Denial of Service Traffic Validation Using K-Fold Cross Validation on Software Defined Network

Abdhul Rhohim[1], Vera Suryani[2], Muhammad Arief Nugroho[3]

[1,2,3] Telkom University, Bandung
[1]rhohim@students.telkomuniversity.ac.id, [2] verasuryani@telkomuniversity.ac.id,
[3]arif.nugroho@telkomuniversity.ac.id

Abstrak

SDN *(Software Defined Network)* adalah sebuah teknologi baru dalam jaringan yang telah hadir dan terus dikembangkan. Dalam teknologi ini masih terdapat kemungkinan adanya serangan DoS. Namun riset deteksi serangan DoS masih menggunakan dataset NSL-KDD dan tidak melakukan validasi dari model atau algoritma yang digunakan, sedangkan untuk mengetahui seberapa baik sebuah model atau algoritma tersebut perlu dilakukan proses validasi. Maka dengan hal itu perlu dilakukan validasi dan juga menggunakan dataset yang berasal dari jaringan SDN. Metode validasi yang digunakan dalam riset ini adalah *K-Fold Cross Validation.* Pada validasi ini akan ada pengulangan proses sesuai nilai dari K dan hasilnya diambil dari nilai rata-rata. Dari hasil pengujian diperoleh nilai validasi sebesar 99,79 % dari algoritma Support Vector Machinne (SVM), 99.84% dari Decision Tree dan 96.84% untuk Naïve Bayes yang dijadikan model, artinya nilai tersebut menunjukan model yang digunakan dapat melakukan deteksi DoS dengan sangat baik dan model SVM memiliki hasil yang tinggi dibawah Decision Tree.

Kata kunci : SDN, Validasi, Cross Validation, DoS, NSL-KDD

Abstract

SDN (Software Defined Network) is a technology in networking that has been present and continues to be developed. In this technology there is still the possibility of a DoS attack. However, DoS attack detection research still uses the NSL-KDD dataset and does not validate the model used, while to find out how good a model is, a validation process needs to be carried out. So with that it is necessary to validate and also use datasets originating from the SDN network. The validation method used in this research is K-Fold Cross Validation. In this validation there is a repetition of the process according to the value of K and the results are taken from the average value. From the test results obtained a validation value of 99.79% from the Support Vector Machine (SVM), 99.84% from Decision Tree and 96.84% for Naïve Bayes which is used as a model, meaning that this value indicates the model used can perform DoS detection very well and SVM model has high score under Decision Tree model.

Keywords: SDN, Validation, Cross Validation, DoS, NLS-KDD

1. Preliminary

Background

Software Defined Network (SDN) is a new technology in the concept of computer networking, where SDN can separate the data plane and control plane. However, SDN still has potential for Denial of Service (DoS) attacks. This happens because SDN performs network control centrally through a controller [6]. Denial of Service (DoS) attacks are among common attacks on the network. DoS attacks prevent users from normal service access, due to the excessive consumption of network resources, memory, processor, etc. The most common DoS attacks are when attackers “flood” the network with many request at the same time, making the server unavailable to answer to that many request[1]. A lot of research has conducted the detection and mitigation of DoS attacks, in which research on DoS attack detection is carried out using machine learning models or algorithms trained by the NSL-KDD dataset. This dataset is often used for DoS detection because some of the attributes in the dataset are closely related to DoS. Research that has been done often concludes the results on the accuracy of the model obtained after being trained without validating it. therefore, it is necessary to validate the model used and has been trained by datasets taken from the SDN network, so that the accuracy results of the model can be said to be more precise.

In [2] that research was conducted to detect DoS attacks using the NSL-KDD dataset. The research detects DoS attacks using Probabilistic Neural Network (PNN) algorithm as a model that has been trained with NSL-KDD dataset, the accuracy of that model is 98.06%. That means based on the research, the NSL-KDD can be used to train the model. in order to detect DoS attacks. in [3] that research was conducted to detect DoS attacks on SDN networks using the NSL-KDD dataset. In this research, the models used were Decision Tree and Nave Bayes, where in this research was conducted using SDN and using the NSL-KDD dataset as the dataset. The accuracy score of the models used include 99.0% for the decision tree and 97.0% for nave Bayes. Research was also conducted [10], where in this research the model used was Deep Neural Network (DNN). The result of this model is 75.75%. Based on these studies, it can be concluded that the NSL-KDD dataset can used to detect DoS attacks on SDN networks. Because in [[2],[3],[10]] used NSL-KDD for the dataset to detect DoS on the SDN and the features on the NSL-KDD can classify DoS attacks and dataset in this research from Canadian Institute for Cybersecurity on UNB.

In [4] conducted a research using data train results from the SDN network data generation that had been made and classified using algorithms without validating the model. Meanwhile, one way that we can see how good the model we choose is by validating it. So from this research it can be stated that the model used has not been validated so it cannot be said that the model is good for classification. Based on these problems, this research validates a model that is used to detect DoS attacks. This validation will use machine learning validation, namely K-Fold Cross Validation. This method was chosen because match with the model to be used and the dataset. This research will use the dataset resulting from capturing traffic on the Software Defined Network (SDN) network and will also use the NSL-KDD dataset as a reference, so that it can be used by later modeling. The purpose of this research is to prove how well a model is used for DoS attack detection through the results obtained and find out how to validate the model so that it can be used for other models.

Problem Statement and Problem Identification

As discussed in the previous section, the validation process has never been carried out on a SDN network and also these research as a reference only used accuracy from the model to assess the algorithm perform. While the validation process is one way to find out how well the model is used. So with that, in this research validation is carried out on the SDN network and using K-Fold Cross Validation to know algorithm perform, this method is a validation method commonly used in machine learning and also this method is compatible with the machine learning algorithm and the dataset that used in this research. This research will use datasets obtained from the Software Defined Network (SDN). The dataset will also be shaped like the NSL-KDD, this is because that dataset has been used in several research to conduct training on DoS detection models on Software Defined Network (SDN). In this research, there are 12 features of NSL-KDD dataset that are used as a reference. This research will only carry out the validation process. where the results of the validation process will show a good model used for detection DoS.

Purposes

The purpose of this research is to find out how to carry out the validation process and also prove how well the model is used by looking at the results of the validation values using the K-Fold Cross Validation method. The relationship between objectives, testing and conclusions can be seen in Table 1.

Tabel 1. The relationship between objectives, testing and conclusions

No	Objectives	Testing	Conclusion
1	Knowing how to carry out the validation process and also proving how well the model is used to detect DoS by validating it using K-Fold Cross Validation,	The model used is SVM, Naïve Bayes and Decision Tree. the model is trained with datasets taken from the SDN network and validates using the K-Fold Cross Validation method.	results of that model in validation process are 99.79% for SVM, 99.84% for Decision Tree and 96.84% for Naïve Bayes. these results prove that the model is good for detecting. Although no the highest, it's because the SVM algorithm still has shortcomings in the optimization process.

Writing Sections

The next part is the research methodology which will discuss the things that support this research. After that section is the system built where in this section it will be explained about how to validate this research to get the results. The output of the system built section will be explained and analyze further in the evaluation section and the last section is conclusion which will conclude the results of this research.

2. Related Work

Software Defined Networking (SDN) is new idea in computer networking, promises to dramatically simplify control and management through network programmability. Computer networks are constructed from a lot of network devices, with many complex protocols (software), which are implemented and embedded on them. [7]. In the SDN Network there is still possibility of a DoS attack, where DoS determine a specific category of information where a malicious user blocks legitimate users from accessing network services by exhausting the resources of the victim system. DoS attacker make network congestion by generating a large volume of traffic in the area of the system. The size of that overload is enough to prevent any packet from reaching to destination. With that, research has been done on how to detect and mitigate DoS attacks, in research [2] conducted a research to detect DoS attacks using PNN algorithms, where the accuracy can be 98.06%, at [3] also research on the detection of DoS attacks in the SDN network using naïve bayes and decision tree algorithms which are 97.0% and 99.0% respectively. [10] also conducted the same research using deep neural network algorithms and obtained a result of 75.75%. all of the research was conducted using NSL-KDD datasets, then in the research [4] similar research was conducted using data train results from sdn network and classification using machine learning algorithm without validation process. [5] The dos attack traffic validation process is performed using mathematical methods and performed on Virtual Private Servers (VPS).

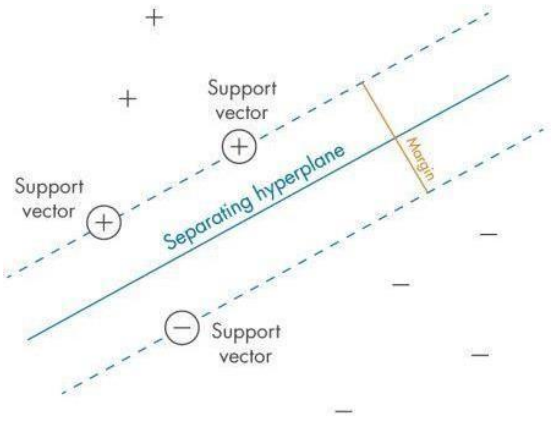
Therefore, this research will continue research [5] and at the same time know how to validate the model used for the detection of DoS attacks on SDN networks. By using the same dataset NSL-KDD as a reference dataset. NSL-KDD is a dataset of which it is an enhanced version and is also called the successor to the KDD Cup '99 dataset. NSL-KDD is an open source dataset so it can be downloaded easily. This dataset consists of 42 features classified into nominal, binary and numeric. In this dataset, there are two classes called normal and anomaly, along with the DoS features on the NSL-KDD which can be used in this research :

Tabel 2. Features on NSL-KDD

No	Feature	No	Feature
1	Duration	22	Is_Guest_Login
2	Protocol Type	23	Count
3	Service	24	Srv_Count
4	Flag	25	Serror_Rate
5	Src_Byte	26	Srv_Serror_Rate
6	Dst_Byte	27	Rerror_Rate
7	Land	28	Srv_Serror_Rate
8	Wrong_Fragment	29	Same_Srv_Rate
9	Urgent	30	Diff_Srv_Rate
10	Hot	31	Srv_Diff_Host_Rate
11	Num Failed Logins	32	Dst_Host_Count
12	Logged in	33	Dst_Host_Srv_Count
13	Num Compromised	34	Dst_Host_Same_Srv_Rate
14	Root Shell	35	Dst_Host_Diff_Srv_rate
15	Su_Attempted	36	Dst_Host_Same_Src_Port
16	Num_Root	37	Dst_Host_Srv_Diff_Host_rate
17	Num_File_Creations	38	Dst_Host_Serror_Rate
18	Num_Shells	39	Dst_Host_Srv_Serror_Rate
19	Num_Access_File	40	Dst_Host_Rerror_Rate
20	Num_outboundemds	41	Dst_Host_Srv_Rerror_Rate
21	Is_Host_Login	42	Class

The dataset feature will be used as a reference to change the shape of the resulting dataset from the SDN network, this is done so that the dataset can be used to train the model. In this research the model to be validated is SVM, Decision Tree and Naïve Bayes. this is because in [[11,[3]] the classification of DoS attacks with the NSL-KDD dataset has been carried out using that algorithm as the model used.

SVM is a model for the classification of both nonlinear nonlinear and linear data [14]. As shown in picture 1 SVM has a hyperplane and a margin value to maximize the classification.



Picture 1. Support Vector Machine Architecture

$$½||W||^2 \quad (1)$$

$$Y_i\ (X_1\ .\ W+B)\ -1\ \geq 0 \quad (2)$$

$$Yi\ (X_1\ .\ W_1 + X_2\ .\ W_2 + b\ ) \geq 1 \quad (3)$$

Equation (1) is the method used to create a hyperplane provided that equations (2) and (3) are met. While equations (4) and (5) are used to classify test data.

$$W_1\ .\ X_1 + W_2\ .\ X_2 + b\ \geq\ 1 \text{ for } Y_i = +1 \quad (4)$$

$$W_1\ .\ X_1 + W_2\ .\ X_2 + b\ \leq\ 1 \text{ for } Y_i = -1 \quad (5)$$

In this research, the Naïve Bayes algorithm and Decision Tree are also applied as models and comparisons of the models used, in [3] this algorithm has been used as a model to classify DoS and use the NSL-KDD dataset. With that, it can be concluded that this algorithm can also be used as a model. This Naïve Bayes algorithm uses the probability value of each attribute to make predictions. The equations used in the Naïve Bayes algorithm are described in equation (6).

$$P(H|X) = \frac{P(X|H).P(H)}{P(X)} P(X) \quad (6)$$

In equation (6) describes the equation used in the Naïve Bayes algorithm, the value of P(X|H) is probability value of attribute or feature with the class, the value of P(H) is the probability of the class and P(X) is the probability value that obtained from attributes or features. While the Decision Tree algorithm is a classification algorithm that utilizes the value of entrophy and information gain to make a decision. This algorithm changes the way to make decisions to be simpler. The equation for entrophy is described in equation (7)

$$\text{Entrophy}(S) = \sum_i^c -p_i \log_2 p_i \quad (7)$$

Where c is the number of detection classes, pi is the number of data belonging to that class. After getting the entrophy value, the next step is to calculate the information gain value. It aims to determine the root node that will be used in this algorithm.

$$\text{Gain}(S,A) = \text{Entrophy}(S) - \sum_{VgValue(A)} \frac{|Sv|}{|S|} Entrophy(Sv) \quad (8)$$

Equation (8) describes how to calculate the information gain. Where S is the entropy value that is owned, A is the attribute or feature that we want to calculate. |S| is the amount of data and |Sv| is the amount of data that A has.

To perform validation, this research will use K-Fold Cross Validation, the initial data are randomly partitioned to k. As seen in picture 2 and algorithm 1 this method is validation methods in machine learning where Cross validation is a technique to analyze whether a model has good generalizations (able to have good performance on unseen examples). In cross validation the original sample was divided into several subsample with as many partitions as K (K-fold) [9].The result of each iteration will then be calculated the average value using the equation (6).

Picture 2. K-Fold Cross Validation

Algorithm 1 for K-Fold Cross Validation

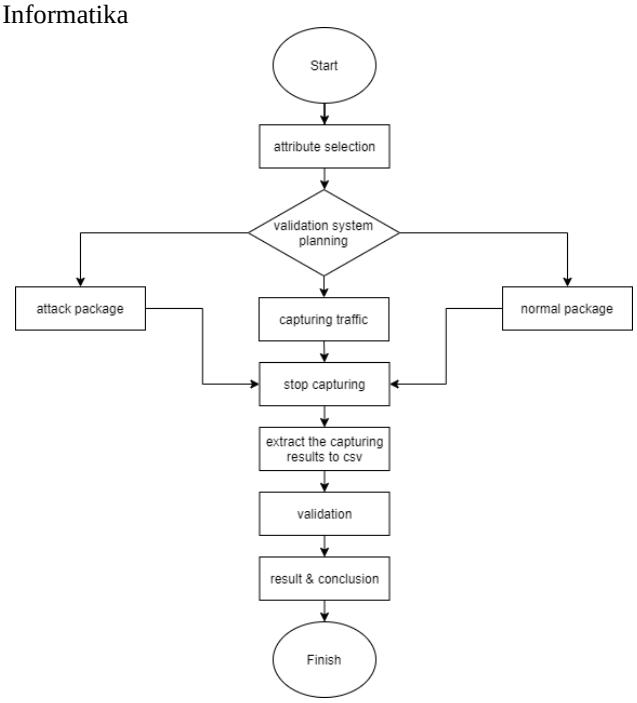
input:
training set S
set of parameter value P
learning algorithm A
integer K

output:
h_{p*} is a set of accuracy from validation process from P
E is mean for all of accuracy score and will be a validation score

partition S into $S_1, S_2, \ldots, S_k$
foreach $p \in P$
for i = 1 …. K
$h_{i,p} = A(S\backslash S_i ; p_i)$
endfor
endfor
$E = \frac{1}{K} \sum_{i=1}^{K} h_{i,p})$
return h_{p*} and E

3. System Built

Here is an overview of the system flow used in this research:



Picture 3. System Flow

Attribute Selection

The dataset used as a reference is the NSL-KDD dataset, this is because the dataset has been used to classify DoS attacks. Of the 42 features in the dataset, attribute selection will be made. It serves so that the features used have a very strong relationship with DoS. The attribute selection method in this research is information gain. Because information gain detects features that have the most information based on a certain class [5]. After doing attribute selection with information gain. Then the dataset features that will be used are as follows:

Tabel 3. The result of attribute selection

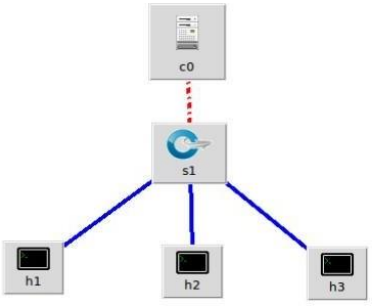
No	Feature Name	Description	Information Gain
1	src_bytes	Number of data that transferred from source to destination	0.8162
2	Service	Destination network that service used	0.6715
3	dst_bytes	Number of data that transferred from destination to source	0.6330
4	Flag	Status of connection	0.5193
5	diff_srv_rate	The percentage of connections with different services	0.5186
6	same_srv_rate	The percentage of connections with same service	0.5098
7	dst_host_srv_count	Number of connections that have same port number	0.4759
8	dst_host_same_srv_rate	Percentage of a connection that has same host and service	0.4382
9	dst_host_diff_srv_rate	Percentage of a connection that has different host and service	0.4109
10	dst_host_serror_rate	The percentage of connections that have activated flag s0, s1, s2	0.4059

		or s3, among the connections aggregated in dst_host_count	
11	logged_in	Login Status : 0 otherwise and 1 if successfully logged in	0.4047
12	dst_host_srv_serror_rate	The percent of connections that have activated flag (4) s0, s1, s2 or s3, among the connections aggregated in dst_host_srv_count (33)	0.3980
13	serror_rate	The percentage of connections that have activated the flag (4) s0, s1, s2 or s3, among the connections aggregated in count	0.3927
14	Count	Number of connections to the same destination host as the current connection in the past two second	0.3791
15	srv_serror_rate	The percentage of connections that have activated flag (4) s0, s1, s2 or s3, among the connections aggregated in srv_count	0.2708

From the results above, there are 15 features that have been successfully selected. The feature highlighted in red is not used because it is not identical to a DoS attack [5]. The selection of this feature is taken from the feature that has a minimum gain value of 0.3, because in [5] those features has 97% of F-Measurement and 0.0002 for False Positive Rate (FPR), which F-Measurement is the way for evaluation metrics based on precision and recall, for FPR is the important factors in design of any IDs [13].

Validation System Planning, Capturing, Normal Package and Attack Package

In this research, using a mininet emulator to create a network simulation which will later be used to obtain information as a dataset. Picture 4 is the topology used in this research and also based on research [8] and [12] that used same topology and simulate the system, it can be concluded that topology in the Figure 2 can also be used to perform simulation. In this case h1 will act as server, h2 will send normal packet and h3 will send attack packet. Sending attack packets will use hping3 and along with all that h1 will capture the network using wireshark. In this research also use non SDN network for comparing the validation result, the topology consist of three host and standalone switch with same simulation.



Picture 4. Tolopogy

Extract The Capturing Result to CSV

At this stage the results of the capturing which are still in the form of PCAP will be converted into CSV. After that, the results of the capturing in the form of CSV will be formatted according to the NSLKDD dataset using the header reference in accordance with table 3. The modified dataset will then be used for the validation process.

Validation

At this stage the captured data that has been formatted to be like the NSL-KDD dataset will be continued with the validation process. The data will be used as a dataset for modeling. After successfully modeling with the SVM algorithm, the validation process will then be carried out using K-Fold Cross Validation. The value of K will be set to 10 and the result will be averaged.

4. Evaluation

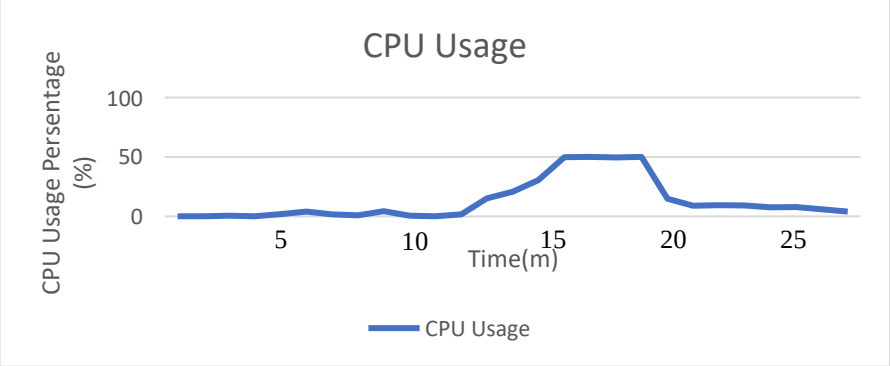
4.1 Test Result

From the classification results using the SVM , Naïve Bayes and Decision Tree, the accuracy are 99.87%, 96.79% and 99.96%. These results are good enough to be used as a model for detecting DoS attacks. Furthermore, from the accuracy results, a validation process was also carried out in the next test, using a K value of 10 and using the same dataset. The validation value of the model is 99.79% for SVM, 99.84% for Decision Tree and 96.84% for Naïve Bayes, this validation value also proves that the model used is proven to be good for detecting DoS Attacks. In addition, a confusion matrix and error rate process is also carried out, where TP is a record of detection of attacks that are detected as attacks, TN is a record of normal detection that is detected normally, FP is a record of normal detection of detected attacks and FN is a record of detection of attacks that are detected normally.

Tabel 4. Confusion Matrix and Error Rate

Algorithm	Accuracy	Error rate	TP	TN	FP	FN
Decision Tree	99.96%	0.04%	54751	65	12	8
Naïve Bayes	96.79%	3.21%	53002	77	0	1757
SVM	99.87%	0.15%	54754	16	61	5

In Figure 5 it can be seen the difference in CPU Usage traffic when receiving normal packets and when an attack occurs, when the attacker floods the victim by sending many packets so that the value of CPU Usage also increases. it also proves that the feature count is something that needs to be considered, because count is the number of packets sent in seconds and is in accordance with the character of DoS, which is to overwhelm its victims by sending many packets.



Picture 5. CPU Usage

4.2 Test Result Analysis

Based on the test results, this research proves that the SVM model has a good validation value for detecting DoS attacks. Although not to be the highest, it's because the SVM algorithm still has shortcomings in the optimitazion process. namely the two separate classes are not always perfectly separated, it causes the value of accuracy in SVM to still change a lot. As shown in table 5, the validation process also proves that there is a change in the accuracy value of each iteration carried out. This happened because the validation method used was random sampling and recalculated its accuracy. Also in that table comparing validation value between SDN and Non SDN network, the

value has different result because each network has different traffic. In table 4, the confusion matrix and error rate processes are also carried out, where it serves to support the value that the model used can properly detect DoS attacks. The table also shows that the SVM algorithm has a good enough value to detect DoS and also

Tabel 5. Accuration Result of the Model

Iteration	Result					
	SVM		Decision Tree		Naïve Bayes	
	SDN	Non SDN	SDN	Non SDN	SDN	Non SDN
Iteration 1	98.97%	92.38%	98.92%	92.37%	97.22%	22.89%
Iteration 2	99.91%	92.38%	99.84%	90.93%	97.01%	12.61%
Iteration 3	99.90%	92.38%	100 %	92.43%	97.52%	07.61%
Iteration 4	99.86%	92.38%	99.97%	92.46%	97.74%	12.44%
Iteration 5	99.87%	92.32%	99.97%	91.78%	96.10%	15.28%
Iteration 6	99.91%	92.38%	100%	90.59%	97.75%	10.97%
Iteration 7	99.88%	92.40%	99.94%	92.06%	95.58%	18.66%
Iteration 8	99.92%	92.40%	99.98%	90.81%	97.49%	12.56%
Iteration 9	99.86%	92.40%	99.96%	92.43%	97.37%	07.62%
Iteration 10	98.86%	92.40%	99.89%	93.10%	94.66%	14.67%

5. Conclusion

Based on the results that have been carried out a validation value of 99.79% from the Support Vector Machine (SVM), 99.84% from Decision Tree and 96.84% for Naïve Bayes which is used as a model, meaning that this value indicates the model used can perform DoS detection very well and SVM model has high score under Decision Tree model. it proves that the SVM algorithm is one of the algorithms that is capable and well used to detect DoS attacks, although still not the highest. In addition, the test results prove that the accuracy results are not enough to be used as a reference that an algorithm can be used properly for modelling. This is evidenced by the change in the value of each accuracy made during the validation process. Some suggestions that can be made for further research are to validate using another algorithm and more attacking tools.

References

[1] Masetic Z. 2017. A Review Of Machine Learning Techiques Efficiency In Dos Attack Detection. Internasional Burch University. Sarajevo. Bosnia amd Herzegovina. Vol 6, ISSN No 2277-8176. 461-462

[2] Cahyaningtyas A. 2019. Deteksi Serangan Denial Of Service (DoS) Menggunakan Alogaritma Pronabilistic Neural Network (PNN). E-proceeding of Engineering. Vol.6, No. 2, ISSN : 2355-9365

[3] Afif S. R, Sukarno P, dan Nugroho M. A. 2018. Analisis Perbandingan Logaritma Bayes dan Decision Tree Untuk Deteksi Serangan Denial of Service (DoS) pada Aristektur Software defined Network (SDN). Eproceeding of Engineering : Vol. 5, No. 2, ISSN: 2355-9365

[4] Panjaitan A. F. A, Sukarno P, dan Nugroho M. A. 2018. Pendeteksi DoS pada Controller Software Defined Networking Dengan Menggunakan Algoritma Berbasis Entropi. E-proceeding of Engineering : Vol. 5, No. 3, ISSN: 2355-9365

[5] Pratama A. S, Sukarno P, dan Nugroho M. A. 2019. Validasi Traffic Denial of Service pada Live Network. E-proceeding of Engineering : Vol. 6, No. 2, ISSN: 2355-9365

[6] Sezer, S., Scott-Hayward, S., Chouhan, P. K., Fraser, B., Lake, D., Finnegan, J., Vilijoen, N., Miller, M., dan Rao, N. (2013). Are We Ready for SDN? Implementation Challenges for Software-Defined Networks. IEEE Communications Magazine, 51 (7), 36-43

[7] Azodolmolky S. 2013. Software Defined Networking with OpenFlow. Packt Publishing, ISBN 978-1-84969872-6

[8] Firmansyah M. B, Muldina R, dan Sanjayo D. D. Mengimplementasikan Sistem Keamanan Jaringan Intrusion Prevetion System Berbasis Snort Pada Arsitektur Software Define Network. Bandung: Universitas Telkom

[9] Putra Jan W, G. 2020. Pengenalan Konsep Pembelajaran Mesin dan Deep Learning. Tokyo, Jepang, edisi 1.4

[10] Tang Tuan A, Mhamdi L, McLemon Des, Zaidi S. A. R, dan Ghogho M. 2016. Deep Learning Approach for Network Intrusion Detection in Software Defined Networking. IEEE, 978-1-5090-3837-4/16

[11] Dhanabal L, Shantharajah S. P. 2015. A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms. International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 6, ISSN 2278-1021

[12] Aleroud A, Izzat Alsmadi. 2016. Identifying DoS Attacks on Software Defined Networks : A Relation Context Approach. IEEE/IFIP Network Operations and Management Symposium

[13] Wahba Y, ElSalamouny E and EITaweel. 2015. Improving the Performance of Multi-class Intrusion Detection Systems using Feature Reduction. IJCSI International Journal of Computer Science Issues, Volume 12, Issue 3, ISSN (Online): 1694-0784

[14] Han J, Micheline Kamber Jian Pei. 2012. Data Mining Concepts and Techniques. The Morgan Kaufmann, ISBN 978-0-12-381479-1