International Journal on Advanced Science Engineering Information Technology

Vol.12 (2022) No. 2
ISSN: 2088-5334

Feature Selection using Information Gain Method for Building Classification Model DDoS Attack at Application Layer

Muhammad Afrizal Amrustian [a,*], Heru Sukoco [b,*], Shelvie Nidya Neyman [b]

[a] *Department of Informatics, Institut Teknologi Telkom Purwokerto, Banyumas, Central Java, 53147, Indonesia*
[b] *Department of Computer Science, IPB University, Dramaga, Bogor, 16680, Indonesia*
*Corresponding author: *afrizal.amru@ittelkom-pwt.ac.id*

Abstract—**Distributed Denial of Services (DDoS) is one of the digital attacks that often occurred, the record for DDoS attacks in the second quartal of 2018 reaches 5.7Gbps. The application layer becomes one of the targets for this attack type; this type of DDoS attack always mimicks the user's request, making it harder to detect than DDoS attack at the network and transport layer. The classification has been offered as one method to overcome this problem. Before classification, the selection feature becomes important due to some features that lead to error classification and make the process classification longer. This research uses information gain as a selection feature method and using CICIDS 2017 as the dataset. The CICIDS2017 has 692.704 records consist of 78 features and five classes. The result of feature selection using the information gain method reduces the numbers of features from 78 to 5. To prove that these five features can classify DDoS attacks correctly, we use a randomForest method as a classification method. The randomForest was used to classify the data into five classes: normal, DDoS Goldeneye, DDoS Hulk, DDoS Slowhttptest, and DDoS Slowloris. The result of performance for accuracy is 99.43%, for recall of each class are 99.48%, 99.81%, 99.41%, 96.01%, 99.97% respectively. Besides the result of performance for precision each class are 99.65%, 96.04%, 99.90%, 98.63%, 71.37%, respectively. The results of performance for classification time using five features are decreasing execution time 3.1 seconds.**

Keywords—**Application layer; classification; DDoS; feature selection; information gain.**

Manuscript received 20 Dec. 2019; revised 28 Dec. 2020; accepted 5 Feb. 2021. Date of publication 30 Apr. 2022.

I. INTRODUCTION

Distributed Denial of Services (DDoS) is a kind of attack that the internet faces. DDoS attacks the server to slow down the server's performance, even though they intend to stop the server from handling real users' requests [1]. The sectors that become the victim of DDoS attacks always incur losses, such as the banking [2], flight [3], and entertainment sector [4]. Nowadays, applications for launching DDoS attacks are easy to find [5], and it is not tough to operate. The report has proved this problem that in the second quartile of 2018, DDoS attack's peak reaches 5.7Gbps [6]. This large amount of DDoS attacks is targeting the transport, network, and application layer.

The DDoS attack in the application layer differs from the DDoS attack in the network and transport layer. The last type of DDoS attack focused on flooding bandwidth that affects users who cannot access the server. The DDoS attack in the application layer focused on sending a request to the server continuously. The purpose of DDoS attacks has been to make the server busy, and the server cannot respond to the request from a real user. The request sent by the attacker mimics the request sent by a real user, so it makes more effort to distinguish the real and the attack [7].

On the other hand, there are two general problems in DDoS attack at the application's layer: large amounts of data in network traffic and difficulty distinguishing the real request and DDoS attack in application's layer. Some researchers used classification as one of the solutions to overcome the problems [8], [9]. However, the classification process of DDoS attacks depends on the features used in the process. Features used to classify DDoS attacks must be selected carefully because some features often make misclassification and reduce classification performance [10]. Therefore, feature selection at DDoS attack is the critical phase before doing the classification.

Osonaiye overcome the selection of DDoS attacks problems using features selection. Different method was employed using the information gain method to build the ensemble-based multi-filter feature selection (EMMFFS) method. That method selects 13 features from 41 features

DDoS attacks [11]. Kumar *et al.* used the information gain method to select 20 features from 41 features for detecting an attack in IDS; one of the attack types is DDoS. They compare the result from a classification that used all features and 12 features by j48 classifier. Classification with 12 features has 61% accuracy and not significantly different when used all features [12]. However, both existing studies used the NSL-KDD dataset, which was not focused on the application layer DDoS attack.

In the application layer DDoS attack research, some researchers proposed a different way to deal with the feature selection problems. Wang *et al.* proposed a multilayer perceptron to conquer the problem in selecting DDoS attack features. The proposed method by wang *et al.* could yield the feature is used to detect DDoS attack [13]. Agrawal and Rajput proposed the randomForest method as a classifier to classify the type of DDoS attack. The result showed that the randomForest method has the highest accuracy and precision compared with other methods, such as Naive Bayes, OneR, and Multilayer Perceptron [14]. Besides, randomForest is also utilized for detecting DDoS attacks in intrusion detection systems (IDS). This method detected DDoS attacks with an accuracy of the classification performance is 99.84% [15]. Hakim *et al.* has improved the three-sigma value into a six-sigma value to increase DDoS attack's detection rate [16]. However, this method is explicitly employed for SDN-based networks. Ahmed *et al.* tried to overcome DoS attack detection or classification problem by utilizing routers in a network environment. They classified DoS attacks based on neighbor router information, and they called it Secure Neighbor Discovery(SeND) [17]. The idea is good, but deploying SeND is not easy, and SeND has not proven to classify any DoS attack.

This research focuses on the feature selection of DDoS attack in the application layer based on the previous research. Furthermore, information gain was used to select features to reduce misclassification and increase the classification performance. Moreover, the randomForest was used as a classifier to identify the type of attack in DDoS attack and build the accuracy and easy training [18].

II. MATERIAL AND METHOD

This section describes the proposed classification model of DDoS attacks at the application layer. This model is built using the information gain method to select the application layer's DDoS attack feature. The conceptual model is shown in Fig. 1. Fig 1 shows the process in this research. The first is preparing the dataset, and the used dataset is described in another subsection. The next step is data cleaning and feature selection. Data splitting is the procedure after the critical feature is obtained. Model classification is built in the classification step, and the last is analysis and evaluation of the model.

A. Dataset

The data used in this research is the CICIDS2017 dataset from the University of New Brunswick (UNB). UNB is a research center in Canada that focuses on cybersecurity. Application layer DDoS attack is one of the topics that UNB has provided the data on its website. CICIDS2017 dataset consists of regular traffic and application layer DDoS attacks such as slowloris, slowhttptest, hulk, and goldeneye. CICIDIDS2017 contains 692.704 records with 78 features and one label.

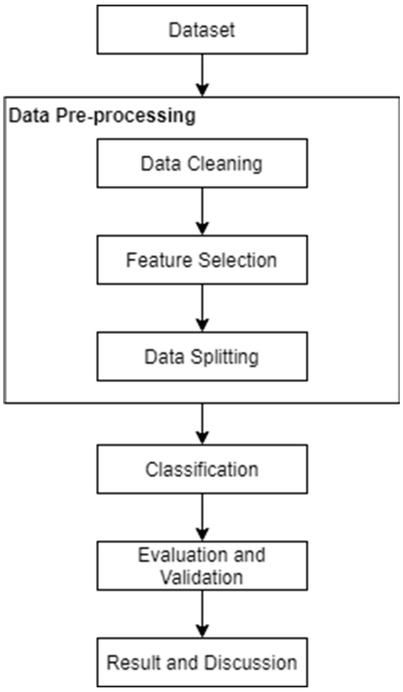


Fig. 1 The Conceptual Model

B. Data Cleaning

Data pre-processing is applied to improve the accuracy for classification performance on the dataset. All collected data need to be pre-processed before the classification process starting. The first step of pre-processing data in this research is data cleaning. There is a nan record in the CICIDS2017 dataset, or the other name is a missing value. The summary of missing values in the CICIDS2017 dataset is described in Table 1.

TABLE I
SUMMARY MISSING VALUE

Class	Feature	Records	
		Data	**%**
DDoS	Flow bytes per second	949	0.14
Hulk	Flow packets per second	348	0.05
Normal	Flow bytes per second	949	0.14
	Flow packets per second	348	0.05

Missing value records in CICIDS 2017 dataset is up to 1 percent. Therefore, deleting the missing value is possible because the deleting missing value data does not affect the entirety data [19].

C. Feature Selection

This research used information gain as a method to select features that have a strong correlation with the application layer DDoS attack to identify this type of attack. Information gain has been employed to find correlated features and reduce the features [20]. Information gain worked with reducing uncertainty value Y with uncertainty value Y has given the observation of value X. The uncertainty value Y can be calculated by its entropy, as shown in Equation 1. The uncertainty value Y given observation value X is the value from calculating the entropy of value Y given the observation of value X, as indicated in Equation 2. From the explanation above, we can conclude that the bigger the value information gain of attribute X, the more significant the correlation between attribute X and class Y [20].

$$H(Y) = -\sum_i P(y_i)\, log_2(P(y_i)) \quad (1)$$

$$H(Y|X) = -\sum_j P(x_j) \sum_i P(y_i|x_j)\, log_2\left(P(y_i|x_j)\right) \quad (2)$$

$$IG(Y|X) = H(Y) - H(Y|X) \quad (3)$$

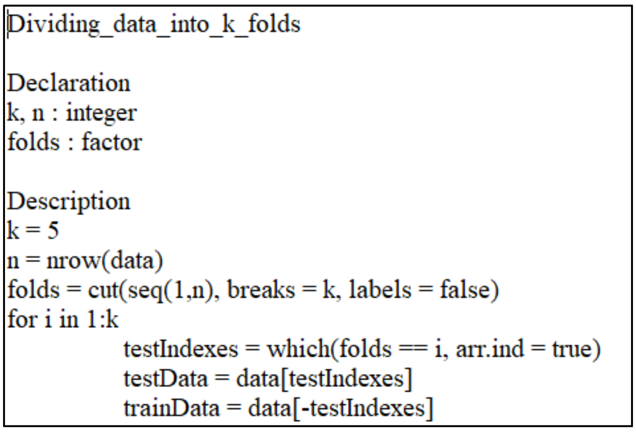
$P(y_i)$ = Prior probability from Y

$P(y_i|x_i)$ = Posterior probability Y given by X

D. Data Splitting

The next step of pre-processing data is splitting data. We use the K-fold Cross-validation method for splitting data [21]. The data dividing into five folds consist of one-fold as test data and four folds as train data. How to split data using k-fold cross-validation in r can be seen in Fig 2.

We have variables 'k' and 'n' in integer type. Variable 'folds' is a factor type. Variable 'k' is the number of folds, and 'n' is the amount of data. We gained the number of data from calculating the number of rows in data. Variable 'folds' is a factor type variable that consists of a number. We use cut to dividing range from the seq function into some interval. The interval is the total fold that we created before. TestIndexes has a function to take index from folds that we created before. TestIndexes created the index for taking data from the dataset. We use this one-fold data as a data test and the rest of the fold as train data. We created testData variables from data that have an index as testIndexes and the rest of the data as data train.



```
Dividing_data_into_k_folds

Declaration
k, n : integer
folds : factor

Description
k = 5
n = nrow(data)
folds = cut(seq(1,n), breaks = k, labels = false)
for i in 1:k
        testIndexes = which(folds == i, arr.ind = true)
        testData = data[testIndexes]
        trainData = data[-testIndexes]
```

Fig. 2 Pseudocode for splitting data

E. Classification

The decision tree is a method to support decision-making. The method is a tree-like model of decisions. The decision tree consists of three elements: node, condition, and production (p-value) [22]. Fig 3 illustrates the architecture decision tree method. In Fig 3 A or black border oval shape is the root node, B or the red border oval shape is an internal node, C or the green border oval shape is a terminal node. The blue rectangle is the condition. The grey box below is the p-value of each class and the number of cells.

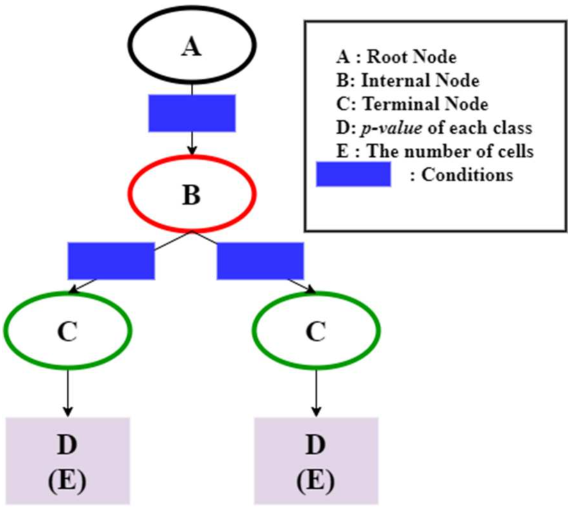


Fig. 3 Decision tree architecture [14]

The randomForest is a classification method that combines some of the tree classifications. Breiman developed a randomForest method. The randomForest consists of some tree model classifications {h(x,Θ$_k$ Θ$_k$), k=1, …} Where {Θ$_k$ Θ$_k$} is a vector that is chosen randomly and independently. Each tree showed their result, then randomForest vote, and chose the result most [23].

The last result from randomForest classification is the result of voting each tree classification. Fig 4 explains how randomForests work. Tree 1, Tree 2, …, tree b is trees in the randomForest method. k_1k_1, k_2k_2,…, k_bk_b are the result of tree classification. The randomForest voted the most class from the tree and made that class results from randomForest classification [24].

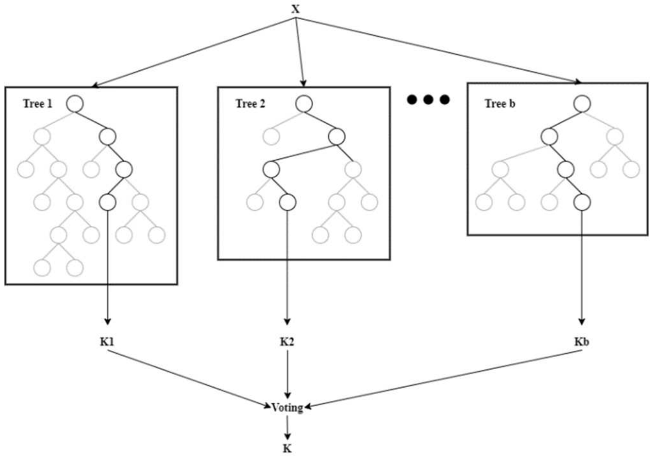
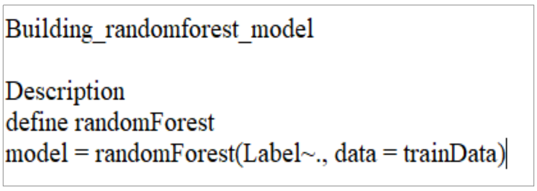


Fig. 4 The randomFforest's architecture [16].

The randomForest counts the first learner or result of classification for each tree, then the total from each class was revealed. The randomForest was used to choose the class that has a bigger total than the last result as indicated in the equation for the voting class in the randomForest in Equation 4 below.

$$f(x) = arg_{y\epsilon Y}^{max} \sum_{j=1}^{J} I(y = h_j(x)) \quad (4)$$

F(x) is the result of randomForest classification, and h$_j$(x) is the result of each tree classification [25]. *I* is the indicator function that returns 1 if the tree classification results are the same with the *y* class. The function returns 0 if the tree classification result does not match with the *y* class. The randomForest classification is utilized by the randomForest package in R that shown in Fig 5.



```
Building_randomforest_model

Description
define randomForest
model = randomForest(Label~., data = trainData)
```

Fig. 5 Pseudocode for making classification model.

First, we call randomForestpackages using library function. Then we make the variable model as a model to classify. Last, we use randomForest function to make a model. We use the label as a reference and trainData as a data train.

The randomForest has been chosen to be a classifier and build the model because of its superiority over other traditional machine learning methods in terms of accuracy and easy training [18]. The model can measure the accuracy, precision, recall, and process time; with this measurement can be concluded that using selected features has no different result than using all features and describing it in the next section.

III. Results and Discussion

The information gain method produced the score. Suppose the IG score is higher, the stronger the correlation between feature and class. It means that the correlation showed the best result. After calculating the information gain score and ranking the score, we found five top features with a higher score. The result of feature selection is shown in Table 2.

TABLE II
After Feature Selection

Feature	IG Score
Init win bytes forward	0.6094
Flow inter-arrival time max	0.5478
Max packet length	0.5403
Average backward segment size	0.5380
Backward packet length means	0.5380

A. Comparison of Previous Feature Selection

In this section, we compare our work with the previous research about DDoS attack feature selection. The comparison of the previous research is shown in Table 3. Table 3 showed that this study could reduce the feature and select the significant feature compared to the other existing research. Our research can reduce features of the application layer of DDoS attacks to only five features. These features can reduce misclassification and enhance classification performance.

TABLE III
Summary of the Comparison Result

Researcher	Method	Feature Selected
Osonaiye *et al* [11]	Ensemble-based multi-filer feature selection	13 Features
Kumar *et al* [12]	OneR+Relief	12 Features
Author	Proposed Method	5 Features

B. Evaluation and Validation

The evaluation process is used to evaluate one or more methods to improve quality and effectiveness. In this research, the previous studies performance and comparison used feature selection, including J48 and Random Forest, for different attack type problems in the application layer. We use the confusion matrix in the analysis and evaluation step. From the confusion matrix, we can know the accuracy model, recall, and precision. Before comparing selected and all features, we compare our result with previous research. Table 4 shows the compared result between them.

TABLE IV
Compared Accuracy

Attack Type	Method	Accuracy
DoS	J48	99.25%
DoS	RandomForest	99.97%
DDoS	RandomForest	99.43%

Table 4 shows us that the RandomForest method has a higher accuracy than J48 classifier with 99.97% for DoS attack and 99.43% for DDoS attack. The RandomForest has been successfully classified for both DoS and DDoS attacks. Hence, the RandomForest has successfully identified the type of DDoS attack in the application layer. Table 5 shows compared accuracy model between five features and all features that is used in this research.

TABLE V
Accuracy

Folds	Accuracy	
	5 Features	78 Features
1	99.42%	99.92%
2	99.45%	99.91%
3	99.41%	99.93%
4	99.47%	99.93%
5	99.43%	99.93%
Average	99.43%	99.92%

Accuracy between using all features and five features showing have not significantly different. Two of them reach 99% accuracy. However, the accuracy that reaches the same result has a different execution time as noted in the execution time in Table 6. In Table 6, the results of the execution time of five features reach 4.4 seconds to classify the DDoS attack in the application layer. Compared with using all features that have execution time is 7.5 seconds. There is a time difference for both compared results, 3.1 seconds. The execution time difference proves that feature selection can reduce the execution time for classifying the DDoS in the application layer.

TABLE VI
Execution Time

Execution time (second)	
5 Features	4.4
78 Features	7.5

Table 7 gives us information about the category range for the result of precision and recall. There are three categories for precision and recall they are low, standard, and high [26]. The score for each category can be seen in Table 7 below.

TABLE VII
CATEGORIES FOR PRECISION AND RECALL SCORE

Low (%)	Standard (%)	High (%)
0-33	34-66	67-100

After we know the category for precision and recall scores, we calculate all precision for each class. The precision calculation is needed to know the proportion of the correct classification in each class. Table 8 shows the precision from normal class and reached 99% for each feature used. Both five and seventy-eight features reached a high category and using five features has the advantage than seventy-eight features in execution time. Execution time using five features is faster 3.1 seconds than using seventy-eight features.

TABLE VIII
NORMAL CLASS PRECISION SCORE

Folds	Precision normal	
	5 Features	78 Features
1	99.61%	99.91%
2	99.66%	99.90%
3	99.64%	99.92%
4	99.67%	99.92%
5	99.67%	99.92%
Average	99.65%	99.91%

DDoS goldeneye's precision score in Table 9 shows the difference between using five features and seventy-eight features in table 9 is 3.4%. The differences are proximate. Using five features still including in high category because it reaches above 90% score. Even the score smaller, using five features does the classification faster.

TABLE IX
DDOS GOLDENEYE PRECISION SCORE

Folds	Precision goldeneye	
	5 Features	78 Features
1	96.28%	99.75%
2	96.44%	99.80%
3	95.76%	99.80%
4	96.11%	99.80%
5	95.64%	99.80%
Average	96.04%	99.79%

The precision score from the DDoS hulk class is shown in Table 10.

TABLE X
DDOS HULK PRECISION SCORE

Folds	Precision hulk	
	5 Features	78 Features
1	99.91%	99.98%
2	99.89%	99.98%
3	99.90%	99.98%
4	99.89%	99.98%
5	99.90%	99.98%
Average	99.90%	99.98%

Precision using five and seventy-eight features reaches the same 99% score, and both go into the high category. Table 11 shows the precision score from DDoS slowhttptest class.

TABLE XI
DDOS SLOWHTTPTEST PRECISION SCORE

Folds	Precision slowhttptest	
	5 Features	78 Features
1	98.78%	99.25%
2	98.98%	99.25%
3	98.53%	99.25%
4	99.11%	99.25%
5	97.75%	99.25%
Average	98.63%	99.25%

Precision score from DDoS slowhttptest class using five features reach 98%. Meanwhile, using seventy-eight features reach a 99% precision score. Even using five features has a smaller score, but still belongs to a high category and the execution time faster 3.1 seconds. Table 12 shows the precision score from the last class, DDoS slowloris.

TABLE XII
DDOS SLOWLORIS PRECISION SCORE

Folds	Precision slowloris	
	5 Features	78 Features
1	71.37%	99.39%
2	71.80%	99.39%
3	69.73%	99.39%
4	72.08%	99.39%
5	71.84%	99.39%
Average	71.37%	99.39%

The precision score for using five features at slowloris class reached 70.8%. There is a big difference between using all features that reach a 99% precision score. Even there is a significant difference between them, the precision score using five features still belongs to the high category.

Based on these categories, precision scores from using five features for all classes belong to a high category. That is means that the proportion of the right classification for each class is high. We see that the difference between using five features and all features not much, but execution time from using five features shorter than using all features.

The recall is one of a method for analyzing classification result for each class. Recall function is proportion calculation to know how much the real class from data has been classified correctly. Table 13 shows the recall score from a normal class.

TABLE XIII
NORMAL RECALL SCORE

Folds	Recall normal	
	5 Features	78 Features
1	99.48%	99.98%
2	99.50%	99.98%
3	99.44%	99.98%
4	99.50%	99.98%
5	99.46%	99.98%
Average	99.48%	99.98%

Recall scores in normal class have no difference between using five features and all features; both reached a 99% score and belong to a high category. Even in the same category, execution time using five features better than seventy-eight features by 3.1 seconds. Table 14 below shows the recall score from the goldeneye class.

TABLE XIV
DDOS GOLDENEYE RECALL SCORE

Folds	Recall goldeneye	
	5 Features	78 Features
1	99.95%	99.46%
2	99.79%	99.46%
3	99.74%	99.46%
4	99.79%	99.46%
5	99.80%	99.46%
Average	99.81%	99.46%

The recall score for both features reaches the same score, which is 99%. It is no difference between them, and both of them belong to a high category. Only execution time distinguishes between them, using five features faster in execution time than using seventy-eight features. Table 15 below shows the recall score from the DDoS hulk class.

TABLE XV
DDOS HULK RECALL SCORE

Folds	Recall hulk	
	5 Features	78 Features
1	99.35%	99.83%
2	99.39%	99.82%
3	99.39%	99.86%
4	99.47%	99.86%
5	99.43%	99.86%
Average	99.41%	99.85%

Recall score for DDoS hulk class reach a 99% score for both five and all features. Recall score using five features belong to a high category, and the execution time faster 3.1 seconds faster. Recall score for DDoS slowhttptest can we see in Table 16.

TABLE XVI
DDOS SLOWHTTPTEST RECALL SCORE

Folds	Recall slowhttptest	
	5 Features	78 Features
1	95.75%	99.71%
2	96.76%	99.71%
3	96.49%	99.71%
4	95.65%	99.71%
5	95.85%	99.71%
Average	96.10%	99.71%

Recall scores using five features reach 95.4%, this scores smaller 3.6% than all features score but still belong to the high category. The recall score from the last class DDoS slowloris is shown in Table 17.

TABLE XVII
DDOS SLOWLORIS RECALL SCORE

Folds	Recall slowloris	
	5 Features	78 Features
1	99.88%	99.73%
2	100%	99.73%
3	100%	99.73%
4	100%	99.82%
5	100%	99.73%
Average	99.97%	99.75%

Recall scores from DDoS slowloris class at second till fifth fold reach 100%. These scores mean at that fold all real class has been classified correctly. The average recall score for DDoS slowloris class reaches 99% for both features, and both belong to the high category. All recall scores from each class belong to the high category for both feature usage. That means the real class has classified highly correctly. Even in the same category, using five features still has the advantage of execution time.

IV. CONCLUSION

The DDoS attacks have increased recently because of many tools and easy to use, researching DDoS attacks continuously. To enhance the classification results, all features that represent DDoS attacks must be selected. This research makes feature selection from 78 features at CICIDS 2017 dataset. Using the information gain method, we calculated the information gain score to know the correlation between feature and class. Therefore, we selected five features from 78 features. They are United Win Bytes Forward, Flow Inter-Arrival Time Max, Max Packet Length, Average Backward Segment Size, and Backward Packet Length Mean. Five features that we selected were used to build a model with the randomForest method.

We obtained the accuracy of the model that we build from the five features has reached 99.43%. The same result that we obtained if we used all the features in making the classification. This accuracy had a higher result than the J48 classifier. Recall and precision were also measured besides the accuracy to evaluate the performance from the proposed model. Recall results from the model using five features for each class, such as normal class 99.48%, DDoS goldeneye class 99.81%, DDoS hulk class 99.41%, DDoS slowhttptest class 96.10%, and DDoS slowloris class 99.97%.

Thus, the real class in the data train has been successfully classified correctly with the model we build from five features. Precision result for each class which is including normal class 99.65%, DDoS goldeneye class 96.04%, DDoS hulk class 99.90%, DDoS slowhttptest class 98.63%, and DDoS slowloris class 71.37%. From the precision result, we can conclude that results from model classification reach outstanding performance with an average above 90% classification result shown the classifier has been classified correctly. Based on this evaluation and validation, these five features have been successfully reduced the execution time of the classification process and help enhance the classification performance of the randomForest that has successfully classified DDoS attacks in the application layer.

REFERENCES

[1] C. Douligeris and D. N. Serpanos, Network security Current Status and Future Direcetions. 2007.

[2] J. Bradshaw, “HSBC online banking crashes after cyber attack,” The Telegraph web, 2016. [Online]. Available: https://www.telegraph.co.uk/finance/newsbysector/banksandfinance/12129411/HSBC-online-banking-service-crashes-again.html.

[3] A. Kharpal, “Hack attack leaves 1,400 airline passengers grounded,” CNBC Web, 2015. [Online]. Available: https://www.cnbc.com/2015/06/22/hack-attack-leaves-1400-passengers-of-polish-airline-lot-grounded.html.

[4] O. Kupreev, E. Badovskaya, and A. Gutnikov, "DDoS attacks in Q3 2018," Securelist, 2018. [Online]. Available: https://securelist.com/ddos-report-in-q3-2018/88617/.

[5] B. Nagpal, P. Sharma, N. Chauhan, and A. Panesar, "DDoS tools: Classification, analysis and comparison," 2015 Int. Conf. Comput. Sustain. Glob. Dev. INDIACom 2015, pp. 342–346, 2015.

[6] Verisign, "Verisign Distributed Denial of Service Report," 2018.

[7] S. Ranjan, R. Swaminathan, M. Uysal, A. Nucci, and E. Knightly, "DDoS-shield: DDoS-resilient scheduling to counter application layer attacks," IEEE/ACM Trans. Netw., vol. 17, no. 1, pp. 26–39, 2009.

[8] K. J. Singh and T. De, "MLP-GA based algorithm to detect application layer DDoS attack," J. Inf. Secur. Appl., vol. 36, pp. 145–153, 2017.

[9] I. Ko, D. Chambers, and E. Barrett, "Self-supervised network traffic management for DDoS mitigation within the ISP domain," Futur. Gener. Comput. Syst., vol. 112, pp. 524–533, 2020.

[10] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, "Feature selection and classification in multiple class datasets: An application to KDD Cup 99 dataset," Expert Syst. Appl., vol. 38, no. 5, pp. 5947–5957, 2011.

[11] O. Osanaiye, H. Cai, K. K. R. Choo, A. Dehghantanha, Z. Xu, and M. Dlodlo, "Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing," Eurasip J. Wirel. Commun. Netw., vol. 2016, no. 1, 2016.

[12] K. Kumar, G. Kumar, and Y. Kumar, "Feature Selection Approach for Intrusion Detection System," Int. J. Adv. Trends Comput. Sci. Eng., vol. 2, no. 5, pp. 47–53, 2013.

[13] M. Wang, Y. Lu, and J. Qin, "A dynamic MLP-based DDoS attack detection method using feature selection and feedback," Comput. Secur., vol. 88, 2020.

[14] S. Agrawal and R. Singh Rajput, "Denial of Services Attack Detection using Random Forest Classifier with Information Gain," Int. J. Eng. Dev. Res., vol. 5, no. 3, pp. 929–938, 2017.

[15] N. Farnaaz and M. A. Jabbar, "Random Forest Modeling for Network Intrusion Detection System," Procedia Comput. Sci., vol. 89, pp. 213–217, 2016.

[16] A. K. Hakim, M. Abdurohman, and F. A. Yulianto, "Improving DDoS detection accuracy using Six-Sigma in SDN environment," Int. J. Adv. Sci. Eng. Inf. Technol., vol. 8, no. 2, pp. 365–370, 2018.

[17] A. S. Ahmed, R. Hassan, and N. E. Othman, "Denial of service attack over secure neighbor discovery (SeND)," Int. J. Adv. Sci. Eng. Inf. Technol., vol. 8, no. 5, pp. 1897–1904, 2018.

[18] X. K. Li, W. Chen, Q. Zhang, and L. Wu, "Building Auto-Encoder Intrusion Detection System based on random forest feature selection," Comput. Secur., vol. 95, p. 101851, 2020.

[19] J. Fox and A. Leanage, "R and the Journal of Statistical Software," J. Stat. Softw., vol. 73, no. 2, 2016.

[20] W. Wang and S. Gombault, "Efficient detection of DDoS attacks with important attributes," Proc. 2008 3rd Int. Conf. Risks Secur. Internet Syst. Cris. 2008, pp. 61–67, 2008.

[21] T. Shorey, D. Subbaiah, A. Goyal, A. Sakxena, and A. K. Mishra, "Performance Comparison and Analysis of Slowloris, GoldenEye and Xerxes DDoS Attack Tools," 2018 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2018, pp. 318–322, 2018.

[22] I. Park and S. Lee, "Spatial prediction of landslide susceptibility using a decision tree approach: a case study of the Pyeongchang area, Korea," Int. J. Remote Sens., vol. 35, no. 16, pp. 6089–6112, 2014.

[23] L. Breiman, "Random forests," Mach. Learn., vol. 45, pp. 5–32, 2001.

[24] A. Verikas, A. Gelzinis, and M. Bacauskiene, "Mining data with random forests: A survey and results of new tests," Pattern Recognit., vol. 44, no. 2, pp. 330–349, 2011.

[25] C. Zhang and Y. Ma, Ensemble machine learning: Methods and applications. 2012.

[26] N. P. Lestari, "Uji Recall and Precision Sistem Temu Kembali," Libr. Net, vol. 5, no. 3, pp. 45–46, 2016.