Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th October 2025. Vol. 5. No. 1; e-ISSN: 2808-4519

Implementation of the Decision Tree Method for Detecting Attacks in Networks

Nissi Sari M A Sitanggang[1*], Octara Pribadi[2], Herman[3]

[1,2,3]*Information Systems Study Program, Stmik Time Medan*
margaretthastg1707@gmail.com[1*], octarapribadi@gmail.com[2], hrman_ang@yahoo.com[3]

Abstract

The rapid development of internet technology has led to an increase in cybercrime, including attacks on computer networks such as Distributed Denial of Service (DDoS). These attacks can cause serious disruption to systems and steal important data. Therefore, early detection of attacks is very important to maintain network security. One effective method for detecting anomalies in networks is the Decision Tree algorithm, due to its ability to classify data quickly and easily. This study aims to develop a Decision Tree-based attack detection model using the CICIDS 2017 dataset. The results of the study show that this algorithm is capable of classifying attack data with a high degree of accuracy and is reliable in handling large-scale data. Thus, the resulting model can help improve the security and performance of computer networks.

Keywords: *Accuracy, Network Security, CICIDS 2017, Decision Tree, DDoS*

1. Introduction

Decision Tree is an algorithm that is often used because of its ability to perform classification and regression based on a series of easy-to-understand decision rules. This algorithm works by building a tree model that is then used to categorize new data based on the rules generated. The main advantage of this method is its ability to handle large and complex data with relatively fast computation time [1]. Cybercrime is currently a widespread issue, with computers being used as tools for criminal activities. These crimes include various offenses such as theft, fraud, and attacks on computer systems and networks [2]. It is the responsibility of computer systems to protect data from user failure or unauthorized access by unauthorized parties. Computer networks are valuable assets that must be protected, both physically and non-physically [1]. One of the main challenges in network security management is vulnerability to attacks or system destruction. The number of attacks caused by system vulnerabilities through networks has increased significantly [1]. With the rapid development of internet technology, an increasing number of irresponsible individuals are exploiting this to launch attacks to gain access to computer systems, thereby obtaining important data and gaining an advantage for themselves [3]. Vulnerability to attacks and actions that cause system damage are the main challenges in managing network security. DDoS detection can be performed using machine learning, one of which is the Decision Tree algorithm [4]. This research aims to address these issues by developing and evaluating an effective Decision Tree model for anomaly detection, thereby enhancing network security and performance [1]. Based on this background, the author is interested in conducting research and has chosen the title "Implementation of the Decision Tree Method for Network Attack Detection."

2. Research methods

2.1. Analysis

Network analysis refers to the process of examining, monitoring, and evaluating data or activities in a computer network to understand patterns, detect anomalies, or improve network performance and security. The steps taken in this study can be seen in Fig 1.

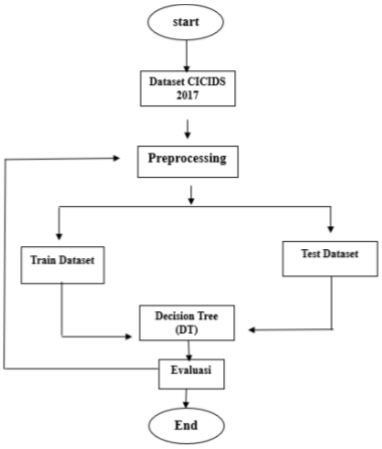

Fig. 1: Research Stages

The research stages can be explained as follows:

1. Data collection: In the data collection stage of this research, the CICIDS 2017 dataset was used.
2. Dataset division: After the data was collected, the dataset from CICIDS 2017 was divided into two parts, one part for training (Train) and the other part for testing (Test).
3. Model training: The dataset for training (train) will be used to train the model using the Distributed Denial of Service (DDOS) algorithm. After the training process is complete, the resulting model will be tested using the test dataset.
4. Model Evaluation: Several evaluation metrics are used to measure the test results, including 80% accuracy, 75% precision, 85% recall, and a 90% F1 score. Accuracy measures how well the model can predict the actual results, precision measures how accurately the model predicts true positive results, and F1 score is the harmonic mean of precision and recall. By calculating these metrics, we can assess how well the model predicts low-rate DDoS attacks on network traffic.

2.2. Data pre-processing

Data preprocessing is an important step that aims to convert data into a format that can be understood by the system. Most of the data used for analysis is often incomplete or inconsistent, so data preprocessing is necessary to improve the accuracy and efficiency of the results obtained. Data preprocessing is crucial in analyzing network traffic because the data has different types and dimensions:

1. Data Cleaning, the first stage in preprocessing is data cleaning. This process is important because there is often missing or corrupted data. Understand the source of the problem, whether it is caused by errors in the data transfer process, system problems, or other causes.
2. Numeric Attribute Normalization, after data cleaning, the next stage is the normalization of numeric and categorical attributes. Before doing so, it is important to understand the difference between numerical attributes and categorical attributes. Numerical attributes are used for data that can be measured quantitatively and processed using mathematical operations. Numerical attributes include interval and ratio data. Meanwhile, categorical attributes are used for data that cannot be measured quantitatively and cannot be processed with mathematical operations, such as addition or multiplication. Categorical attributes consist of nominal, binary, and ordinal data.
3. Categorical Attribute Normalization: In the CICIDS 2017 dataset, there is a Label column containing two categories, namely normal class and attack class. Artificial neural networks cannot process text-based data, so categorical attributes must be normalized into numerical attributes before processing. Through normalization, the attack class, which was originally text-based, will be converted into the number 1.

2.3. Confusion matrix

The confusion matrix is an important step in ensuring that data is ready for use in a decision tree model. As explained in the research objectives, this study aims to test the performance of the proposed attack detection method. In several Intrusion Detection System (IDS) and anomaly detection studies, accuracy matrices are used to measure the accuracy of a detection system, including: (i) Sensitivity and specificity, (ii) Misclassification rate, (iii) Confusion matrix entries, (iv) Precision-recall and F measures. For accuracy evaluation, a confusion matrix is used as shown in Table 1.

Tabel 1: Matrik Konfusi

		Prediksi	
		Normal	Serangan
Aktual	Normal Serangan	TP	FP
		FN	TN

In the context of IDS 3.1, it can be explained as follows:

1. FP (False Positive): The number of normal data that is incorrectly detected as an attack.

2. FN (False Negative): This is a prediction error where an actual attack is detected as normal.
3. TP (True Positive): This is the number of attacks that are correctly detected as attacks.
4. TN (Train Negative): This is a condition where the attack detection system successfully identifies normal traffic as normal.

2.4. Train dataset

The test dataset is part of the dataset used to test the performance of a machine learning model after the model has been trained using the train dataset. The test dataset contains data that the model has never seen during the training process, so it can be used to evaluate the model's ability to recognize patterns or detect attacks on new data. The test dataset used is 30%.

2.5. Design

The design process includes identifying requirements, designing the system, and selecting the dataset to be used. The design stages are described as follows:

2.5.1. Libarary

1. Pandas: The function of pandas is to read, process, and analyze data in formats such as CSV.
2. Numpy: The function of numpy is to perform numerical operations on arrays and matrices. Scikit-learn Its function is to split data into train and test, normalize or standardize features, and encode target labels.
3. Matplotlib: Its function is to create basic graphs such as histograms, scatter plots, or line plots.
4. Dask: Its function is to process large datasets in parallel.
5. PyOD: Its function is specifically used to detect anomalies in data.

2.5.2. System requirements specifications

This research was conducted through simulations using various tools and operating systems, as shown in Table 2.

Tabel 2: System Requirements Specifications

1.	System Operasi Virtual	Ubuntu
2.	Simulasi Tools	Virtualbox
3.	Dataset	CICIDS 2017
4.	CPU	AMD Ryzen 5 7000 series with nvidia geforce Rtx
5.	RAM	8 GB
6.	Operating System	Windows 11
7.	Supporting Tools	Jupyterlab Notebook,Sklearn Library,Matplotlib, Library, Python 3

2.5.3 Decision tree

A decision tree is a classification method that uses a tree structure, where each node represents an attribute, while the leaves of the tree indicate a class. The topmost node of the decision tree is called the root. Decision trees are the most popular classification method used. In addition to being relatively fast to use, the results of the model built are easy to understand [10]. The CICIDS 2017 deep dataset includes 2.8 million records after cleaning (after removing missing or duplicate data). There are two algorithms used in the creation of decision trees to measure the purity of a class in a node, namely Gini and Entropy [4]. The Gini Index decision tree algorithm for low-rate DDOS detection has an accuracy of 80%, precision of 82%, recall of 85%, and an F1 score of 87%. The Entropy Decision Tree algorithm has an accuracy of 75%, Precision of 80%, Recall of 80%, and an F1 Score of 85%. If accuracy in prediction is prioritized, the Gini Index algorithm may be more suitable. However, if Recall is prioritized, the Entropy method may be more suitable.
The parameters used in the CICIDS 2017 dataset are:

1. Flow Duration: Total duration of network flow sessions
2. Fwd IAT (Inter Arrival Time): Interval between packets sent forward
3. Bwd IA: Interval between packets received
4. Total Fwd Packets: Total packets sent by the sender
    a. Total Bwd Packets: Total packets received by the receiver
    b. Flow packets/s: Average number of packets sent per second
    c. Flow active time: Active connection time in milliseconds
    d. Flow idle time: Inactive connection time in milliseconds
    e. Average packet size: Average size of all packets in the session
    f. Min Segment size forward: Minimum segment size sent by the sender

3. Results and discussion

The following are the results of the thesis:

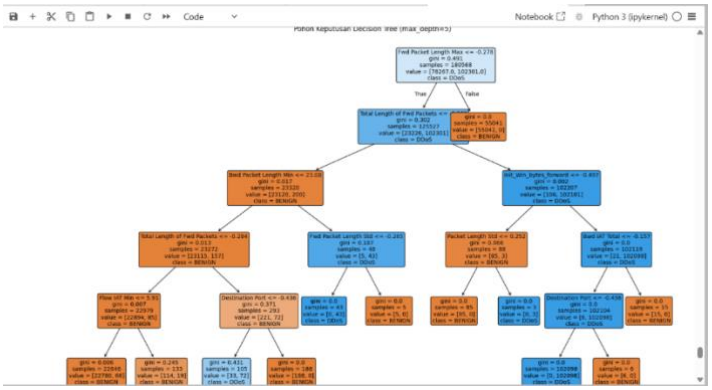

Fig. 2: Decision Tree

3.1. Discussion

The dataset used in this study is CICIDS 2017. This dataset was chosen because it contains various types of attacks and features relevant to network security analysis and accuracy testing, and includes network traffic covering benign (normal) data and various types of attacks such as DDoS, PortScan, Bot, Brute Force, Ftp Patator, SQL Injection, etc.

```
[84]: print("5 Fitur paling signifikan berdasarkan Decision Tree:")
      print(feat_importances.head(5)) #digunakan untuk menampilkan nilai yang lebih soignifican

5 Fitur paling signifikan berdasarkan Decision Tree:
                       Feature  Importance
6         Fwd Packet Length Max    0.573126
4   Total Length of Fwd Packets    0.420396
56       Init_Win_bytes_forward    0.002337
0              Destination Port    0.001232
11        Bwd Packet Length Min    0.000854
```

Fig. 3: Most significant data

Figure 3 shows the five most significant or important features or data in the CICIDS 2017 dataset.

3.2. Data preprocessing

Before being used in model training, data must go through several stages of preprocessing

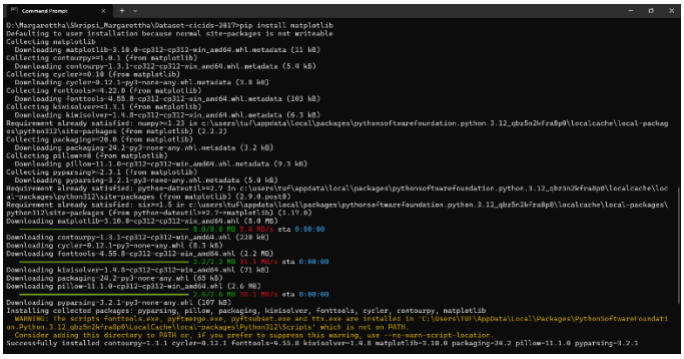

Fig. 4: Install Library

1. Library Installation: Adding or inserting libraries into Jupyter Notebook so that it can run code according to requirements.
2. Data Normalization: Normalizing feature scales using methods such as Min-Max Scaling or Standardization to make the values more uniform.
3. Data Division: Dividing the data into two parts, namely the training set and the testing set, with a certain ratio, for example, 80% for training and 20% for testing.

3.3. Data before processing

The following data has not yet undergone the data deletion process:

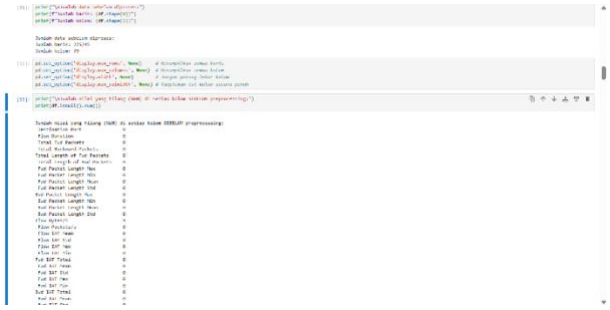

Fig. 5: before data deletion

Before the data was deleted, there were 225,745 rows and 79 columns. Data deletion was performed for the following reasons:

1. To remove irrelevant (unnecessary) data
2. To delete empty data
3. To reduce duplicate data (excessive data)
4. To improve model performance and speed

3.4. After data deletion

This removal is part of preprocessing, which aims to improve data quality before training with machine learning algorithms such as Decision Tree.

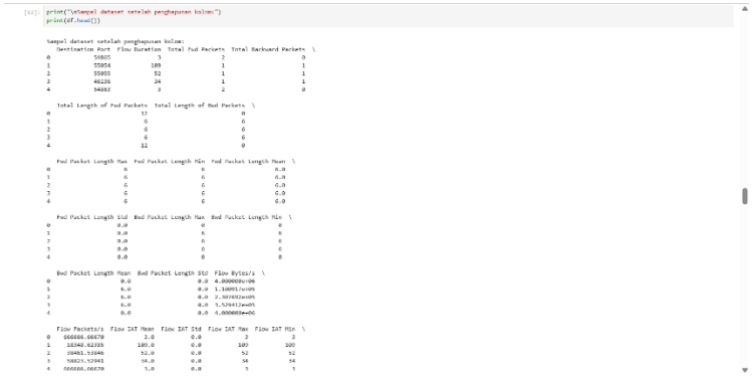

Fig. 6: after data deletion

Before the data was deleted or dropped, there were 225,711 rows and 69 columns.
The Flow ID data was deleted because it was unique and there was no information contained within it other than the IP or port of each stream or device used. Data deletion for Source ID and Destination IP because they only indicate the recipient and sender addresses of the packet using different data or IP addresses. Timestamp data was deleted because the occurrence of flows or packets does not directly indicate attack patterns and can cause the model to become unstable if data from different times is included. Source Port and Destination Port data were deleted because ports can change and do not consistently reflect attack types, making them less effective. Protocol data was deleted because the dataset is dominated by a single protocol, and this column may be redundant (sequential).

3.5. Data Sharing

Data splitting is the process of separating a dataset into two parts: training data and test data, data splitting is an important process in pre-processing before building and evaluating learning models in machine learning.

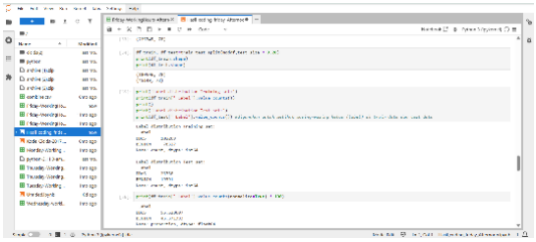

Fig. 7: Division of test data and train data

3.6. Implementation decision tree

Here is the decision tree

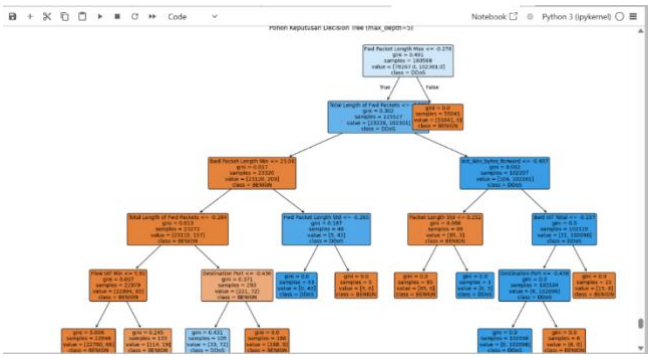

Fig. 8: Decision tree

A decision tree is a method in machine learning used for clarification and regression. In the case of network attack detection, decision trees are used to classify whether network traffic is normal or an attack.

3.6.1 prediction process

The predict method is used to predict labels from new data using methods that have already been trained.

Fig. 9: prediction model

1. Each test data is passed to the tree.
   a. Each row (test data) will start from the root of the tree.
   b. The model will check the features and follow the split conditions from node to node. For example: if feature1<= 0.5 -> left; else -> right. Reach the leaf.
2. After following the path to the leaf node, the model will: Output the majority label from the training data in that leaf node.
3. Repeat for All Rows: This process is repeated for all rows in X_test.

4. Conclusion

This chapter is the final chapter, which contains the conclusions and recommendations from the entire thesis. It includes the following paragraphs:

1. Research Objective: This research aims to implement the Decision Tree method in detecting attacks on computer networks using the CICIDS 2017 dataset. Based on the experimental results, the Decision Tree model is capable of classifying between normal traffic and DDoS attacks with high accuracy.
2. Decision Tree Model Results: The Decision Tree model built demonstrates an accuracy rate of 99%, with high precision, recall, and F1-score values, particularly for the DDoS attack class. The features that contributed most to the model's performance were Flow Duration, Fwd Packet Length Max, Bwd Packet Length Mean, Flow Bytes/s, and Packet Length Variance.
3. Advantages of the Decision Tree Method: The Decision Tree method proved to be effective because it was able to provide clear interpretation of the results through the decision tree structure. It is easy to understand and implement, and has relatively fast training time compared to other methods such as Random Forest or Neural Network.
4. Limitations of the Research: The CICIDS 2017 dataset is very large and unbalanced, so sampling and preprocessing were performed to avoid model bias. The model has not been tested in real-time conditions or in actual system implementations. This study only uses one algorithm (Decision Tree) and does not compare it with other algorithms such as SVM, K-NN, or Random Forest.

5. Suggestions

Based on the thesis research, the following conclusions can be drawn: It is recommended to use ensemble methods such as Random Forest or Gradient Boosting to compare performance with Decision Trees. Additionally, model testing should be conducted in real-time scenarios

to better reflect actual network system conditions. Consider using data balancing techniques like SMOTE to address data imbalance between normal traffic and attacks. For IDS development: Decision Tree models can serve as a foundation for building initial detection modules in an IDS system due to their clear interpretability. Model implementation should be accompanied by a robust logging and alerting system to facilitate attack monitoring.

References

[1] M. I. Manalu and F. P. Hutabarat, "Pendeteksian Anomali Trafik Jaringan Menggunakan Metode Decision Tree," vol. 01, no. 01, 2024.
[2] Y. I. Mahendra and R. E. Putra, "Penerapan Algoritma Gradient Boosted Decision Tree (GBDT) untuk Klasifikasi Serangan DDoS," vol. 06, pp. 158–166, 2024.
[3] M. Fadhlurrohman, A. Muliawati, and B. Hananto, "Analisis Kinerja Intrusion Detection System pada Deteksi Anomali dengan Metode Decision Tree Terhadap Serangan Siber," *J. Ilmu Komput. dan Agri-Informatika*, vol. 8, no. 2, pp. 90–94, 2021, doi: 10.29244/jika.8.2.90-94.
[4] F. Febriansyah, Z. Asti Dwiyanti, and Diash Firdaus, "Deteksi Serangan Low Rate Ddos Pada Jaringan Tradisional Menggunakan Machine Learning Dengan Algoritma Decision Tree," *Cyber Secur. dan Forensik Digit.*, vol. 6, no. 1, pp. 6–11, 2023, doi: 10.14421/csecurity.2023.6.1.3951.
[5] R. Muhendra, *KONSEP DASAR SISTEM INSTRUMENTASI , WIRELESS SENSOR NETWORK DAN INTERNET OF THINGS (IoT)*, no. 39. 2021. [Online]. Available: https://repository.ubharajaya.ac.id/27392/1/Gabung Sistem Instrumentasi.pdf
[6] K. Kurniabudi, A. Harris, and A. E. Mintaria, "Komparasi Information Gain, Gain Ratio, CFs-Bestfirst dan CFs-PSO Search Terhadap Performa Deteksi Anomali," *J. Media Inform. Budidarma*, vol. 5, no. 1, p. 332, 2021, doi: 10.30865/mib.v5i1.2258.
[7] U. Suriani, I. P. Ninditama, and W. Syaputra, "Pemodelan Prediktif Keterlambatan Bicara pada Balita Terkait dengan Penggunaan Smartphone Menggunakan Data Mining," vol. 5, no. 1, pp. 129–138, 2024, doi: 10.51519/journalita.v5i1.589.
[8] D. Firdaus, F. Fahira, and R. Rianti, "Deteksi Anomali Dan Serangan Low Rate Ddos Dalam Lalu Lintas Jaringan Menggunakan Naive Bayes," *Naratif J. Nas. Riset, Apl. dan Tek. Inform.*, vol. 5, no. 2, pp. 140–148, 2023, doi: 10.53580/naratif.v5i2.208.