#### ACKNOWLEDGEMENTS

This manuscript is based on research supported by the Universitas Muhammadiyah Malang under the Grant Number E.2.a/334/BAA-UMM/IV/2022. The authors would also express their gratitude for the UMM Informatics Laboratory, who have supported the implementation of this research.

## REFERENCES

- [1] A. Khraisat and A. Alazab, "A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges," *Cybersecurity*, vol. 4, no. 1, 2021, doi: 10.1186/s42400-021-00077-7.
- [2] A. Arul Anitha and L. Arockiam, "A review on intrusion detection systems to secure iot networks," *International Journal of Computer Networks and Applications*, vol. 9, no. 1, pp. 38–50, 2022, doi: 10.22247/ijcna/2022/211599.
- [3] H. H. Saleh, I. A. Mishkal, and D. S. Ibrahim, "Controller placement problem in software defined networks," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 27, no. 3, pp. 1704–1711, 2022, doi: 10.11591/ijeecs.v27.i3.pp1704-1711.
- [4] M. Aslam *et al.*, "Adaptive machine learning based distributed denial-of-services attacks detection and mitigation system for SDN-enabled IoT," *Sensors*, vol. 22, no. 7, 2022, doi: 10.3390/s22072697.
- [5] M. H. H. Khairi, S. H. S. Ariffin, N. M. Abdul Latiff, A. S. Abdullah, and M. K. Hassan, "A review of anomaly detection techniques and distributed denial of service (DDoS) on software defined network (SDN)," *Engineering, Technology & Applied Science Research*, vol. 8, no. 2, pp. 2724–2730, 2018, doi: 10.48084/etasr.1840.
- [6] W. Zhijun, L. Wenjing, L. Liang, and Y. Meng, "Low-rate DoS attacks, detection, defense, and challenges: A survey," *IEEE Access*, vol. 8, pp. 43920–43943, 2020, doi: 10.1109/ACCESS.2020.2976609.
- [7] H. Cheng, J. Liu, T. Xu, B. Ren, J. Mao, and W. Zhang, "Machine learning based low-rate DDoS attack detection for SDN enabled IoT networks," *International Journal of Sensor Networks*, vol. 34, no. 1, pp. 56–69, 2020, doi: 10.1504/ijnsnet.2020.109720.
- [8] J. A. Perez-Diaz, I. A. Valdovinos, K. K. R. Choo, and D. Zhu, "A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning," *IEEE Access*, vol. 8, pp. 155859–155872, 2020, doi: 10.1109/ACCESS.2020.3019330.
- [9] X. Liu, J. Ren, H. He, Q. Wang, and C. Song, "Low-rate DDoS attacks detection method using data compression and behavior divergence measurement," *Computers and Security*, vol. 100, 2021, doi: 10.1016/j.cose.2020.102107.
- [10] W. Wang, X. Ke, and L. Wang, "A HMM-R approach to detect L-DDoS attack adaptively on SDN controller," *Future Internet*, vol. 10, no. 9, 2018, doi: 10.3390/fi10090083.
- [11] A. J. Altamemi, A. Abdulhassan, and N. T. Obeis, "DDoS attack detection in software defined networking controller using machine learning techniques," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 5, pp. 2836–2844, 2022, doi: 10.11591/eei.v11i5.4155.
- [12] A. Wani and S. Revathi, "Ransomware protection in IoT using software defined networking," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 3, pp. 3166–3174, 2020, doi: 10.11591/ijece.v10i3.pp3166-3175.
- [13] P. M. John and R. M. B. K. Nagappasetty, "An approach for slow distributed denial of service attack detection and alleviation in software defined networks," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 25, no. 1, pp. 404–413, 2022, doi: 10.11591/ijeecs.v25.i1.pp404-413.
- [14] M. M. Azmi and F. D. S. Sumadi, "Low-rate attack detection on SD-IoT using SVM combined with feature importance logistic regression coefficient," *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, 2022, doi: 10.22219/kinetik.v7i2.1405.
- [15] A. S. Soma, T. Kubota, and H. Mizuno, "Optimization of causative factors using logistic regression and artificial neural network models for landslide susceptibility assessment in Ujung Loe Watershed, South Sulawesi Indonesia," *Journal of Mountain Science*, vol. 16, no. 2, pp. 383–401, 2019, doi: 10.1007/s11629-018-4884-7.
- [16] H. M. Rizeei, B. Pradhan, M. A. Saharkhiz, and S. Lee, "Groundwater aquifer potential modeling using an ensemble multi-adoptive boosting logistic regression technique," *Journal of Hydrology*, vol. 579, 2019, doi: 10.1016/j.jhydrol.2019.124172.
- [17] A. Maslan, K. M. Bin Mohamad, and F. B. Mohd Foozy, "Feature selection for DDoS detection using classification machine learning techniques," *IAES International Journal of Artificial Intelligence*, vol. 9, no. 1, pp. 137–145, 2020, doi: 10.11591/ijai.v9.i1.pp137-145.
- [18] T. Khempetch and P. Wuttidittachotti, "Ddos attack detection using deep learning," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 2, pp. 382–388, 2021, doi: 10.11591/ijai.v10.i2.pp382-388.
- [19] L. Huraj, T. Horak, P. Strelec, and P. Tanuska, "Mitigation against ddos attacks on an iot-based production line using machine learning," *Applied Sciences (Switzerland)*, vol. 11, no. 4, pp. 1–18, 2021, doi: 10.3390/app11041847.
- [20] S. Pande, A. Khamparia, D. Gupta, and D. N. H. Thanh, "DDoS detection using machine learning technique," *Studies in Computational Intelligence*, vol. 921, pp. 59–68, 2021, doi: 10.1007/978-981-15-8469-5\_5.
- [21] A. A. Alashhab, M. S. M. Zahid, M. A. Azim, M. Y. Daha, B. Isyaku, and S. Ali, "A survey of low rate DDoS detection techniques based on machine learning in software-defined networks," *Symmetry*, vol. 14, no. 8, 2022, doi: 10.3390/sym14081563.
- [22] S. Wang *et al.*, "Detecting flooding DDoS attacks in software defined networks using supervised learning techniques," *Engineering Science and Technology, an International Journal*, vol. 35, 2022, doi: 10.1016/j.jestch.2022.101176.
- [23] D. Y. Setiawan, S. N. Hertiana, and R. M. Negara, "6LoWPAN performance analysis of IoT software-defined-network-based using mininet-Io," *IoT&S 2020 - Proceedings: 2020 IEEE International Conference on Internet of Things and Intelligence Systems*, pp. 60–65, 2021, doi: 10.1109/IoT&S50849.2021.9359714.
- [24] M. Ushakova, Y. Ushakov, J. Cui, L. Legashev, A. Shukhman, and A. Bolodurin, "Research of performance parameters of virtual switches with OpenFlow support," *2020 International Conference Engineering and Telecommunication, En and T 2020*, 2020, doi: 10.1109/EnT50437.2020.9431289.
- [25] Y. Zhang, J. Bi, Z. Li, Y. Zhou, and Y. Wang, "VMS: Load balancing based on the virtual switch layer in datacenter networks," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1176–1190, 2020, doi: 10.1109/JSAC.2020.2986691.
- [26] S. Asadollahi, B. Goswami, and M. Sameer, "Ryu controller's scalability experiment on software defined networks," *2018 IEEE International Conference on Current Trends in Advanced Computing, ICCTAC 2018*, pp. 1–5, 2018, doi: 10.1109/ICCTAC.2018.8370397.
- [27] E. Al-Masri *et al.*, "Investigating messaging protocols for the internet of things (IoT)," *IEEE Access*, vol. 8, pp. 94880–94911, 2020, doi: 10.1109/ACCESS.2020.2993363.
- [28] J. Singh and S. Behal, "Detection and mitigation of DDoS attacks in SDN: A comprehensive review, research challenges and future directions," *Computer Science Review*, vol. 37, 2020, doi: 10.1016/j.cosrev.2020.100279.
- [29] F. Sumadi, "LRDDoS dataset (CoAP)," *Mendeley Data*, vol. 1, 2022, doi: 10.17632/g9g6g3bmjt.1.
- [30] M. Saarela and S. Jauhiainen, "Comparison of feature importance measures as explanations for classification models," *SN Applied Sciences*, vol. 3, no. 2, 2021, doi: 10.1007/s42452-021-04148-9.

**BIOGRAPHIES OF AUTHORS**

**Muhammad Abizar**     currently pursuing undergraduate degree majoring in informatics at the University of Muhammadiyah Malang since 2019. His areas of interest include SDN, Network Security. he can be contacted at email: aybewrld@gmail.com.



**Muhammad Ferry Septian Ihzanor Syahputra**     graduated from Telkom Banjarbaru Vocational High School with a major in Computer and Network Engineering, and is currently pursuing undergraduate degree majoring in informatics at the University of Muhammadiyah Malang. His areas of interest are IoT, software development, network management and security. He can be contacted at email: mhmdmferry713@gmail.com.



**Ahmad Rizky Habibullah**     graduated from Telkom Banjarbaru Vocational High School majoring in computer and network engineering. He is currently studying in Malang to be precise at the University of Muhammadiyah Malang in informatics Department. He interested in SD-IoT, network management, and network security. He can be contacted at email: rizky3habibullah@gmail.com.



**Christian Sri Kusuma Aditya**     graduated with a Master of Computer from Sepuluh Nopember Technological Institute (ITS), Surabaya. Currently, he is a lecturer in the Informatics Department University of Muhammadiyah Malang (UMM). His areas of interest are Data Science, Machine Learning, and Text Processing. He can be contacted at email: christianskaditya@umm.ac.id.



**Fauzi Dwi Setiawan Sumadi**     achieved his master degree program in computer science at the University of Queensland, Australia which focused on analyzing the vulnerability in software-defined networks. Nowadays, he has become one of the main lecturers in the Informatics Department at the University of Muhammadiyah Malang and maintains his research in the implementation of artificial intelligence in a computer network, distributed computing, Cyber security, IoT, and the SDN. He can be contacted at email: fauzisumadi@umm.ac.id.