Detection of dos attacks using naive bayes method based on internet of things (iot)

F F Setiadi1,a, M W A Kesiman 1,b and K Y E Aryanto 1,c

1Department of Computer Science, Graduate Program, Universitas Pendidikan Ganesha, Singaraja, Bali, Indonesia

Email: afransferysetiadi@gmail.com, bantara.kesiman@undiksha.ac.id, cyota.ernanda@undiksha.ac.id

Abstract. Internet of Things (IoT) is a technology that is currently on a trend. Interconnecting networks on IoT are useful in the automation process, but it has vulnerabilities to network-based disruptions and attacks; such as Denial of Service (DoS). This study aimed at implementing the Naive Bayes algorithm to predict attribute classes using training data-sets from NSLKDD with the KDD99 format and obtained testing-data from the logging process of DoS attacks on IoT-based devices. The advantage of using Naive Bayes is that this method only requires a small amount of training data to determine the estimated parameters needed in the classification process. The results of the conducted research could detect attacks on IoT devices by using the help of snort tools to capture traffic logs. The results from the log were then converted into KDD99 format and processed by the Naive Bayes method. This research uses a training dataset from NSLKDD with KDD99 format which is widely used in various studies and testing data obtained from the IDS log process on the Raspberry Pi 3. The attributes used are 9 attributes namely; service, flag, src_bytes, dst_bytes, srv_serror_rate, same_srv_rate, diff_srv_rate, dst_host_srv_diff_host_rate and dst_host_srv_serror_rate. The results of the research analysis showed an accuracy of 64.02%. These results were smaller than previous results, but some aspects were still different from the actual results because the testing data and training data were taken from two different data-sets, thereby they had different characteristics.

1. Introduction

Intrusion Detection is analyzing and monitoring activity occurring in a computer system in sequence to detect signs of security problems. Currently, the issue of feature selection is the most focus of intrusion detection. It is because some features are irrelevant and redundant, which results in a long detection process and degrades the performance of an intrusion detection system (IDS) [1]. IoT has become quite familiar in recent years. The number of everyday routine devices is increasingly connected to the Internet, which includes many capabilities, such as sensing, autonomy and contextual awareness [2]. Interconnection networks in IoT can connect various objects that have identified identities and IP addresses, so they can communicate with each other and exchange data information [3].

Denial of Service, or commonly called DoS, is one of the attacks carried out through computer networks. Dos is a type of an attack on a computer or server in the internet network that makes the computer unable to perform and function properly and correctly [4]. According to Witten et al. [5], the Naive Bayes algorithm is an algorithm that uses a statistical approach in making decisions. The Naive Bayes Algorithm is based on the Bayes Theorem that all attributes contribute equally important and
independently of a certain class. The advantage of using Naive Bayes is that this method only requires a small amount of training data to determine the estimated parameters required in the classification process. Naive Bayes often performs much better in most complex real-world situations than might be expected [6]. Different from the previous study, the present study proposed to predict whether or not the attribute classes would be an anomaly or normal by using training data-sets from NSL-KDD with the KDD99 format and by testing data obtained from the log process of DoS attacks on IoT-based devices. The method used in this research was Naive Bayes. This study used NSLKDD data as a training data and snort IDS on IoT as a data testing.

2. Literature Review
A study showed that despite having a simple structure, Naive Bayes gave very competitive result [7]. Considered three levels of attack granularity depending on whether it deals with the entire attack, or classifies it into four main categories, or focuses solely on normal and anomaly behavior. In the whole experiment, the performance of Naive Bayes networks was compared with one of the well-known machine learning techniques, the decision tree. Besides, we compared the good performance of Bayes nets concerning the best results performed on the KDD99. The results of this study indicate that the decision tree shows a yield at 86% and Naive Bayes is only slightly different at 85%.

The previous research explained that IDS uses the KDDCUP99 data-set. The data used in this study were NSL-KDD data. NSL-KDD has provided a training data and a testing data for the attack classification of the research process. From the NSL-KDD data, the attack classification will be carried out using the Naive Bayes method so that new attacks can be classified. The KDDCUP '99 data-set is preprocessing data based on 1998 DARPA data provided for intrusion detection system design, which is used to evaluate different methodologies of intrusion detection. Research using the Naive Bayes method has succeeded to classify the attacks with the accuracy in a range of 81-84.67% [8].

2.1. Intrusion Detection System (IDS)
IDS is a system that can detect suspicious activity in a system or network. If suspicious activities are found related to network traffic, the IDS will give a warning to the system or network administrator. Several reasons for obtaining and using an IDS [9], include:

a. Prevent the increasing security risk, because many illegal activities are found by irresponsible people and punishments are given for these activities.

b. Detect network system security attacks and breaches that cannot be prevented by common systems such as firewalls, so they cause so many security holes, such users who don't understand the system, so the networks and protocols they use have security holes and users and administrators make mistakes in configuration and in using the system.

c. Detecting the initial attack, the attacker will attack a system that usually takes the first steps that are easily known, namely by investigating or testing the network system that will be the target, to get to the points where they will enter.

d. Secure files out of the network.

e. As a controller for security design and administrators, especially for fast companies.

f. Provide accurate information on the disturbance directly, improve diagnosis, recovery, and correct the factors that cause any attacks.

2.2. Dataset
Dataset is a collection of organized information data which is arranged to facilitate a data retrieval. A data set corresponds to one or more database tables, where each table column represents a specific variable, and each row corresponds to a specific record of the intended data-set [10]. The data-set lists values for each variable, such as the object's height and weight, for each member of the data set. A data-set can also consist of a collection of documents or files.
2.2.1. **NSL-KDD Dataset.** NSL-KDD is a recommended dataset for solving some inherent problems mentioned in the KDD'99 dataset. Although the new KDD dataset is still experiencing some problems and may not be a perfect representation of the existing real network, due to the lack of public data sets for network-based IDS, we believe it can still be applied as an effective benchmark data set to help researchers compare various intrusion detection methods [11]. In addition, the number of the records in the NSL-KDD training and testing sets is compatible. This advantage makes it affordable to run experiments on the full set without the need to randomly select a fraction. As a result, evaluation results from different research jobs will be consistent and comparable[12].

2.2.2. **KDD'99 Dataset Format.** KDD'99 dataset (kdd.ics.uci.edu) is pre-processing data based on 1998 DARPA data provided for intrusion detection system design, which is used to evaluate different methodologies of intrusion detection. In 1999, pre-processing of tcpdump data was carried out to be used in intrusion detection in the "International Knowledge Discovery and Data Mining Tools Competition" activity [13]. The KDD'99 format has 41 features consisting of three feature groups, namely the basic features of individual TCP connections, the content features in the connections suggested by the domain and the traffic features calculated by using time [14].

3. **Methods**

This section explains how this research was carried out so that it can provide details about the flow or steps that are made systematically and can be used clearly in solving problems, and analysing the research results.

3.1. **System Identification**

This is the design and implementation phase of DoS Attack Detection Using the Naive Bayes Method on the Internet of Things (IoT) Technology-Based Device which will be used as the object of research. Consists of several components in the form of:

a. Preparation of hardware devices
b. Simulated attacks
c. Preparation of testing data using KDD99 format
d. Preparation of data training that will use NSL-KDD data which can be obtained at www.unb.ca/cic/datasets/nsl.html
e. Feature selection to find out which feature subset of the dataset is the most significant feature
f. Implementation of the Naive Bayes method in classifying the types of attacks in the form of normal or anomaly
g. Requires a web interface used to present the results of the classification of attack types.

3.2. **Hardware Device**

In the first stage is the preparation of the used devices. The used devices include Routers, Switches, Access Points, PC Client and Raspberry Pi3 as IDS. The router in this study uses the Mikrotik RB450R2 Router with port 1 connected to the internet and port 2 connected to the switch. All devices such as PC, access point and raspberry Pi3 are connected to a switch.

3.3. **Testing Data Collection Simulation**

In Error! Reference source not found., the proposed IDS architecture for this simulation explains that one core switch is placed with the Router, so that users who are accessing it will have to pass through the core switch so that it can be monitored by the IDS server. DOS attacks will try to make the attacker prevent users from accessing the network system in several ways such as traffic flooding, which is to flood network traffic in large amounts of data so that users cannot access the network system. Also, it can flood the network with many requests for network services provided by the host so that requests that come from registered users cannot be served by a service which is also known as request flooding.
When the attacker attacks, the system will detect it as an attack. All data filtered by the system will be entered into a log that will be processed into the data testing. Log data will be converted using KDD99 format and become a testing.

3.4. Dataset Used

This section describes the used training dataset and the testing dataset. The training data is obtained from the NSL KDD training which can be obtained at www.unb.ca/cic/datasets/nsl.html. In this study, using full training data of 125,973 consisting of 67,343 normal classes data and 58,630 anomaly data. While the testing data used in this study were 4,906 consisting of 3,122 normal classes data and 1,784 anomaly data.

3.5. Naïve Bayes Method

The Naïve Bayes Classifier (NBC) is a simple probabilistic classifier that applies the Bayes Theorem with the assumption of high independence.[15] In the implementation of the IDS system, DoS Attack Detection uses the Naïve Bayes method. Naïve Bayes is a data mining method that is commonly used in text-based document classification. Currently, there is still data using continuous value attribute. There are several ways to deal with this continuous attribute; one of which is by modelling or calculating the continuous attribute data directly into the probability or probability function. The most commonly used is the Gaussian function [16]. The advantage of this method is it is a simple algorithm with low calculation complexity. However, the Naïve Bayes method has a weakness where the independence of the Naïve Bayes feature cannot always be applied, thereby it will affect the accuracy of the calculation [17]. However, the Naïve Bayes method can be optimized by using the KDD99 format so that the data gets its own weight. There are several flows in implementing the Naive Bayes method which can be seen as follows:

![Figure 1. The flow of the Naive Bayes method.](image)

3.5.1 Feature Selection. Feature selection is an important technique and is often used in pre-processing. This technique reduces the number of features involved in determining a target class value, reduces irrelevant, redundant features and data that cause misunderstanding of the target class [18]. The feature selection method used in this study contains two stages of feature selection that will be carried out using Information Gain (IG) and Correlation Feature Selection (CFS).

The first stage of this experiment is to select features with filter techniques to eliminate insignificant features using the Info Gain evaluator. IG is a feature selection technique that uses a scoring method for nominal or continuous attribute weighting, which is discounted using maximum entropy. The reason for using IG to select a feature is to declare the feature with the most significant information to the category. The features of the dataset that is having a Gain Ratio value equal to zero will be eliminated [19].

The next stage is feature selection based on their correlation using CFS. The CFS method is a part of the heuristic evaluation that considers the benefits of individual features for class predictions together with the level of inter-correlation among them. The CFS assigns high scores as a subset of data that contains features with high correlation with class and yet low interrelationships with each other [20]. CFS will select several attributes that have the greatest value which state the eligibility of these attributes to be selected. The results of the 41 field feature selection, nine fields were selected to be used as a reference for the training data and testing data. That fields are: service, flags, src_bytes, dst_bytes, srv_serror_rate, same_srv_rate, diff_srv_rate, dst_host_serror_rate and dst_host_srv_serror_rate.
3.5.2 Counting Class Probability. Class Probabilities are the probability value of each class of problems from the expected goals. The dataset has 2 classes of problem, so that the probability (P) of each class can be found by dividing the value of the frequency of data for each class of problem, by the total value of the data frequency related class. Therefore, from the calculations based on the provided training data, the probability value of each class of problem is obtained.

3.5.3 Counting Conditional Probability. The next step is to calculate Conditional Probabilities, which is the probability of each input value against the class value. In the calculations, there are two kinds of calculations. First, for discrete data it is done by calculating the number and probability. Second, for data in the continues data, it is necessary to remodel the shape of the data by finding the mean and standard deviation using the Gaussian function in Figure 2.

Figure 2. Flow for counting conditional probability as continues data.

a. Counting Mean Value. The equation of the mean value is as follows:

\[ \mu = \frac{\sum_{i=1}^{n} x_i}{n} \] (1)

b. Counting Standard Deviation Value. The equation the standard deviation value is as follows:

\[ \sigma = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \mu)^2}{n - 1}} \] (2)

c. Counting with Gaussian Function. To figure the conditional probabilities of the continues data, it is recommended to use the Gaussian Function. The equation of the Gaussian Function can be seen as follows:

\[ P(X_i = x_i | Y = y_i) = \frac{1}{\sqrt{2\pi\sigma_{ij}}} e^{-\frac{(x_i - \mu_{ij})^2}{2\sigma_{ij}^2}} \] (3)

3.5.4 Choose Max Score from Probability. The last step is to choose the maximum score from two probabilities between normal and anomaly. The maximum score will be the predicted data.

4. Results and Discussion
Testing the results using the Naive Bayes method can be done by finding how much accuracy has been achieved using the Confusion Matrix. The total testing data used was 3,122 data based on the actual data and predictive data. The two probabilities will be used are normal and anomaly. Normal is the condition if the traffic flow is normal, like idle or browsing. The anomaly is when the condition of the traffic is under attack. From these results, the accuracy of the Naive Bayes method could be determined.

4.1. The Results of the Confusion Matrix of the Naive Bayes Method.
This section will discuss the results of the confusion matrix using Naive Bayes from all the testing data. The calculation of the Accuracy is using the confusion matrix. A confusion matrix is a method that is usually used to calculate the accuracy of the concept of data mining. The results of the confusion matrix can be seen in the following table:
Table 1. The results of the confusion matrix.

| Predicted | Actual | positive | negative |
|-----------|--------|----------|----------|
| positive  | 1366   | 1756     |
| negative  | 9      | 1775     |

From the confusion matrix table, there are several values can be named:

a. **True Positive (TP).** It is normal prediction data and normal data. From Table 1 shows the TP data is 1366
b. **False Positive (FP).** It is normal prediction data and the actual data is an anomaly. From Table 1 shows the FP data is 9
c. **False Negative (FN).** It is anomaly prediction data and the actual data is normal. From Table 1 shows the FN data is 1756
d. **True Negative (TN).** It is the predicted data for anomalies and the actual data is anomalies. From Table 1 shows the TN data is 1775

This result explains that from 3,122 normal data, only 1,366 data are predicted to be normal explained in TP, while 1,756 normal data are predicted as anomaly data explained in FN. This is because the dataset cannot meet the normal data requirements so that there are many errors in predicting the data. Therefore, there are many differences in the normal data used in training and testing data sets.

While from 1,784 anomaly data, only 9 data for which actual anomaly data were predicted as normal were described in FP and the remaining 1,775 anomaly data were predicted as anomaly data correctly, which is described in TN. This is because the dataset has the necessary anomaly data requirements. So for anomaly data, there are many similarities in the training and testing data sets.

4.2. The results of the Recall, Precision, Error rate and Accuracy of the Naive Bayes method.

From the confusion matrix table we can find the Recall, Precision, Error rate and Accuracy values. The results of these values can be seen in the following table.

Table 2. The results of the Recall, Precision, Error rate and Accuracy values.

| Recall (%) | Precision (%) | Error rate (%) | Accuracy (%) |
|------------|---------------|----------------|--------------|
| 99.50      | 50.27         | 35.98          | 64.02        |

Based on the Table 2, it can be explained that:

a. The Recall is the ratio of true positive predictions to overall correct data. The result of the recall is 99.50%. These results indicate the success of the model in recovering information to classify predicted data as normal compared to the entire dataset which is 99.50%.
b. Precision is the ratio of true positive predictions to overall positive predictions. The result of the precision is 50.27%. These results indicate that the level of accuracy between the requested data and the predicted results given by the model classifies the data that are in fact normal from the entire dataset which is 50.27%.
c. The error rate is the ratio of false predictions (positive and negative) to the overall data. The result of the error rate was 35.98%. These results indicate that the accuracy of the model to classify correct data, predicted to be normal or anomaly, of the entire dataset is 35.98%.
d. Accuracy is the ratio of accurate predictions (positive and negative) to the overall data. The result of the accuracy was 64.02%. These results indicate that the accuracy of the model to classify the correct data, predicted to be normal or anomaly, of the entire dataset is 64.02%.

This result is lower than the results of Prasetyo’s et al. research[8] with the results of 81–84.67%. This is because the results of Prasetyo’s et al research used the NSLKDD training and testing dataset, while the researchers used the testing data from IoT-based devices, which had different characteristics.

So this result has advantages in the process of obtaining testing data that already has the value of each attribute, so there is no need to give manual weights because we have changed the data in KDD99 format. However, it has a weakness as its accuracy results is lower due to the results of the study using a different dataset. The problem that might arise from this weakness is an error in determining whether the traffic is safe or there is an indication of an attack. This is because of the low level of the accuracy. In implementing IoT related to Dos detection, the data set used can be considered meeting the requested accuracy level.

5. Conclusion and Future Works
Interconnection networks on IoT, apart from being useful in the automation process, are vulnerable to network-based attacks and disruptions such as DoS. This study aims to classify the detection of DoS attack systems on a network, but specifically for IoT devices that use the Raspberry pi. Making classifiers needs training and testing datasets. This study uses a training dataset from NSLKDD with KDD99 format. The testing data is obtained from the IDS log process on the Raspberry Pi 3 device.

NSLKDD is one of the widely used datasets, it's just that this dataset is not specifically used for IoT devices. However, the dataset can still reflect a network, even though it does not use IoT devices due to the behaviour in the network or attacks will still exist. In order to find out that the training dataset can still work in an IoT device network, it is necessary to build a new testing dataset. This is because NSLKDD is not a specific dataset to the IoT network but is suitable as an initial training data to further learn about the system.

Based on the research results, the conclusions obtained from this study are, this system is designed using the Naive Bayes method in which this method is used to predict class attributes. The dataset uses a training dataset from NSLKDD with KDD99 format, which is widely used in various studies. The testing data is obtained from the IDS Log process on the Raspberry Pi3 device. The attributes used are the 9 attributes, namely service, flag, src_bytes, dst_bytes, srv_serror_rate, same_srv_rate, diff_srv_rate, dst_host_srv_diff_host_rate and dst_host_srv_serror_rate. This system uses the Google Spreadsheet tool as a tool in predicting class and Google AppSheet as a tool in the interface using 125,973 training data consisting of 67,343 normal data and 58,630 anomaly data. Additionally, the testing data used in this study were 4,906 testing data consisting of 3,122 data with predicted normal class and 1,784 anomaly data. Referring to the results of DoS Attack Detection Using the Naive Bayes Method on the Internet of Things (IoT) Technology-Based Devices, Table 2 shows an accurate result of 64.02%. This result is lower than the previous study conducted by Prasetyo., Et al. This is because the results of previous research used the training and testing dataset from the NSLKDD. However, the researchers in this current study used the testing data from IoT-based devices which had different characteristics.

For some suggestions that can be given to For the future works of researchers who want to develop a system it is suggested to do a system analysis using a method other than Naive Bayes in addition to determine the consistency of the obtained results. Due to getting a good enough accuracy when classifying testing data, the training set must be able to represent the state of the testing data, it is necessary to do a training dataset using a dataset other than NSLKDD. This research is only limited to training datasets from system simulations. The suggestions for further research is to develop how to get the training datasets at large companies that specialise in datasets.
References

[1] Atamli A W and Martin A 2014 Threat-based security analysis for the internet of things *Proceedings - 2014 International Workshop on Secure Internet of Things, SloT 2014*

[2] Marsel Sampe Asang & Irwan Sembiring 2017 Keamanan Data Pada Perangkat Internet Of Things Menggunakan Metode Public-Key Cryptography *Teknol. Informasi-Aiti* 14 80–7

[3] Rizal R 2018 Network Forensics Untuk Mendeteksi Serangan Flooding Pada Perangkat Internet Of Things (IoT ) *J. Teknol. dan Sist. Komput.* 67

[4] Nihri H, Pramukantor E S and Trisnawani P H 2018 Pengembangan IDS Berbasis J48 Untuk Mendeteksi Serangan DoS Pada Perangkat Middleware IoT *J. Pengemb. Teknol. inf. dan Ilmu Komput. Univ. Brawijaya* 2

[5] Witten I H, Frank E and Hall M a 2011 *Data Mining: Practical Machine Learning Tools and Techniques (Google eBook)*

[6] Kanagalakshmi R and Raj V N 2014 Network Intrusion Detection Using Hidden Naive Bayes Multiclass Classifier Model 76–84

[7] Amor N Ben, Benferhat S and Elouedi Z 2004 Naive Bayes vs decision trees in intrusion detection systems *Proc. ACM Symp. Appl. Comput.* 1 420–4

[8] Prasetyo A, Aflandi L and Arpandi D 2018 Implementasi metode naive bayes untuk intrusion detection system (ids) *J. Inform. Polinema* 4 280–4

[9] Mukherjee S and Sharma N 2012 Intrusion Detection using Naive Bayes Classifier with Feature Reduction *Procedia Technol.* 4 119–28

[10] Jupriyadi 2018 Implementasi Seleksi Fitur Menggunakan Algoritma FVBRM Untuk Klasifikasi Serangan Pada Intrusion Detection System (Ids) *Semin. Nas. Teknol. Inf.* 17 1–6

[11] Dhanabal L and Shantharajah S P 2015 A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms *Int. J. Adv. Res. Comput. Commun. Eng.* 4 446–52

[12] Wahba Y, ElSalamouny E and ElTaweel G 2015 Improving the Performance of Multi-class Intrusion Detection Systems using Feature Reduction

[13] El-Sappagh S, Mohammed A S and AlSheshtawy T A 2019 Classification Procedures for Intrusion Detection Based on Kdd Cup 99 Data Set *Int. J. Netw. Secur. Its Appl.* 11 21–9

[14] Siddiqui M K and Naahid S 2013 Analysis of KDD CUP 99 Dataset using Clustering based Data Mining *Int. J. Database Theory Appl.* 6 23–34

[15] Kusrini S E D A 2017 Algoritma K-Means untuk Diskretisasi Numerik Kontinyu Pada Klasifikasi Intrusion Detection System Menggunakan Naive Bayes *Konf. Nas. Sist. Infom.* 61 6

[16] Panda M and Patra M R 2007 Network Intrusion Detection Using Naive Bayes *IJCSNS Int. J. Comput. Sci. Netw. Secur.* 7 258–63

[17] Wirawan I N T and Ekististanto I 2015 Penerapan Naive Bayes Pada Intrusion Detection System Dengan Diskritisasi Variabel *JUTI J. Ilm. Teknol. Inf.* 13 182

[18] Chae H, Jo B, Choi S and Park T 2013 Feature Selection for Intrusion Detection using NSL-KDD *Recent Adv. Comput. Sci.* 20132 184–7

[19] Ginting A, Isnanto R R and Windasari I P 2015 Implementasi Algoritma Kriptografi RSA untuk Enkripsi dan Dekripsi Email *J. Teknol. dan Sist. Komput.* 3 253

[20] Jain Y K 2012 An Efficient Intrusion Detection Based on Decision Tree Classifier Using Feature Reduction *Int. J. Sci. Res. Publ.* 2 1–6