Distributed Denial of Service (DDOS) Attack Detection on Zigbee Protocol Using Naive Bayes Algorithm

Ibnu Mas'ud[a,1,], Kusrini Kusrini[a,2], Agung Budi Prasetio[a,3]

[a] *Universitas Amikom Yogyakarta, Address, Sleman 55281, Indonesia*
[1] *ibnu.1198@students.amikom.ac.id;* [2] *kusrini@amikom.ac.id;* [3]*agungbp@amikom.ac.id*

ARTICLE INFO

Article history:
Received: 08 May 2021
Revised: 25 July 2021
Accepted: 12 Oct 2021

Keywords:
IDS
DDoS
ZigBee
Naïve Bayes
Classification

ABSTRACT

Distributed Denial of Service or better known as DDoS is an attempted attack from several computer systems that target a server so that the amount of traffic becomes too high so that the server cannot handle the request. DDoS is usually done by using several computer systems that are used as sources of attacks. So, they attack one server through several computers so that the amount of traffic can also be higher. A DDoS attack is like a traffic jam that prevents a driver from reaching their desired destination on time. According to data, 33% of businesses in the world have fallen victim to DDoS attacks. DDoS is hard to trace. Some types of DDoS attacks can be very powerful and even reach speeds of 1.35 Tbsp. Additionally, DDoS attacks can cause losses of $ 40,000 per hour if they occur. ZigBee is a standard from IEEE 802.15.4 for data communication on personal consumer devices as well as for business scale. ZigBee is designed with low power consumption and works for low level personal networks. ZigBee devices are commonly used to control another device or as a wireless sensor. ZigBee has a feature which is able to manage its own network or manage data exchange on the network. Another advantage of ZigBee is that it requires low power, so it can be used as a wireless control device which only needs to be installed once, because only one battery can make ZigBee last up to a year. In addition, ZigBee also has a "mesh" network topology so that it can form a wider network and more reliable data. In the previous research of Muhammad Aziz, Rusydi Umar, Faizin Ridho based on the results of the analysis carried out that the attack information that has been detected by the IDS based on signatures needs to be reviewed for accuracy using classification with statistical calculations. Based on the analysis and testing carried out with the artificial neural network method, it was found that the accuracy was 95.2381%. The neural network method can be applied in the field of network forensics in determining accurate results and helping to strengthen evidence at trial. The Naïve Bayes model performed relatively poor overall and produced the lowest accuracy score of this study (45%) when trained with the CICDDoS2019 dataset. For the same model, precision was 66% and recall was 54%, meaning that almost half the time, the model misses to identify threats.

I. Introduction

Distributed Denial of Service or better known as DDoS is an attempted attack from several computer systems that target a server so that the amount of traffic becomes too high so that the server cannot handle the request. DDoS is usually done by using several computer systems that are used as sources of attacks. So, they attack one server through several computers so that the amount of traffic can also be higher. A DDoS attack is like a traffic jam that prevents a driver from reaching their desired destination on time. According to data, 33% of businesses in the world have fallen victim to DDoS attacks. DDoS is hard to trace. Some types of DDoS attacks can be very powerful and even reach speeds of 1.35 Tbsp. Additionally, DDoS attacks can cause losses of $ 40,000 per hour if they occur.

ZigBee is a standard from IEEE 802.15.4 for data communication on personal consumer devices as well as for business scale. ZigBee is designed with low power consumption and works for low level personal networks. ZigBee devices are commonly used to control another device or as a wireless sensor. ZigBee has a feature which is able to manage its own network or manage data exchange on the network [1]. Another advantage of ZigBee is that it requires low power, so it can be used as a wireless control device which only needs to be installed once, because only one battery can make ZigBee last up to a year. In addition, ZigBee also has a "mesh" network topology so that it can form a wider network and more reliable data.

In the previous research of Muhammad Aziz, Rusydi Umar, Faizin Ridho based on the results of the analysis carried out that the attack information that has been detected by the IDS based on signatures needs to be reviewed for accuracy using classification with statistical calculations. Based on the analysis and testing carried out with the artificial neural network method, it was found that the accuracy was 95.2381%. The neural network method can be applied in the field of network forensics in determining accurate results and helping to strengthen evidence at trial.

In previous research, Jodi Chris Jordan Sihombing, Dany Primanita Kartikasari, Adhitya Bhawiyuga based on the tests that have been carried out, SDMD's performance in detecting DDoS attacks is very good. The accuracy obtained in detecting DDoS attacks is 96.08%, 95.66%, and 98.76% for syn flooding, udp flooding, icmp flooding, respectively. The system can also cope with and minimize the impact of DDoS attacks. This can be proven from the number of attack packets that enter the victim host decreasing when SDMD is activated.

In previous research Nadila Sugianti, Yayang Galuh, Salma Fatia, Khadijah Fahmi Hayati Holle (2020) based on the discussion that has been explained and the results of tests that have been carried out regarding the problem of detecting HTTP-based DDOS attacks based on the number of users, number of packages, number of packages / user and length. The data captured, the fuzzy logic method using Sugeno method can be used as a detector in determining HTTP-based DDOS attacks with an accuracy of up to 90%.

In previous research, Kurniabudi, Abdul Harris, Abdul Rahim, (2020) based on experimental data that the Information Gain feature selection technique was able to improve the performance of the classification method, especially Random Forest which has better performance than Naïve Bayes, Bayes Network, OneR, AdaBoost and Random A tree with 99.99% accuracy in testing all training data and 99.95% on testing using 10-fold cross validation. But on the other hand, Random Forest has a longer time to build models and training processes when compared to Naïve Bayes, Bayes Network, OneR, AdaBoost and Random Tree. In the experiments conducted in this study, researchers used Information Gain as a feature selection technique for the CICIDS-2017 dataset in detecting DDoS attacks. For further research, other feature selection techniques can be used that might improve DDoS attack detection performance. Apart from the use of other classification techniques need to be considered in the next research, especially those which have better performance with lower computation time.

In previous research, Arif Wirawan Muhaammad, Cik Feresa Mohd Foozy, Ahmad Azhari (2020) based on experimental results that the combination of the seven key data set features selected used as input for the classification of artificial neural networks in this study gave the highest accuracy value of 97.76%.

Lila Dini Utami, Romi Satria Wahono (2015) that Naïve Bayes is a classifier that classifies a text, one example is restaurant reviews. Naïve Bayes is very simple and efficient, is also very popular for text classification and performs well on many domains. There are 3 stages of data processing, namely naïve bayes, naïve bayes and information gain, and naïve bayes, information gain, and adaboost. And it turns out, if only naïve bayes are used, the accuracy will only reach 70% and AUC = 0.500. Likewise, if naïve Bayes are accompanied by information gain, the accuracy achieved is only 70% and AUC = 0.500, it proves that the information gain does not affect the accuracy of naïve Bayes. However, if naïve bayes and information gain are accompanied by adaboost, the accuracy increases 29.5% to 99.5% and AUC = 0.995.

Al Riza Khadafy, Romi Satria Wahono based on the results of experiments and evaluations in this study, in general it can be concluded that the application of the NB classification algorithm can reduce noise data on large datasets and have many classes or multi-classes so that the classification accuracy of the DT algorithm can be increased. The accuracy results obtained indicate that the proposed method DT + NB is superior to the DT method, with an accuracy value for each test dataset such as Breast Cancer 96.59% (21.06% increase), Diabetes 92.32% (increase 18 , 49%), Glass 87.50% (increase 20.68%), Iris 97.22% (increase 1.22%), Soybean 95.28% (increase 3.77%), Vote 98.98% (increased 2.66%), Image Segmentation 99.10% (increased 3.36%), and Tic-tac-toe 93.85% (increased 9.30%). Comparison of accuracy values is carried out by t-test or t-test between the DT method and the proposed method of DT + NB to obtain a significant difference in accuracy between the two methods. From the comparison results, the P value (T <= t) is 0.01321, this indicates that the p value is smaller than the alpha value (0.01321 <0.05).

The Naive Bayes algorithm is a classification method using probability and statistical methods proposed by the English scientist Thomas Bayes. The Naive Bayes Algorithm predicts future opportunities based on past experiences, so it is known as Bayes' Theorem. The main characteristic of this Naïve Bayes Classifier is a very strong assumption (naive) of the independence of each condition / event. Naive Bayes Classifier performs very well compared to other classifier models, "Naïve Bayes Classifier has a better level of accuracy than other classifier models. The advantage of using this method is that this method only requires a small amount of training data to determine the estimated parameters required in the classification process. Because it is assumed to be an independent variable, only the variance of a variable in a class is needed to determine the classification, not the entire covariance matrix.

Many studies have been done before and the Naive Bayes algorithm is the best model compared to other models such as: logistic regression, neural network, random forest, decission tree, support vector machine and k-nearest neigbor. Naive Bayes is a classification algorithm that is simple and easy to implement so that this algorithm is very effective when tested with the right dataset, especially if it is naïve bayes with feature selection, then naive bayes can reduce redudants in data (Witten, Frank, & Hall, 2011). The Naive Bayes algorithm is included in supervised learning and one of the fastest learning algorithms that can handle a number of features or classes (Lee, 2015).

In this study, the Instruction Detection System in the ZigBee Protocol will be implemented using the Naïve Bayes algorithm. The Naïve Bayes algorithm is a machine learning method that uses probability calculations. This algorithm makes use of probability and statistical methods to predict future probabilities based on past experiences.

II. Detection Approach

The DDoS attack detection approach implemented in this study is divided into several stages namely:

Retrieving Dataset

CICDDoS2019 contains benign and the recent DDoS attacks, resembling real data (PCAPs). It also includes the analysis of network traffic analysis using CICFlowMeter-V32 [51] and labelled flows. B-Profile system [47] was used to profile the abstract behaviour of human interactions and generate naturalistic benign background traffic. For this dataset, the abstract behaviour of 25 users was constructed based on the HTTP, HTTPS, FTP, SSH, and email protocols [47]. The dataset includes different modern reflective DDoS attacks such as Port Map, NetBIOS, LDAP, MSSQL, UDP, UDP-Lag, SYN, NTP, DNS, and SNMP. The capturing period for the training day on January 12th started at 10:30 and ended at 17:15, and for the testing day on March 11th started at 09:40 and ended at 17:35. Attacks were subsequently executed during this period.

Table 1 OS Specification and Machine IPs for CICDDoS2019. Adapted from DDoS Evaluation Set [47].

Machine	OS	IPs
Server	Ubuntu 16.04 (Web Server)	192.168.50.1 (first day) 192.168.50.4 (second day)
Firewall	Fortinet	205.174.165.81
PCs (first day)	Win 7	192.168.50.8
	Win Vista	192.168.50.5
	Win 8.1	192.168.50.6
	Win 10	192.168.50.7
PCs (second day)	Win 7	192.168.50.9
	Win Vista	192.168.50.6
	Win 8.1	192.168.50.7
	Win 10	192.168.50.8

A. Spesifics for CISCDDoS2019

Table 2 Time of Attacks for CICDDoS2019 dataset [49].

Days	Attacks	Attack Time
First Day	PortMap	9:43 - 9:51
	NetBIOS	10:00 - 10:09
	LDAP	10:21 - 10:30
	MSSQL	10:33 - 10:42
	UDP	10:53 - 11:03
	UDP-Lag	11:14 - 11:24
	SYN	11:28 - 17:35
Second Day	NTP	10:35 - 10:45
	DNS	10:52 - 11:05
	LDAP	11:22 - 11:32
	MSSQL	11:36 - 11:45
	NetBIOS	11:50 - 12:00
	SNMP	12:12 - 12:23
	SSDP	12:27 - 12:37
	UDP	12:45 - 13:09
	UDP-Lag	13:11 - 13:15
	WebDDoS	13:18 - 13:29
	SYN	13:29 - 13:34
	TFTP	13:35 - 17:15

III. Method Research

The naïve Bayes classifier is built on Bayes’ Theorem, where event independence is assumed. In statistics, two events are said to be independent if the likelihood of one does not impact the other [54]. Table 9 presents the algorithm of the Bayesian classifier to calculate probability. For instance, let P(B|A) equal the conditional probability of any given event. Then, let P(B) be the probability of B, and P(A) be the probability of A. Furthermore, let P(A|B) be equal to the likelihood of A given B. As such, the theorem is formally presented as:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Table 3 Pseudo code for the naïve Bayes algorithm.

```
Algorithm 1 Naïve Bayes
start
Let S = {a1, a2, …, an}, where S = training set and a = articles:
Calculate the probability of the classes P(C)
Calculate likelihood of attribute A for each class P(A|C)
Calculate the conditional probability P(C|A)
Assign the class with the highest probability
end
```

```
res1 = time. Time ()
from sklearn. naive_bayes import Gaussians
nb=Gaussians ()
nb= nb.fit(X_train , yttrian)
nb
res2 = time. Time ()
print ('Naive Bayes took ',res2-res1,' seconds')
accuracy = []
for tr_in,val_in in StratifiedKFold(shuffle = True,n_splits=5).split(X_val,y_ val):
        nb.fit (X_val.iloc[tr_in],y_val.iloc[tr_in])
        accuracy.append(nb.score(X_val.iloc[val_in],y_val.iloc[val_in]))
y_pred1 = nb.predict(X_test)
print ('Accuracy score= {:.8f}’. format(np.mean(accuracy)))
from sklearn. metrics import classification report, confusion matrix
print('\n')
print ("Precision, Recall, F1")
print('\n')
CR=classification_report(y_test, y_pred1)
print (CR)
print('\n')
```

IV. Result and Discussion

This section outlines the details of the design and implementation of the proposed solution. The solution is implemented in Python 3. Firstly, an overview of the solution is presented, briefly describing the phases of this implementation. describes the data preparation process, including details on data cleaning and transformation, and dataset splitting. presents the modelling process, with a detailed account of the training, validation and testing processes. concludes with an overview of the evaluation procedure, including a summary of the performance metrics used to analyse the intrusion detection performance of the DDoS datasets.

Data Cleaning and Transformation

Missing data. Handling missing data is vital in machine learning, as it could lead to incorrect predictions for any model. Accordingly, null values are eliminated by propagating the last valid observation forward along the column axis. This is implemented using the fillna method from the pandas library [52], as shown below.

```
data.fillna(method ='ffill', inplace = True)
```

Undefined Data. The elimination of null values can result in undefined data. A null field with no cells on its left becomes NaN after propagation, since there are no cells to provide a value. Consequently, these values are decoded into 0. This is all done using the fillna method [52].

```
data=data.fillna(0)
```

Transformation. The format of the collected data might not be suitable for modelling. In such cases, data and data types need to be transformed so that the data can then be fed into the models, as described by the CRISP-DM method. Accordingly, some data features were transformed into numeric or float, since models do not perform well with strings, or do not perform at all.

Class Labels. Each dataset instance represents a snapshot of the network traffic at a given point in time. These instances are labelled according to the nature of the traffic, that is, whether the traffic is benign or malicious. The labels across the four datasets vary, therefore they are encoded to have homogeneity in the class labelling system. Classification is binary, where benign traffic is labelled as NORMAL, and malicious traffic is labelled as ATTACK. Table 5 summarises the classification system.

Table 4 Labelling system for binary classification.

Label	Scenario
NORMAL	Traffic is benign
ATTACK	Traffic is malicious

A. *Volume and Class Distribution*

In the CICDDoS2019 dataset, there were 121,980 (41.4%) records classified as normal traffic and 172,647 (58.6%) classified as attack traffic.

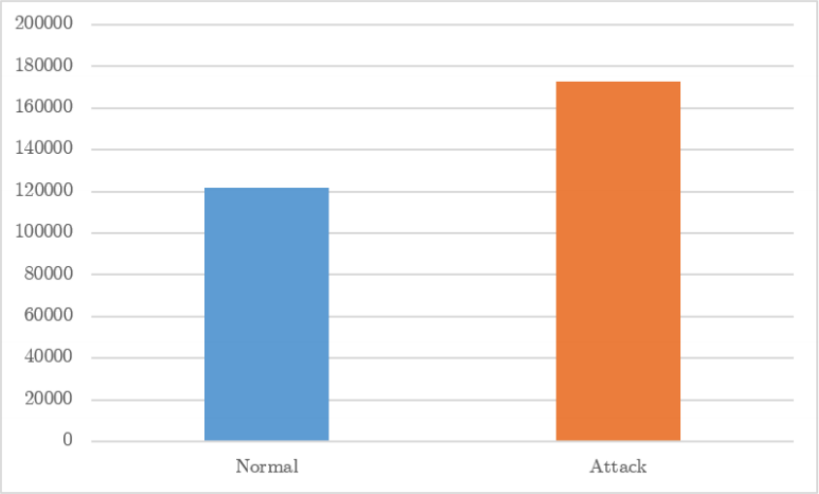

Fig. 1. Bar chart showing the spreading of traffic type in the CICDDoS2019 dataset.

B. *Splitting Datasets*

A key characteristic of a good learning model is its ability to generalise to new, or unseen, data. A model which is too close to a particular set of data is described as overfit, and therefore, will not perform well with unseen data. A generalised model requires exposure to multiple variations of input samples. Primarily, models require two sets of data, one to train and another to test. The training data is the set of instances that the model trains on, while the testing data is used to evaluate the generalisability of the model, that is, the performance of the model with unseen data. The train/test split can yield good results; however, this approach has some drawbacks. Although splitting is random, it can happen that the split creates imbalance between the training and the testing set, where the training set has a large number of instances from only one class. In such cases, the model fails to generalise and overfits. To mitigate this, the datasets are split into three subsets; training, validation and testing. This split is done in a 60:20:20 ratio, for training, validation and testing respectively. The train_test_split helper method from the scikit-learn library [53] is used for the split, as presented in the code snippet below. With this approach, training is done in two phases, with the training and the validation sets. Firstly, the training set is used to train the model. Then, the validation set is used to estimate the performance of the model on unseen data (data that the model is not trained on). For the purpose of this study, validation is done using a stratified k-fold approach. The k-fold validation method is described in Section 6.3.3.

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.40, random_state=100)
 X_val,X_test,y_val,y_test =
train_test_split(X_test,y_test,test_size=0.5,random_state=100)
```

During the training process, the selected algorithms are provided with training data to learn from to eventually create machine learning models. Accordingly, the training set is used, as specified in Section. At this point in the process, the input data source needs to be provided and should contain the target attribute (class label). The training process involves finding patterns in the training set that map the input features with the target attribute. Based on the observed patterns, a model is produced.

In this study, four DDoS datasets are being used as the input data source, where the target attribute is the type of network traffic, that is, attack or normal. Six algorithms are trained with each of the four sets. Training is conducted using several methods from the scikit-learn libraries. Table provides breakdown of the methods used for each algorithm. Appendix B contains the sources code for the models that were built to analyse the intrusion detection capacity of each dataset.

Following the training process, the model is validated using k-fold cross validation. Cross validation is applied to assess the generalisability of a model. This method aims to reduce the errors of overfitting that occur when a model is too closely fit a range of data instances. Cross validation is done in iterations, and each iteration involves splitting the dataset into k subsets, referred to as folds. The model is trained on k-1 folds, and the other fold is held back for testing, as illustrated in Figure 11. This process is repeated until all folds have served as a test fold. Once the process is completed, the evaluation metric is summarised by calculating the average value [54].

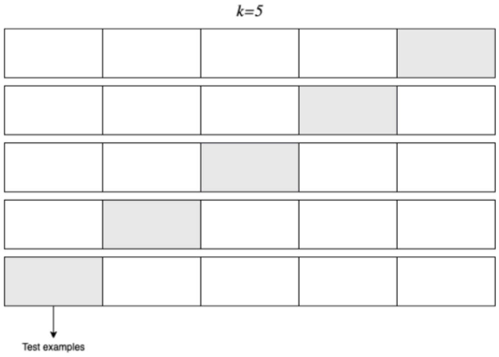

Fig. 2. K-fold cross validation with 5 folds.

In this study, a stratified k-fold approach is used using the validation dataset (20% of the global set). Stratified k-fold is a variation of k-fold cross validation that ensures that the distribution of classes is the same across all folds. This is implemented using the StratifiedKFold method from the scikit-learn library [64], with k=5. Below is a code snippet of the stratified k-fold, where n_splits specifies the number of folds.

```
for tr_in,val_in in StratifiedKFold(shuffle True,n_splits=5).split(X_val,y_val):
{{model}}.fit(X_val.iloc[tr_in],y_val.iloc[tr_in])
accuracy.append(knn.score(X_val.iloc[val_in],y_val.iloc[val_in]))
```

Fig 12 illustrates a comparative bar graph for the accuracy rates achieved by models that were trained with the CICDDoS2019 dataset [47]. From initial observations, it is clear that the naïve Bayes model performs poorly in comparison to the rest, with an accuracy rate of 45% (see table 15). The F-measure of the same model is also low. Taking a more granular look into this metric, it shows that both the precision and recall of the model are problematic, with 66% and 54% respectively. For this dataset, the best performing model was the random forest, achieving an accuracy of 99%, with a 99% precision and 99% recall. The model also took the longest to train, with a computation time of 84.2 seconds. Meanwhile, the other models took under 10 seconds to train.

V. Conclution

The Naïve Bayes model performed relatively poor overall and produced the lowest accuracy score of this study (45%) when trained with the CICDDoS2019 dataset [47]. For the same model, precision was 66% and recall was 54%, meaning that almost half the time, the model misses to identify threats.

Acknowledgment

Thanks to Universitas Amikom Yogyakarta

References

[1] S. Dua and X. Du, Data Mining and Machine Learning in Cybersecurity. Boca Raton, Florida: Auerbach Publications, 2016.

[2] C. Canongia and R. Mandarino, “Cybersecurity: The new challenge of the information society,” in Handbook of Research on Business Social Networking: Organizational, Managerial, and Technological Dimensions, 2011.

[3] P. Twomey, “Cyber Security Threats.” The Lowy Institute for International Policy, Sydney, 2010.

[4] R. Von Solms and J. Van Niekerk, “From information security to cyber security,” Comput. Secur., vol. 38, pp. 97–102, 2013.

[5] J. B. Fraley and J. Cannady, “The promise of machine learning in cybersecurity,” in SouthEastCon 2017, 2017, pp. 1–6.

[6] OWASP, “OWASP Top 10 - 2017 - The Ten Most Critical Web Application Security Risks,” Top 10 2017, 2017.

[7] C. Douligeris and A. Mitrokotsa, “DDoS attacks and defense mechanisms: Classification and state-of-the-art,” Comput. Networks, vol. 44, no. 5, pp. 643–666, 2004.

[8] S. K. Sahu, S. Sarangi, and S. K. Jena, “A detail analysis on intrusion detection datasets,” in Souvenir of the 2014 IEEE International Advance Computing Conference, IACC 2014, 2014, pp. 1348–1353.

[9] J. O. Nehinbe, “A critical evaluation of datasets for investigating IDSs and IPSs researches,” in Proceedings of 2011, 10th IEEE International Conference on Cybernetic Intelligent Systems, CIS 2011, 2011, pp. 1–6.

[10] A. Gharib, I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “An Evaluation Framework for Intrusion Detection Dataset,” in ICISS 2016 - 2016 International Conference on Information Science and Security, 2017, pp. 1–6.

[11] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, “A survey of networkbased intrusion detection data sets,” Comput. Secur., vol. 86, pp. 147–167, 2019.

[12] O. Yavanoglu and M. Aydos, “A review on cyber security datasets for machine learning algorithms,” in Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017, 2017, pp. 2186–2193.

[13] C. Thomas, V. Sharma, and N. Balakrishnan, “Usefulness of DARPA dataset for intrusion detection system evaluation,” in Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security 2008, 2008.

[14] L. Dhanabal and S. P. Shantharajah, “A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms,” Int. J. Adv. Res. Comput. Commun. Eng., vol. 4, no. 6, 2015.

[15] M. Małowidzki, P. Berezi, and M. Mazur, “Network Intrusion Detection: Half a Kingdom for a Good Dataset,” in ECCWS 2017 16th European Conference on Cyber Warfare and Security, 2017.

[16] R. Bace and P. Mell, “NIST special publication on intrusion detection systems,” Special Publication (NIST SP), 2001.

[17] Nexus Guard, “Nexusguard Research Shows DNS Amplification Attacks Grew Nearly 4,800% Year-over-Year; Highlighted by Sharp Increase in TCP SYN Flood,” 2019. [Online]. Available: https://www.nexusguard.com/newsroom/press-release/dns-amplificationattacks-rise-twofold-in-q1-0-0.

[18] J. Mirkovic and P. Reiher, “A taxonomy of DDoS attack and DDoS defense mechanisms,” Comput. Commun. Rev., vol. 34, no. 2, pp. 39–53, 2004.

[19] K. Scarfone and P. Mell, “Guide to Intrusion Detection and Prevention Systems (IDPS), National Institute of Standards and Technology. Special Publication (NIST SP), 2007.

[20] P. Ferguson and D. Senie, “Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing,” RFC Editor, 2000. [Online]. Available: https://tools.ietf.org/html/rfc2827.

[21] G. C. Kessler and D. E. Levin, Denial-of-Service Attacks, 4th ed. John Wiley & Sons, 2015.

[22] R. Das and T. H. Morris, “Machine learning and cyber security,” in 2017 International Conference on Computer, Electrical and Communication Engineering, ICCECE 2017, 2018, pp. 1–7.

[23] I. Sofi, A. Mahajan, and V. Mansotra, “Machine Learning Techniques used for the Detection and Analysis of Modern Types of DDoS Attacks,” Int. Res. J. Eng. Technol., 2017.

[24] N. Sharma, A. Mahajan, and V. Mansotra, “Machine Learning Techniques Used in Detection of DOS Attacks: A Literature Review,” Int. J. Adv. Res. Comput. Sci. Softw. Eng., 2016.

[25] M. Zekri, S. El Kafhali, N. Aboutabit, and Y. Saadi, “DDoS attack detection using machine learning techniques in cloud computing environments,” in Proceedings of 2017 International Conference of Cloud Computing Technologies and Applications, CloudTech 2017, 2018.

[26] D. M. Farid, N. Harbi, E. Bahri, M. Z. Rahman, and C. M. Rahman, “Attacks classification in adaptive intrusion detection using decision tree,” World Acad. Sci. Eng. Technol., pp. 368–372, 2010.

[27] Y. C. Wu, H. R. Tseng, W. Yang, and R. H. Jan, “DDoS detection and traceback with decision tree and grey relational analysis,” in 3rd International Conference on Multimedia and Ubiquitous Engineering, MUE 2009, 2009.

[28] A. Andhare, P. Arvind, and B. Patil, “Denial-of-Service Attack Detection Using GeneticBased Algorithm,” vol. 2, no. 2, pp. 94–98, 2012.

[29] M. Aamir and S. M. A. Zaidi, “DDoS attack detection with feature engineering and machine learning: the framework and performance evaluation,” Int. J. Inf. Secur., pp. 1–25, 2019.

[30] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, “Survey of intrusion detection systems: techniques, datasets and challenges,” Cybersecurity, vol. 2, no. 1, p. 20, 2019.

[31] R. Koch, “Towards next-generation intrusion detection,” in 2011 3rd International Conference on Cyber Conflict, ICCC 2011 - Proceedings, 2011.

[32] J. O. Nehinbe, “A simple method for improving intrusion detections in corporate networks,” in Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, 2010.

[33] . Moustafa and J. Slay, “UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set),” in 2015 Military Communications and Information Systems Conference, MilCIS 2015 - Proceedings, 2015.

[34] M. Ring, S. Wunderlich, D. Grüdl, D. Landes, and A. Hotho, “Flow-based benchmark data sets for intrusion detection,” in European Conference on Information Warfare and Security, ECCWS, 2017

[35] M. Ghorbani, Ali A., Lu, Wei, Tavallaee, Network Intrusion Detection and Prevention. Springer, 2010.

[36] The Cooperative Association for Internet Data Analysis, “CAIDA - The Cooperative Association for Internet Data Analysis,” CAIDA. 2010.

[37] J. Mchugh, “Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory,” ACM Trans. Inf. Syst. Secur., vol. 3, no. 4, pp. 1094–9224, 2000.

[38] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, “Detailed Analysis of the KDD CUP 99 Data Set,” Submitted to Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009. .

[39] University Of California, “KDD-Cup Dataset ’99,” The UCI KDD Archive, 1999. .

[40] University Of California, “KDD-Cup Dataset ’98,” The UCI KDD Archive, 1998. .

[41] J. Heidemann and C. Papadopoulos, “Uses and challenges for network datasets,” in Proceedings - Cybersecurity Applications and Technology Conference for Homeland Security, CATCH 2009, 2009.

[42] Defense Advanced Research Projects Agency, “1999 DARPA Intrusion Detection Evaluation Dataset,” 1999. [Online]. Available: https://www.ll.mit.edu/r-d/datasets/1999- darpa-intrusion-detection-evaluation-dataset.

[43] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, “Network anomaly detection: Methods, systems and tools,” IEEE Commun. Surv. Tutorials, vol. 16, no. 1, pp. 303–336, 2014.

[44] A. Nisioti, A. Mylonas, P. D. Yoo, and V. Katos, “From intrusion detection to attacker attribution: A comprehensive survey of unsupervised methods,” IEEE Commun. Surv. Tutorials, vol. 20, no. 4, pp. 3369–3388, 2018.

[45] T. H. Morris, Z. Thornton, and I. Turnipseed, “Industrial Control System Simulation and Data Logging for Intrusion Detection System Research,” Seventh Annu. Southeast. Cyber Secur. Summit, 2015.

[46] R. Wirth, “CRISP-DM : Towards a Standard Process Model for Data Mining,” Proc. Fourth Int. Conf. Pract. Appl. Knowl. Discov. Data Min., pp. 29–39, 2000.

[47] University of New Brunswick, “DDoS Evaluation Dataset (CICDDoS2019),” unb.ca, 2019. [Online]. Available: https://www.unb.ca/cic/datasets/ddos-2019.html.

[48] University of New Brunswick, “CSE-CIC-IDS2018 on AWS,” 2018. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2018.html.

[49] F. Beer, T. Hofer, D. Karimi, and U. Bühler, “A new attack composition for network security,” in Lecture Notes in Informatics (LNI), Proceedings - Series of the Gesellschaft fur Informatik (GI), 2017.

[50] Canadian Institute for Cybersecurity, “CICIDS2017,” unb.ca, 2017. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html.

[51] A. H. Lashkari, Y. Zang, G. Owhuo, M. S. I. Mamun, and G. D. Gil, “CICFlowMeter,” Github. 2017.

[52] Pandas.pydaya.org, “Pandas.Dataframe.Fillna,” Pandas 1.0.3 Documentation, 2014. [Online]. Available: https://pandas.pydata.org/pandasdocs/stable/reference/api/pandas.DataFrame.fillna.html.

[53] Scikit-learn, “Train_test_split,” Scikit-learn 0.22.2 Documentation, 2019. [Online]. Available: https://scikitlearn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html.

[54] T. Mitchell, Machine Learning. Burr Ridge, IL: McGraw Hill, 1997.

[55] M. Mohri, A. Rostamizadeh, and A. Talwalkar, Foundations of Machine Learning, 2nd ed. London, England: The MIT Press, 2018.

[56] L. Rokach, “Ensemble-based classifiers,” Artif. Intell. Rev., vol. 33, pp. 1–39, 2010.

[57] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, and B. Thirion, “Scikit-learn: Machine learning in Python,” J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

[58] Scikit-learn, “KNeighborsClassier,” scikit-learn.org, 2019. [Online]. Available: https://scikitlearn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html.

[59] Scikit-learn, “LinearSVC,” scikit-learn.org, 2019. [Online]. Available: https://scikitlearn.org/stable/modules/generated/sklearn.svm.LinearSVC.html.

[60] Scikit-learn, “GaussianNB,” scikit-learn.org, 2019. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html.

[61] Scikit-learn, “DecisionTreeClassifier,” scikit-learn.org, 2019. [Online]. Available: https://scikitlearn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html.

[62] Scikit-learn, “RandomForestClassifier,” scikit-learn.org, 2019. [Online]. Available: https://scikitlearn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.

[63] Scikit-learn, “LogisticRegressionClassifier,” scikit-learn.org, 2019. [Online]. Available: https://scikitlearn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html.

[64] Scikit-learn, “StratifiedKFold,” Scikit-learn 0.22.2 Documentation, 2019. .

[65] D. M. Powers, “Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation,” J. Mach. Learn. Technol., vol. 2, 2007.

[66] A. a, Alfantookh, “DoS Attacks Intelligent Detection using Neural Networks,” J. King Saud Univ. - Comput. Inf. Sci., Vol. 18, no. 2006, pp. 31-51, 2006.

[67] Alfine Ridho, M, Molavi Arman, “Analisis Serang DDoS Menggunakan Metode Jaringan Saraf Tiruan”, Sisfokom, Palembang, vol. 09. No. 03, PP 373-379, 2020.

[68] Alfa Saleh, Implementasi Metode Klasifikasi Naïve Bayes Dalam Memprediksi Besarnya Penggunaan Listrik Rumah Tangga , Yogyakarta, Citec Journal, Vol. 2, No. 3, Mei 2015 – Juli 2015.

[69] Al Riza Khadafy, Romi Satria Wahono, “Penerapan Naive Bayes untuk Mengurangi Data Noise pada Klasifikasi Multi Kelas dengan Decision Tree”, Jakarta, Journal of Intelligent Systems, Vol. 1, No. 2, December 2015.

[70] Arif Wirawan Muhammad, Cik Feresa Mohd Foozy, Ahmad Azhari, “Machine Learning-Based Distributed Denial of Service Attack Detection on Intrusion Detection System Regarding to Feature Selection”, New South Wales, vol. 4, no. 1, pp. 01-08, 2020.

[71] Aziz Muhammad, Rusydi Umar, Faizin Ridho, “Implementasi Jaringan Saraf Tiruan Untuk Untuk Mendeteksi Serang DDoS Pada Forensik Jaringan”, Yogyakarta, Vol. 3, No. 01, 2019.

[72] Chris Jordan Sihombing Jodi, Dany Primanita Kartikasari, Adhitya Bhawiyuga, “Implementasi Sistem Deteksi dan Mitigasi Serangan Distributed Denial of Service (DDoS) menggunakan SVM Classifier pada Arsitektur Software Defined Network (SDN)”, Malang, Vol. 3, No. 10, halm. 9608-9613, 2019

[73] Dong Shi, Khushnood Abbas, Raj Jain, “A Survey on Distributed Denial of Service (DDoS) Attact in SDN and Cloud Computing Environments”, India, Vol. 7, pp. 80813-80828, 2019.

[74] E. D. Meutia, J. Teknik, E. Universitas, and S. Kuala, “Internet of Things–Keamanan dan Privasi,” in *Seminar Nasional dan Expo Teknik Elektro ISSN*, 2015, pp. 2088–9984.

[75] Fitriyani,Romi Satria Wahono, “Integrasi Bagging dan Greedy Forward Selection pada Prediksi Cacat Software dengan Menggunakan Naïve Bayes”, Jakartaa, Journal of Software Engineering, Vol. 1, No. 2, December 2015.

[76] I. Alsmadi and D. Xu, “Security of Software Defined Network: A Survey, ” Comput. Secur., vol. 53, pp. 79-108, 2015.

[77] Iman Sharafaldin, Arash Habibi Lashkari, Saqib Hakak, and Ali A. Ghorbani, "Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy", IEEE 53rd International Carnahan Conference on Security Technology, Chennai, India, 2019.

[78] J. S. Reddy, “ZigBee Security,” pp. 1–22, 2004.

[79] Kurniabudi, Abdul Harris, Abdul Rahim, “Seleksi Fitur dengan Information Gain untuk Meningkatkan Deteksi Serangan DDoS Menggunakan Random Forest”, Jambi, Vol. 19, No. 1, halm. 56-66, 2020.

[80] K. Hengst, “DDoS through the Internet of Things,” pp. 1-9, 2016.

[81] K. Masica and K. Masica, “Recommended Practices Guide For Securing ZigBee Wireless Networks in Process Control System Environments Networks in Process Control,” Program, no. April, 2007.

[82] Lila Dini Utami, Romi Satria Wahono, “Integrasi Metode Information Gain Untuk Seleksi Fitur dan Adaboost Untuk Mengurangi Bias Pada Analisis Sentimen Review Restoran Menggunakan Algoritma Naïve Bayes”, Jakarta, Journal of Intelligent Systems, Vol. 1, No. 2, December 2015

[83] R. Sokullu, “GTS Attack : An IEEE 802 . 15 . 4 MAC Layer Attack in Wireless Sensor Networks,” Int. J., vol. 2, no. 1, pp. 105–116, 2009.

[84] S. Nadila, Y. Galuh, S. Fatia, K. Fahmi Hayati Holle, Deteksi Serangan Distributed Denial of Services (DDOS) Berbasis HTTP Menggunakan Metode Fuzzy Sugeno, JISKa, Vol. 4, No. 3, Pp. 156 – 164, Januari 2020.

[85] S. Raza, L. Wallgren, and T. Voigt, “SVELTE: Real-time intrusion detection in the Internet of Things,” Ad Hoc Networks, vol. 11, no. 8, pp. 2661–2674, 2013.

[86] Sukmawati Anggraini Putri, Romi Satria Wahono,” Integrasi SMOTE dan Information Gain pada Naive Bayes untuk Prediksi Cacat Software”, Jakarta, Journal of Software Engineering, Vol. 1, No. 2, December 2015.

[87] V. Hema and C. E. Shyni, “DoS Attack Detection Based on Naive Bayes Classifier,” Middle-East J. Sci. Red. Signal Process. Secur., Vol, 23, pp. 398- 405, 2015.

[88] W. Razouk, G. V. Crosby, and A. Sekkaki, “New security approach for ZigBee weaknesses,” Procedia Comput. Sci., vol. 37, pp. 376–381, 2014.