Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control
Journal homepage: http://kinetik.umm.ac.id
ISSN: 2503-2267
Vol. 7, No. 2, May, Pp. 121-128

Low-rate attack detection on SD-IoT using SVM combined with feature importance logistic regression coefficient

Mirza Maulana Azmi[1], Fauzi Dwi Setiawan Sumadi*[2]
Universitas Muhammadiyah Malang, Indonesia[1,2]

Article Info

Keywords:
Low-Rate Attack, DDoS, SD-IoT, SVM, IDS

Article history:
Received: February 02, 2022
Accepted: February 25, 2022
Published: May 31, 2022

Cite:
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, vol. 7, no. 2, May. 2022.
https://doi.org/10.22219/kinetik.v7i2.1405

*Corresponding author.
Fauzi Dwi Setiawan Sumadi
E-mail address:
fauzisumadi@umm.ac.id

Abstract

The evolution of computer network technology is now experiencing substantial changes, particularly with the introduction of a new paradigm, Software Defined Networking (SDN). The SDN architecture has been applied in a variety of networks, including the Internet of Things (IoT), which is known as SD-IoT. IoT is made up of billions of networking devices that are interconnected and linked to the Internet. Since the SD-IoT was considered as a complex entity, several types of attack on vulnerabilities vary greatly and can be exploited by careless individuals. Low-Rate Distributed Denial of Service (LRDDoS) is one of the availability-based attack that may affect the SD-IoT integration paradigm. Therefore, it is necessary to have an Intrusion Detection System (IDS) to overcome the security hole caused by LRDDoS. The main objective of this research was the establishment of an IDS application for resolving LRDDoS attack using the SVM algorithm combined with the Feature Importance method, namely the Logistic Regression Coefficient. The implemented approach was developed to reduce the complexity or resource's consumption during the classification process as well as increasing the accuracy. It could be concluded that the Linear kernel SVM algorithm acquired the highest results on the test schemes at 100% accuracy, but the training time required for this model was longer, about 23.6 seconds compared to the Radial Basis Function model which only takes about 1.5 seconds.

1. Introduction

Rapid Developments in computer network technology are currently experiencing significant changes, especially with the establishment of SDN [1]. SDN is the latest architecture that is used to replace the traditional scheme on the network. SDN architecture separates the control plane and data plane, the separation aims to simplify the network resources management (abstraction layer) and to reduce the tasks performed by network administrators [2], [3]. The concept that distinguishes traditional architecture from SDN is the centralization of the network with all network settings configured in the control plane. The control plane itself in the SDN architecture functions as a controller that has an obligation for regulating and monitoring all data flows in the network. The control plane is responsible for the network configuration while the data plane runs the configuration [4]. The SDN architecture has three layers, namely the infrastructure plane layer or the data plane layer, the control plane layer, and the application layer. The data and control plane layer are connected by the southbound interface (OpenFlow) while the control plane layer and application layer are regulated by the northbound interface (API) [5]–[7]. The SDN architecture itself has been implemented into various kinds of networks and one of them was the IoT. Currently, IoT devices have developed exponentially and can be found in everyday life. IoT consists of billions of devices that are interconnected with each other applied into various sectors e.g. public facilities, household appliances, medical equipment, transportation [8]. According to the latest report from Juniper's Research, the number of IoT devices in 2021 will reach 46 billion and continue to increase in the future. With so many IoT devices connected to the internet, the security gaps will vary greatly and can be exploited by irresponsible people (attackers). The integration of SDN and IoT is a potentially viable solution to strengthen IoT's management and control capabilities namely SD-IoT. IoT can take advantage of the SDN to have a centralized control, abstraction of network devices, and flexibility [8]. The SD-IoT controller can be easily programmed according to the needs of network administrators. Although this layered network architecture provides more flexible configuration and capabilities than traditional networks, it increases the likelihood of attacks, especially the controller [9]. One of the attacks that could impact the SD-IoT integration model is DDoS. In general, DDoS attacks are classified into two types, namely High Rate Distributed Denial Of Service (HRDDoS) and Low Rate Distributed Denial Of Service (LRDDoS) [9], [10]. LRDDoS attacks initiated on data fields are characterized by low speed, concealment, and persistence, which make them difficult to detect. This is because the LRDDoS attack is hidden in the normal data flow. LRDDoS attacks for the data layer are quite different from HRDDoS attacks for the control layer. LRDDoS attacks only launch attacks at a lower rate by controlling a smaller number of bots. With a low attack traffic rate, the LRDDoS attacks are difficult to detect and can have an impact on links that are directly connected to the controller. Upon receiving the attack, the controller is forced to exceed the normal limit, the controller will process the incoming dummy packet so that it consumes all available resources and causes the controller to become unstable. Therefore, it is necessary to have an Intrusion Detection System (IDS) to overcome security holes that exist by LRDDoS attacks.

In previous research, the method used for detection and mitigation has been proposed. The authors in [8] used the SD-IoT Framework to detect and mitigate DDoS attacks using the proposed algorithm, namely the cosine similarity of the message vector rate of incoming packets through the switch port on SD-IoT. The proposed algorithm was compared with the other two algorithms and had good results in mitigating attacks. Research [9] proposed a machine learning-based low rate DDoS attack detection system (MLDD) by using the stateful and stateless features of the Openflow package to identify attack traffic on the SDN controller. Researchers used datasets generated in real time from the experimental network environment. The data collected from the controller and switch contained 204,888 and 48,509 data records. The researcher trained the classifier on the training set with 80% of the combined normal dataset and attack traffic, then the researcher tested the classification accuracy on the test set from the remaining datasets. The researcher used the Support Vector Machine (SVM) algorithm with Radial Basis Function (RBF), KNN, Multinomial Naïve Bayes (NB), and Random Forest (RF) kernels using Gini impurity. Classifier accuracy ranged from 79% to 100% on controllers and from 65% to 100% on switches. Thus, the test results in the controller were better than the switch, especially for using stateless feature datasets. The SVM, KNN and RF algorithms achieved the same accuracy results on the controller with an accuracy rate of 97% and on the switch the RF accuracy rate was slightly faster than SVM and KNN which have a difference of about 1% and 2% by using the stateless feature. While the NB algorithm had the worst accuracy rate among other algorithms with an accuracy rate of 79% on the controller and 66% on the switch. Furthermore, research [1] proposed a reactive application-based solution that could identify, detect, and mitigate attacks comprehensively. In his research, an application had been built using machine learning algorithms including, Support Vector Machine (SVM) with the Linear and RBF kernel, K-Nearest Neighbor (KNN), Decision Tree (DTC), Random Forest (RFC), Multi-Layer Perceptron (MLP), and Gaussian Naïve Bayes (GNB). The dataset was a dataset from the University of Muhammadiyah Malang which was compiled by Oxicusa Gugi Housman in 2020 [11]. In terms of the data proportion, the number of train data was 420,000 data, while the test data was divided into two different schemes, namely 18,000 and 36,000 data tests. Experiments were carried out using two different scenarios including a classification scheme without an SDN controller and using an SDN controller. The results of the experiment using 36,000 datasets without a controller both RBF and Linear SVM algorithms achieved the highest results with accuracy, precision, recall and f1 average at 100%, while the MLP algorithm produced the worst results among other algorithms with accuracy, precision, recall, and f1 averaged at 50.4%. In the experimental results using 36,000 datasets using the SDN controller, the SVM RBF and LINEAR algorithms still got the highest value at 100%. The comparative results analysis showed that the SVM kernel RBF or Linear algorithm produced the highest accuracy among several others. SVM was the most efficient method to identify DDoS attacks which had been proved with accuracy, precision and recall at around 100% which could be considered as the main algorithm for detecting DDoS attacks.

Based on previous research, the contributions made in this research are listed as follows:

- The implementation of the Intrusion Detection System application for LRDDoS attacks with machine learning methods using the SVM algorithm (RBF and Linear kernels).
- Proposing a new dataset scheme with 22 features.
- Using the Feature Importance method, namely the Logistic Regression Coefficient to carry out the feature selection process in order to ease the performance of the controller in the classification process and improve the final classification result.

2. Research Method

2.1 Emulation Topology

In this research, the testing phase was carried out using emulation on mininet-iot as an emulator software [12] on the Ubuntu 20.04 Operating System. The hardware specifications of the emulated OS were composed of Intel core i5-3337U, Nvidia GeForce 710M, and 8GB of RAM. The topology used is a tree topology (depth=3 and fanout=2) as a network architecture consisting of 1 RYU Controller [13], 7 Open Virtual Switch (OvS) [14], interconnected using OpenFlow [15] version 1.3.0, and 8 hosts. It can be seen in Figure 1 that the network topology shows H1 as an attacker who sends 39994 packets of LRDDoS using Constrained Application Protocol (CoAP) [16] with tcpreplay tools [17] targeted to H6 as CoAP Server. The purpose of the attack was to exhaust the controller's performance. The LRDDoS attacks were carried out by sending 3 different packet rates which were divided into 50, 100, and 200 packet rates per second (pps).

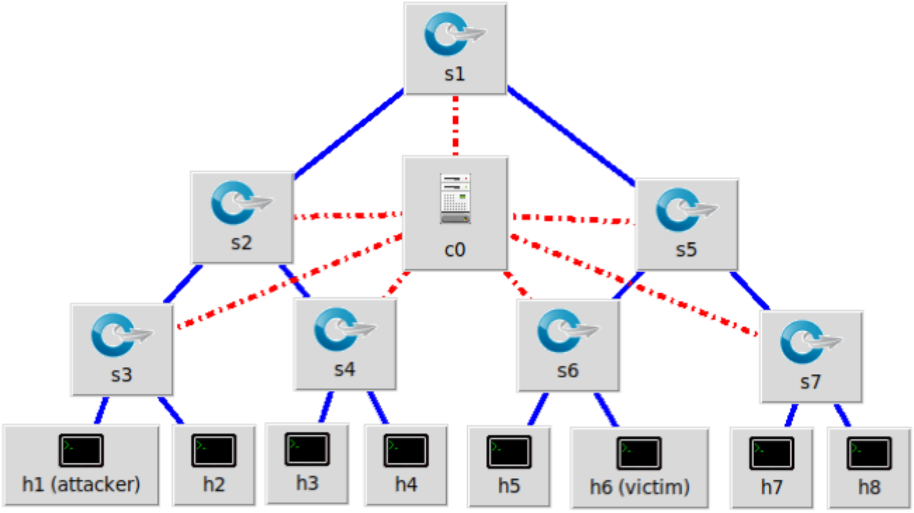

Figure 1. Emulation Topology

The test scheme was divided into two different scenarios which includes a prediction scheme without involving the SD-IoT controller and a classification using an SD-IoT controller. On the SD-IoT controller itself, a machine learning model with the SVM algorithm (RBF and Linear kernel) has been generated for the classification process [18]. Before implementing machine learning on the SD-IoT, the SVM algorithm was tested first without using the network, then the results were extracted and utilized on the SD-IoT network controller.

2.2 Data Preprocessing (Feature Importance Logistic Regression Coefficient)

The dataset used during the research was considered as a new dataset schema that utilized several features available in the OpenFlow protocol [19]. The data extraction process employed IPv4, TCP, and UDP Header information as well as some information about port statistics extracted from OFPMP_PORT_STATS requests. The data was divided into two kinds of data, namely the train data which contained 160,006 packets and the test data which contained 39,994 packets. The total dataset was 200,000 packets with DDoS and Normal label separated equally (100,000 packets). There were 22 features in total on the dataset. The list of these features can be seen on Table 1.

Table 1. Feature's List

Feature's Name	Feature's Origin
datapath_id	OFPT_PACKET_IN
version	IPv4's Header
header_length	IPv4's Header
tos	IPv4's Header
total_length	IPv4's Header
flags	IPv4's Header
offset	IPv4's Header
ttl	IPv4's Header
proto	IPv4's Header
csum	IPv4's Header
src_ip	IPv4's Header
dst_ip	IPv4's Header
src_port	UDP's/TCP's Header
dst_port	UDP's/TCP's Header
port_no	OFPPortStatsReply
rx_bytes_ave	OFPPortStatsReply (rx_bytes / rx_packets)
rx_error_ave	OFPPortStatsReply (rx_bytes / rx_packets)
rx_dropped_ave	OFPPortStatsReply (rx_bytes / rx_packets)
tx_bytes_ave	OFPPortStatsReply (tx_bytes / tx_packets)
tx_error_ave	OFPPortStatsReply (tx_bytes / tx_packets)
tx_dropped_ave	OFPPortStatsReply (tx_bytes / tx_packets)

The large number of features could have an impact on the controller, because it required more resources from the controller in carrying out the classification process [20]. Therefore, it is necessary to have a feature selection process for easing the burden on the Controller [21]. This research applied the Feature Importance method [22], namely the Logistic Regression Coefficient for performing the feature selection process [23]–[25]. With this Feature Importance, only certain features were processed to reduce the controller resource utilization. Figure 2 shows the Feature Importance Logistic Regression Coefficient process implemented with the SVM algorithm to create a model that was used on the SD-IoT network. Feature Importance refers to a technique that applies a score to an input feature based on how useful the feature is in predicting the target variable [26]. The Feature Importance Logistic Regression Coefficient score can be calculated for problems involving prediction of numerical values, called regression, and problems involving prediction of class labels, called classification [27]–[29]. Feature Importance plays an important role in modeling for prediction. The selection of the right features can increase efficiency and effectiveness in the prediction model. The Feature Importance score sorts out which features are the most relevant and which features are irrelevant, so that relevant features can be identified for the prediction process to improve model performance.

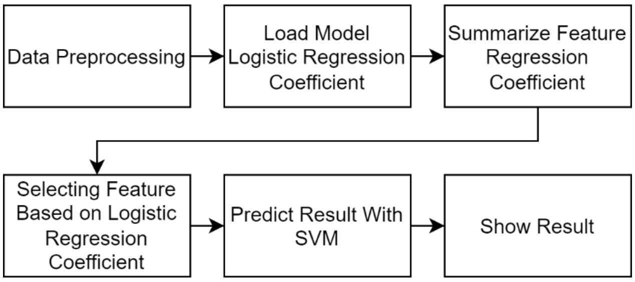
Data Preprocessing
Load Model Logistic Regression Coefficient
Summarize Feature Regression Coefficient
Selecting Feature Based on Logistic Regression Coefficient
Predict Result With SVM
Show Result

Figure 2. Logistic Regression Coefficient Block Diagram

Table 2 shows that there are 8 features that are relevant to be used during the classification process. The feature is selected based on the coefficient value less than 0 and more than 0. If the coefficient value was 0 then the feature was dropped or not used on the classification process.

Table 2. Score Features Based on Logistic Regression Coefficient

Feature Name	Logistic Regression Coefficient Score
datapath_id	0
version	0
header_length	0
tos	0
total_length	-16.172
flags	67.658
offset	0
ttl	0
proto	0
csum	-0.00195
src_ip	-185.064
dst_ip	0
src_port	-0.26961
dst_port	0
port_no	-0.08737
rx_bytes_ave	30.446
rx_error_ave	0
rx_dropped_ave	0
tx_bytes_ave	0.08789
tx_error_ave	0
tx_dropped_ave	0

2.3 Classification Process

When the switch received an incoming packet, the switch performed the filtering of incoming packets by matching the packet header with Flowrule. If no match was found, the switch automatically considered the packet as a new packet [30]–[32]. The packet was considered new because there was no mapping for the IP and MAC address of the packet on the switch. Then the new packet was encapsulated and sent to the controller by sending an OFPT_PACKET_IN message as shown in Figure 3. The OFPT_PACKET_IN message could be received by the controller when there was a table miss event according to the OpenFlow protocol standard for networking discovery purposes. When the OFPT_PACKET_IN message entered the controller, the controller analyzed the incoming packet header, then processed the packet header information and made the packet a reference for the classification process. During the classification process, packets were classified into 2 types, namely Normal or DDoS using the SVM algorithm as a classification model combined with the Feature Importance Logistic Regression Coefficient method as the relevant feature selector in order to ease the burden on the controller. If the incoming packet was detected as a normal packet, the packet was notified as a normal flow then the packet was transferred for packet processing on the learning switch application. However, if the incoming packet was detected as a DDoS packet then the packet was classified as an LRDDoS attack and was forwarded to the DDoS packet notification process.

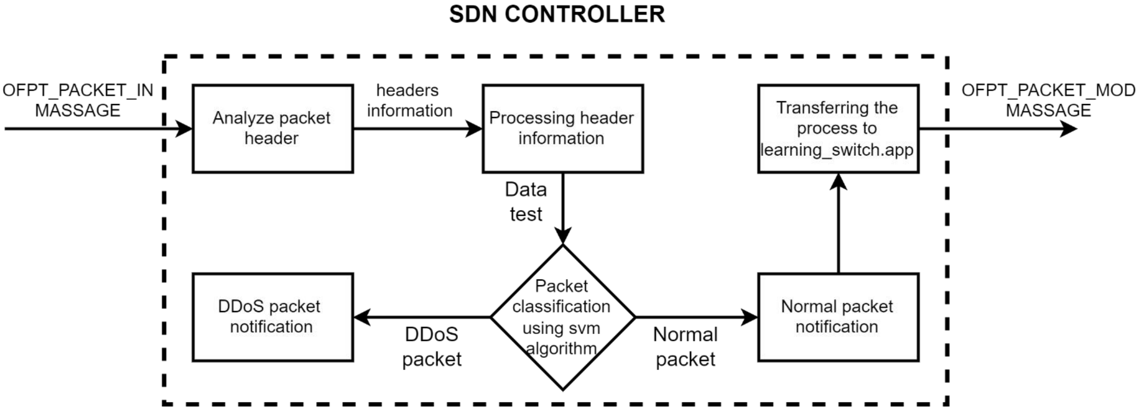

Figure 3. SD-IoT Controller Block Diagram

After the model was completed, the model was used on the controller in the SD-IoT network, then the data extraction process began as shown in Figure 4. The data extraction process started from the attacker by sending an attack containing test data to the victim, then the test data was processed by the SD-IoT switch. The data extraction process utilized the information that has been presented in the previous sub-chapter. The extracted data was the result of the classification of DDoS packages or Normal packages using models that have been made previously using SVM and Feature Importance. The DDoS packets were labeled as “0” while the Normal package was represented as “1”. The results of the classification data were stored on a *.csv file. The results of the classification carried out by the controller were analyzed using several test variables including training time, accuracy, precision, recall, F1 score, and prediction loss. The prediction loss is the value obtained when there is a packet loss or failure on packet’s categorization.

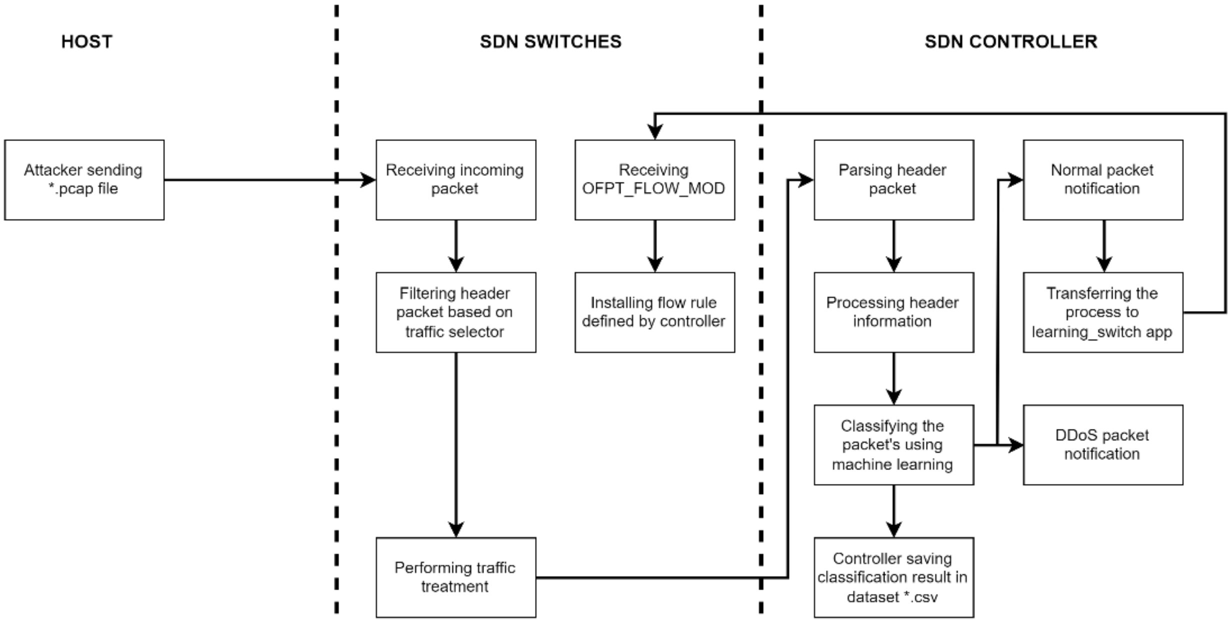
HOST
SDN SWITCHES
SDN CONTROLLER
Attacker sending *.pcap file
Receiving incoming packet
Receiving OFPT_FLOW_MOD
Parsing header packet
Normal packet notification
Filtering header packet based on traffic selector
Installing flow rule defined by controller
Processing header information
Transferring the process to learning_switch app
Classifying the packet's using machine learning
DDoS packet notification
Performing traffic treatment
Controller saving classification result in dataset *.csv

Figure 4. Extraction of Dataset

3. Results and Discussion

3.1 Training and Classification Result Without SD-IoT

The main puspose of the training deployment process without the SD-IoT network was directed to generate the best classification model. This model was pointed as a classification resource during the real emulated attack deployed in SD-IoT environment. Therefore, the controller only performed the classification process without training the model.

Based on Table 3, the experiment of classification model without using SD-IoT, it can be seen that the Linear SVM model acquires perfect results with an average value of 100% for accuracy, precision, recall, and f1-score results. As for the SVM RBF model, the accuracy is 1% lower than the SVM Linear with a value of 99% and 100% for precision, recall, and f1-score. In the process of learning the model itself, Linear SVM takes longer than RBF SVM. The SVM Linear model takes about 23.6 seconds while the SVM RBF model takes about 1.5 seconds for performing data trains.

Table 3. Experiments Result Without SD-IoT

SVM Kernel	Accuracy %	Precision %	Recall %	F1 %	Training Time (s)
Linear	100	100	100	100	23.648
RBF	99	100	100	100	1.566

3.2 Classification Result in SD-IoT Network

The classification model that has been generated from the results of the training without an SD-IoT network, was utilized by the SD-IoT controller for classifying the real emulated LRDDoS packets. As shown in Table 4, the results of testing the SVM algorithm using SD-IoT were carried out with several kinds of packet delivery rates to detect the low-rate attacks. The packet rate was divided into 3 parts, namely 50, 100, and 200 packets per second. Based on Table 4, it can be seen the results of the two algorithms, where the Linear SVM model still has the highest value with the results of accuracy, precision, recall, and f1-score each at a value of 100%. The prediction loss value in the Linear model obtains the highest value at a delivery rate of 200 pps, at 99.5% while the lowest value at a delivery rate of 100 pps at 96%. The prediction loss average result of all delivery rates obtained is 97.8%. Meanwhile, for the SVM RBF model, the accuracy results for each delivery rate produce different values, the highest accuracy is at 200 pps delivery rate with 100% accuracy, the lowest accuracy is 50 pps with 53.3% accuracy, and the average accuracy result obtained from all rates is 74.3%. The highest Prediction Loss obtained in the RBF model is 99.7% at a delivery rate of 200 pps, the lowest value is at 98.8% with a delivery rate of 50 pps, and the average prediction loss result from all delivery rates is 99.2%.

Prediction Loss occurred when a new packet came in, but the controller still performed the classification process of the previous dummy packet. When the Packet In message was received by the controller, the controller indirectly triggered the classification process of the previous packet, so the impact on the controller was overloaded and the prediction process took longer. This activity caused the prediction process to be repeated several times with the same IP. The controller should predict the new packet but the controller was still predicting the old packet. The greater the loss percentage that occurred in the SVM algorithm, the accuracy also increased. Because the data used as a reference in the classification process were fewer in number and could not represent the entire data population.

Table 4. Experiments Result With SD-IoT

SVM Kernel	Packet Rate (pps)	Accuracy %	Precision %	Recall %	F1 %	Prediction Loss %
Linear	50	100	100	100	100	97.9
	100	100	100	100	100	96.0
	200	100	100	100	100	99.5
RBF	50	53.3	53.3	53.3	53.3	98.8
	100	69.7	69.7	69.7	69.7	99.3
	200	100	100	100	100	99.7

4. Conclusion

LRDDoS attacks are one of the significant problems in SD-IoT networks in terms of centralized control management. The author proposed a new dataset scheme in which there are 22 features in total, then used the Feature Importance method, namely the Logistic Regression Coefficient to perform the feature selection process for reducing the performance of the controller during the classification process and improved the final classification result. Based on the results that have been carried out and analyzed comparatively, it can be concluded that the SVM algorithm with the Linear kernel is the most accurate model for predicting and classifying LRDDoS attacks. Linear kernels get the highest results in the test scheme without and with SD-IoT with 100% accuracy, but the training time required for this model is longer, about 23.6 seconds compared to the RBF model which only takes about 1.5 seconds. Although the RBF kernel produces a faster training time, the accuracy results obtained from the testing scheme using SD-IoT are much lower than the Linear kernel with an average accuracy value 74.3%. For further research, researchers will develop an Intrusion

Prevention System (IPS) application to test the LRDDoS attack mitigation process with a model that has been analyzed using a flow modification mechanism on the SD-IoT network. Therefore, it can be pointed out which algorithm that can run effectively assessed based on system performance as an IPS.

Acknowledgement

The authors would like to express their profound gratitude for the support provided by the Informatics Laboratory and Informatic Department in University of Muhammadiyah Malang.

References

[1] F. D. Setiawan Sumadi and C. S. Kusuma Aditya, “Comparative Analysis of DDoS Detection Techniques Based on Machine Learning in OpenFlow Network,” in *2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, Dec. 2020, pp. 152–157, doi: https://doi.org/10.1109/ISRITI51436.2020.9315510.

[2] Kilwalaga, I. F., Sumadi, F. D. S., & Syaifuddin, S. (2020). SDN-Honeypot Integration for DDoS Detection Scheme Using Entropy. Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 5(3), 187-194. https://doi.org/10.22219/kinetik.v5i3.1058.

[3] K. Nisar *et al.*, “A Survey on The Architecture, Application, and Security of Software Defined Networking: Challenges and Open Issues,” *Internet of Things*, vol. 12, p. 100289, Dec. 2020, doi: https://doi.org/10.1016/j.iot.2020.100289.

[4] M. Alsaeedi, M. M. Mohamad, and A. A. Al-Roubaiey, “Toward Adaptive and Scalable OpenFlow-SDN Flow Control: A Survey,” *IEEE Access*, vol. 7, pp. 107346–107379, 2019, doi: https://doi.org/10.1109/ACCESS.2019.2932422.

[5] T. Li, J. Chen, and H. Fu, “Application Scenarios based on SDN: An Overview,” *J. Phys. Conf. Ser.*, vol. 1187, no. 5, p. 052067, Apr. 2019, doi: https://doi.org/10.1088/1742-6596/1187/5/052067.

[6] L. Ben Azzouz and I. Jamai, “SDN, Slicing, and NFV Paradigms for A Smart Home: A Comprehensive Survey,” *Trans. Emerg. Telecommun. Technol.*, vol. 30, no. 10, pp. 1–13, Oct. 2019, doi: https://doi.org/10.1002/ett.3744.

[7] P. P. Ray and N. Kumar, “SDN/NFV Architectures for Edge-Cloud Oriented IoT: A Systematic Review,” *Comput. Commun.*, vol. 169, no. June 2020, pp. 129–153, Mar. 2021, doi: https://doi.org/10.1016/j.comcom.2021.01.018.

[8] D. Yin, L. Zhang, and K. Yang, “A DDoS Attack Detection and Mitigation With Software-Defined Internet of Things Framework,” *IEEE Access*, vol. 6, no. Mcc, pp. 24694–24705, 2018, doi: https://doi.org/10.1109/ACCESS.2018.2831284.

[9] 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,” *Int. J. Sens. Networks*, vol. 34, no. 1, p. 56, 2020, doi: https://doi.org/10.1504/IJSNET.2020.109720.

[10] S. Xie, C. Xing, G. Zhang, and J. Zhao, “A Table Overflow LDoS Attack Defending Mechanism in Software-Defined Networks,” *Secur. Commun. Networks*, vol. 2021, pp. 1–16, Jan. 2021, doi: https://doi.org/10.1155/2021/6667922.

[11] O. Gugi Housman, H. Isnaini, and F. Sumadi, “SDN-DDOS (ICMP,TCP,UDP).” Mendeley Data, p. V1, 2020, doi: https://doi.org/10.17632/hkjbp67rsc.1.

[12] D. Y. Setiawan, S. Naning Hertiana, and R. M. Negara, “6LoWPAN Performance Analysis of IoT Software-Defined-Network-Based Using Mininet-Io,” in *2020 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS)*, Jan. 2021, pp. 60–65, doi: https://doi.org/10.1109/IoTaIS50849.2021.9359714.

[13] S. Asadollahi, B. Goswami, and M. Sameer, “Ryu Controller’s Scalability Experiment on Software Defined Networks,” in *2018 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC)*, Feb. 2018, pp. 1–5, doi: https://doi.org/10.1109/ICCTAC.2018.8370397.

[14] M. Ushakova, Y. Ushakov, J. Cui, L. Legashev, A. Shukhman, and A. Bolodurin, “Research of Performance Parameters of Virtual Switches with OpenFlow Support,” in *2020 International Conference Engineering and Telecommunication (En&T)*, Nov. 2020, pp. 1–4, doi: https://doi.org/10.1109/EnT50437.2020.9431289.

[15] R. Wazirali, R. Ahmad, and S. Alhiyari, “SDN-OpenFlow Topology Discovery: An Overview of Performance Issues,” *Appl. Sci.*, vol. 11, no. 15, p. 6999, Jul. 2021, doi: https://doi.org/10.3390/app11156999.

[16] E. Al-Masri *et al.*, “Investigating Messaging Protocols for the Internet of Things (IoT),” *IEEE Access*, vol. 8, pp. 94880–94911, 2020, doi: https://doi.org/10.1109/ACCESS.2020.2993363.

[17] J. Singh and S. Behal, “Detection and Mitigation of DDoS Attacks in SDN: A Comprehensive Review, Research Challenges and Future Directions,” *Comput. Sci. Rev.*, vol. 37, p. 100279, Aug. 2020, doi: https://doi.org/10.1016/j.cosrev.2020.100279.

[18] J. Sengupta, S. Ruj, and S. Das Bit, “A Comprehensive Survey on Attacks, Security Issues and Blockchain Solutions for IoT and IIoT,” *J. Netw. Comput. Appl.*, vol. 149, p. 102481, Jan. 2020, doi: https://doi.org/10.1016/j.jnca.2019.102481.

[19] Open Networking Foundation, “OpenFlow Switch Specification (Version 1.5.1),” *Current*, vol. 0, pp. 1–36, 2015, [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.1.pdf.

[20] M. P. Singh and A. Bhandari, “New-flow Based DDoS Attacks in SDN: Taxonomy, Rationales, and Research Challenges,” *Comput. Commun.*, vol. 154, no. October 2019, pp. 509–527, Mar. 2020, doi: https://doi.org/10.1016/j.comcom.2020.02.085.

[21] U. M. Khaire and R. Dhanalakshmi, “Stability of Feature Selection Algorithm: A Review,” *J. King Saud Univ. - Comput. Inf. Sci.*, no. xxxx, Jun. 2019, doi: https://doi.org/10.1016/j.jksuci.2019.06.012.

[22] A. A. Megantara and T. Ahmad, “Feature Importance Ranking for Increasing Performance of Intrusion Detection System,” in *2020 3rd International Conference on Computer and Informatics Engineering (IC2IE)*, Sep. 2020, pp. 37–42, doi: https://doi.org/10.1109/IC2IE50715.2020.9274570.

[23] X. Zou, Y. Hu, Z. Tian, and K. Shen, “Logistic Regression Model Optimization and Case Analysis,” in *2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)*, Oct. 2019, pp. 135–139, doi: https://doi.org/10.1109/ICCSNT47585.2019.8962457.

[24] E. Y. Boateng and D. A. Abaye, “A Review of the Logistic Regression Model with Emphasis on Medical Research,” *J. Data Anal. Inf. Process.*, vol. 07, no. 04, pp. 190–207, 2019, doi: https://doi.org/10.4236/jdaip.2019.74012.

[25] H. Hemasinghe, R. S. S. Rangali, N. L. Deshapriya, and L. Samarakoon, “Landslide Susceptibility Mapping Using Logistic Regression Model (A Case Study in Badulla District, Sri Lanka),” *Procedia Eng.*, vol. 212, pp. 1046–1053, 2018, doi: https://doi.org/10.1016/j.proeng.2018.01.135.

[26] G. Eraslan, Ž. Avsec, J. Gagneur, and F. J. Theis, “Deep Learning: New Computational Modelling Techniques for Genomics,” *Nat. Rev. Genet.*, vol. 20, no. 7, pp. 389–403, Jul. 2019, doi: https://doi.org/10.1038/s41576-019-0122-6.

[27] G. Heinze, C. Wallisch, and D. Dunkler, “Variable Selection - A Review and Recommendations for The Practicing Statistician,” *Biometrical J.*, vol. 60, no. 3, pp. 431–449, May 2018, doi: https://doi.org/10.1002/bimj.201700067.

[28] M. E. Shipe, S. A. Deppen, F. Farjah, and E. L. Grogan, “Developing Prediction Models for Clinical Use Using Logistic Regression: An Overview,” *J. Thorac. Dis.*, vol. 11, no. S4, pp. S574–S584, Mar. 2019, doi: https://doi.org/10.21037/jtd.2019.01.25.

[29] C. Gambella, B. Ghaddar, and J. Naoum-Sawaya, “Optimization Problems for Machine Learning: A Survey,” *Eur. J. Oper. Res.*, vol. 290, no. 3, pp. 807–828, May 2021, doi: https://doi.org/10.1016/j.ejor.2020.08.045.

[30] M. A. Aladaileh, M. Anbar, I. H. Hasbullah, Y.-W. Chong, and Y. K. Sanjalawe, “Detection Techniques of Distributed Denial of Service Attacks on Software-Defined Networking Controller–A Review,” *IEEE Access*, vol. 8, pp. 143985–143995, 2020, doi: https://doi.org/10.1109/ACCESS.2020.3013998.

[31] B. Isyaku, M. S. Mohd Zahid, M. Bte Kamat, K. Abu Bakar, and F. A. Ghaleb, “Software Defined Networking Flow Table Management of OpenFlow Switches Performance and Security Challenges: A Survey,” *Futur. Internet*, vol. 12, no. 9, p. 147, Aug. 2020, doi: https://doi.org/10.3390/fi12090147.

[32] S. Kotey, E. Tchao, and J. Gadze, “On Distributed Denial of Service Current Defense Schemes,” *Technologies*, vol. 7, no. 1, p. 19, Jan. 2019, doi: https://doi.org/10.3390/technologies7010019.