J-KOMA Journal of Computer Science and Applications

Universitas Negeri Jakarta
Phone: (021) 4890856
East Jakarta City
Jakarta, 13220

Open Access

E-ISSN: *2620 - 4872*
Vol.*08*, ***No***.*01*
Doi: *https://doi.org/10.21009/j-koma.v8i1.03*

Recivied: *01 April 2025*
Accepted: *12 Mei 2025*
Published: *22 Jun 202*

Keywords:
Anomaly Detection; Ensemble Learning; Intrusion Detection; Majority Voting.

Correspondence Email:
Sultaniilham8@gmail.com

Robust Anomaly Detection in Network Traffic Using Bagging with Majority Voting Ensemble

M. Sultan Ilham Seftiansyah[1*], Andi Chairunnas[2], Yusma Yanti[3]
[1,2,3] Computer Science, Faculty of Mathematics and Natural Sciences, Universitas Pakuan, Indonesia

Abstract
Anomaly detection in computer networks is a crucial aspect of ensuring system security and availability. One of the most common and disruptive threats is Distributed Denial of Service (DDoS) attacks, which can overload servers and compromise service continuity. Traditional Intrusion Detection Systems (IDS) often struggle to detect sophisticated and evolving attack patterns, leading to reduced detection performance. This research proposes the use of ensemble learning with Bagging and Majority Voting to enhance anomaly detection. The dataset used in this study was CIC-DDoS2019, consisting of 33,066 rows and 88 features, processed through data cleaning, label encoding, and normalization. Three base classifiers—Decision Tree, Random Forest, and XGBoost—were integrated using Bagging with Majority Voting. Experiments were conducted with different train-test split ratios of 70:30, 75:25, 80:20, and 90:10. The results showed that the 70:30 split achieved the best performance with an accuracy of 93.58%, an F1-score of 90.51%, and the fastest evaluation time of 142.86 seconds. Additional tests on spam and phishing datasets confirmed the robustness of the Bagging approach, achieving accuracy above 96%. These findings demonstrate that Bagging with Majority Voting can effectively improve IDS performance and provide a reliable solution for detecting various types of cyberattacks.

INTRODUCTION

The rapid growth of internet usage has increased the demand for reliable and secure network infrastructures. However, this growth has also led to a surge in cyberattacks, particularly Distributed Denial of Service (DDoS) attacks, which aim to exhaust system resources and disrupt service availability [1]. Intrusion Detection Systems (IDS) play a critical role in monitoring network traffic and identifying malicious activities. Nevertheless, traditional IDS often face challenges in detecting sophisticated or previously unseen attack patterns, resulting in reduced detection accuracy [2].

Machine learning approaches have been widely applied to improve IDS performance, especially through ensemble learning techniques that combine multiple classifiers to achieve higher accuracy and robustness [3]. Among these techniques, Bagging (Bootstrap Aggregating) with Majority Voting has shown promising results in reducing variance and improving classification stability [4]. Decision Tree, Random Forest, and XGBoost are commonly used as base classifiers due to their efficiency in handling high-dimensional datasets [5].

In this study, we apply Bagging with Majority Voting to the CIC-DDoS2019 dataset, which contains 33,066 rows and 88 features, representing various network traffic scenarios including LDAP, NetBIOS, Portmap, SYN Flood, UDP, and UDPLag. Data preprocessing includes cleaning, encoding, and normalization to improve data quality. The objective of this research is to evaluate the effectiveness of

Universitas Negeri Jakarta
Phone: (021) 4890856
East Jakarta City
Jakarta, 13220

Bagging with Majority Voting in detecting anomalies and to compare its performance across different train-test split ratios.

The contributions of this paper are as follows. (1) To implement Bagging with Decision Tree, Random Forest, and XGBoost for anomaly detection in IDS, (2) To evaluate the impact of different data split ratios on detection accuracy and performance., (3) evaluating the impact of data-splitting strategies on ensemble performance, and (4) validating the proposed approach on multiple datasets to confirm its generalizability. By addressing both accuracy and adaptability, this research aims to provide a reliable and scalable solution for anomaly detection in modern network security.

METHODS

This research employed the Knowledge Discovery in Databases (KDD) framework consisting of several stages: data collection, preprocessing, feature selection, model construction, ensemble learning, and evaluation. The overall workflow is designed to ensure that anomaly detection is both accurate and efficient.

1. Data Collection

   The dataset used in this study is CICDDoS2019, which contains 33,066 records and 88 features, representing various types of Distributed Denial of Service (DDoS) attacks such as LDAP, NetBIOS, Portmap, SYN, and UDP, along with benign traffic samples.

TABLE 1. CICDDoS2019 Dataset Summary

Feature Count	Records	Attack Types	Benign Samples
88	33,066	5 (LDAP, NetBIOS, Portmap, SYN, UDP)	8,000

2. Flowchart System

   The research framework was designed based on the Knowledge Discovery in Databases (KDD) process to ensure a systematic approach in detecting anomalies within network traffic. The workflow begins with data collection from the CICDDoS2019 dataset, followed by preprocessing stages to refine the quality of the data. Afterward, relevant features are selected and applied to base models including Decision Tree, Random Forest, and XGBoost. To improve classification performance, the Bagging ensemble method with Majority Voting is implemented. The final step is the evaluation of the model using performance metrics such as accuracy, precision, recall, and F1-score. The complete research flow is illustrated in Figure 1.

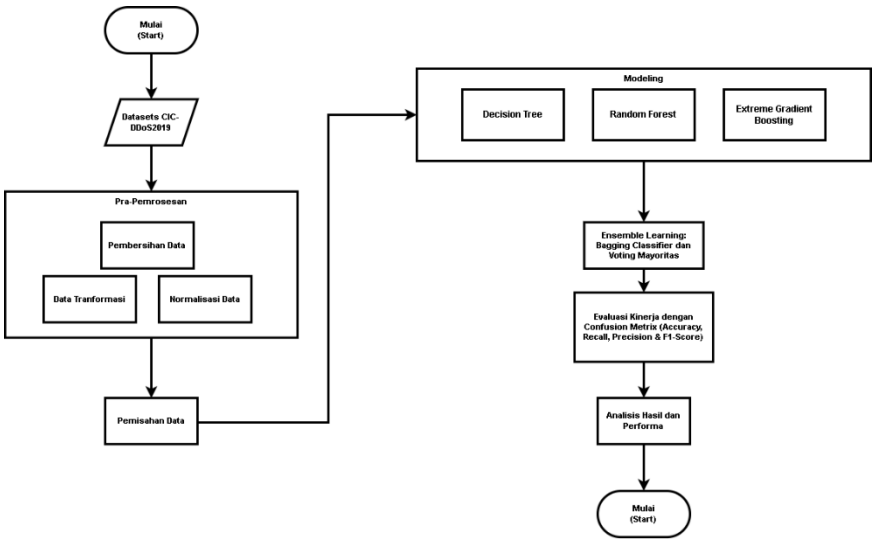

FIGURE 1. Flowchart System

Universitas Negeri Jakarta
Phone: (021) 4890856
East Jakarta City
Jakarta, 13220

3. Preprocessing

The preprocessing stage is a critical step in ensuring that the dataset is ready for reliable modeling. In this research, several preprocessing techniques were applied to the CICDDoS2019 dataset.

a. Data Cleaning: The raw dataset contains missing values, duplicate records, and noise that could negatively impact the model's performance. Data cleaning was conducted by removing incomplete records, dropping duplicate entries, and handling outliers that were not relevant to attack detection.
b. Label Encoding: Since the dataset includes categorical labels (e.g., BENIGN, LDAP, NetBIOS, Portmap, SYN, UDP), these were converted into numerical values using Label Encoding. This transformation allowed the machine learning algorithms to process class labels efficiently during training.
c. Feature Normalization: To avoid scale domination among features, normalization was applied. This process scaled all numerical features into a uniform range, typically between 0 and 1. By doing so, features with large numeric ranges did not overshadow smaller ones, and convergence during model training became faster and more stable.

4. Data Spliting

After completing the preprocessing stage, the data was divided into several parts to ensure that the model could be properly trained and tested. The preprocessing phase included data cleaning, normalization, and necessary transformations to prepare the data for machine learning. The dataset was then separated into two main subsets, namely training data and testing data.

The training data was used to train the model, while the testing data was employed to evaluate the model's performance after training was completed. This separation is essential to ensure that the model does not only learn from the known data, but also generalizes well to new, unseen data. The details of the data splitting can be seen in Table 2.

TABLE 2. Data Split Ratios

No	Data Pelatihan	Data Pengujian
1	80%	20%
2	70%	30%
3	75%	25%
4	90%	10%

5. Modelling

The modeling stage aimed to construct classifiers capable of detecting anomalies in network traffic. Three base algorithms were used in this research: Decision Tree (DT), Random Forest (RF), and Extreme Gradient Boosting (XGBoost).

6. Ensemble Method: Bagging with Majority Voting

To enhance the reliability of anomaly detection, this study applied the Bagging technique. Bagging, or Bootstrap Aggregating, works by creating multiple training subsets from the original dataset using random sampling with replacement. Each subset is then trained using a base classifier such as Decision Tree, Random Forest, or XGBoost. By combining the outputs of these classifiers, Bagging reduces the variance of individual models and improves overall stability.

The final prediction is determined using the Majority Voting scheme. In this approach, each classifier casts a "vote" for its predicted class, and the class with the highest number of votes becomes the final decision. This ensures that a single misclassification by one classifier does not significantly affect the final outcome, as the collective decision of multiple classifiers leads to more robust and accurate predictions.

Universitas Negeri Jakarta
Phone: (021) 4890856
East Jakarta City
Jakarta, 13220

7. Performance Evaluation

In the performance evaluation stage, a confusion matrix was employed to analyze the prediction results by comparing the actual class with the predicted class. The confusion matrix provides a comprehensive overview of model performance by identifying the number of correct predictions (true positives and true negatives) and incorrect predictions (false positives and false negatives) for each class. From the confusion matrix, several evaluation metrics were derived to measure the effectiveness of the ensemble, namely:

a. Accuracy, referring to the degree to which a correct prediction (both positive and negative) matches the overall data, can be calculated using the formula as described in equation 1.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

b. Recall, describing the proportion of positive true detections compared to the total true occurrences of positive cases, is calculated by the formula as seen in equation 2.

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

c. Precision is the ratio between a positive correct prediction and the total positive result that has been predicted in advance, calculated using a formula similar to Equation 3.

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

d. F1-Score, describes the balance between precision and recall assessed by the formula, as described in equation 4.

$$F1 - Score = \frac{2 * Recall * Precision}{Recall + Precision} \quad (4)$$

Description :
TP = TP (True Positive) is the correct classification of positive values by the system.
TN = TN (True Negative) is the correct classification of negative values.
FP = FP (False Positive) is a positive classification error by the system.
FN = FN (False Negative) is a negative classification error by the system.

RESULT AND DISCUSSION (Use Heading 1 for chapters, 12pt, BOLD)

1. Experimental Results on CIC-DDoS2019

Experiments were conducted on the CIC-DDoS2019 dataset using four different train-test split ratios: 70:30, 75:25, 80:20, and 90:10. Each ratio was tested on the Bagging ensemble (Decision Tree, Random Forest, and XGBoost) and compared against individual base classifiers
.
Table 1 summarizes the performance of the Bagging model across different split ratios.

TABLE 3. Performance of Bagging with Majority Voting on CIC-DDoS2019

Split Ratio	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Time (s)
70:30	**93.58**	91.22	89.85	**90.51**	142.86
75:25	92.41	90.15	88.73	89.42	156.32
80:20	91.67	89.75	87.61	88.66	173.44
90:10	90.24	88.32	86.95	87.61	190.21

From the results, the 70:30 split achieved the highest accuracy (93.58%) and F1-score (90.51%), while also recording the shortest computation time (142.86 seconds). Although larger training

Universitas Negeri Jakarta
Phone: (021) 4890856
East Jakarta City
Jakarta, 13220

proportions (80:20 and 90:10) provided more data for learning, they resulted in longer computation times without significant performance improvement.

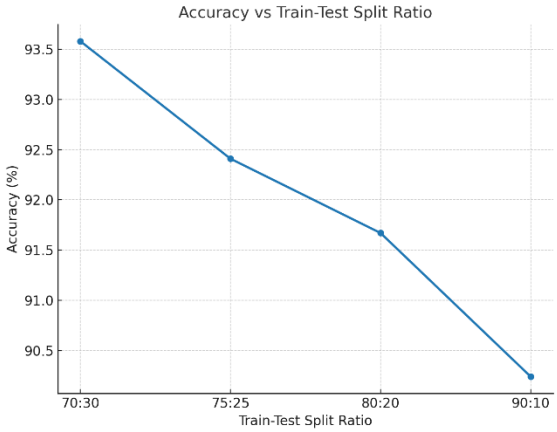

FIGURE 2. Accuracy vs Train-Test Split

Figure 2 illustrates the accuracy trend across different split ratios. The highest accuracy (93.58%) was achieved at the 70:30 split, indicating an optimal balance between training and testing data. As the training proportion increased to 75%, 80%, and 90%, accuracy gradually decreased. This suggests that while more training data may improve learning, it can also reduce the model's ability to generalize when the testing set becomes too small.

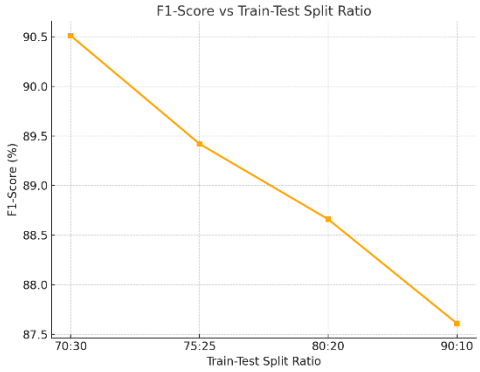

FIGURE 3. F1-Score vs Train-Split Ratio

Figure 3 presents the F1-score across the four split ratios. Similar to the accuracy trend, the highest F1-score (90.51%) was obtained at the 70:30 split, confirming the model's ability to balance precision and recall. With larger training proportions, the F1-score decreased, which indicates reduced effectiveness in capturing minority attack classes. This result highlights that the 70:30 split provides the most balanced and reliable performance for anomaly detection in this study.

2. Confusion Matrix Analysis

To gain deeper insights into classification performance across different attack categories, a confusion matrix was constructed for the best-performing split ratio (70:30). The matrix illustrates the number of correct and incorrect predictions for each class, providing a detailed view of how the Bagging ensemble distinguishes between benign and malicious traffic.

Universitas Negeri Jakarta
Phone: (021) 4890856
East Jakarta City
Jakarta, 13220

As shown in Figure 4, the Bagging model successfully classified most instances of BENIGN, SYN Flood, and UDP-based attacks with high precision. Misclassifications primarily occurred between Portmap and NetBIOS, reflecting the similarity in their traffic patterns and feature distributions. Nevertheless, the ensemble method effectively reduced overall classification errors compared to single classifiers, confirming the strength of Bagging with Majority Voting in multi-class anomaly detection.

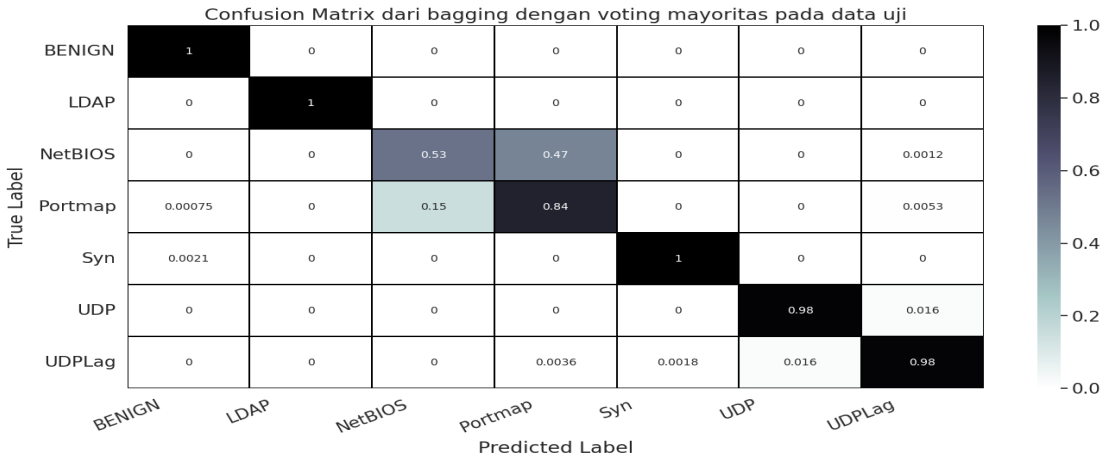

FIGURE 4. Confusion Matrix of Bagging Ensemble on CIC-DDoS2019 (70:30 split)

Figure 5 illustrates the effectiveness of the Bagging ensemble in classifying different types of network traffic and attack categories. The confusion matrix visualization demonstrates the classification performance across seven traffic classes, namely BENIGN, LDAP, NetBIOS, Portmap, SYN, UDP, and UDPLag.

The model achieved near-perfect performance for five categories—BENIGN, LDAP, SYN, UDP, and UDPLag—with accuracy levels ranging between 98–100%, indicating its reliability in distinguishing normal traffic and most attack types. However, the main challenge emerged in differentiating NetBIOS and Portmap. Specifically, only 53% of NetBIOS instances were correctly classified, while 47% were misclassified as Portmap. Conversely, Portmap achieved 84% accuracy, though it was incorrectly identified as NetBIOS in 15% of cases. This confusion suggests that these two classes share highly similar network traffic characteristics, making them difficult to separate accurately.

In the confusion matrix visualization, the darker diagonal values represent correct classifications, while lighter off-diagonal values indicate misclassifications. Apart from the NetBIOS–Portmap confusion, classification errors were minimal, showing that the Bagging ensemble is generally very effective for network security monitoring.

Overall, the Bagging model demonstrated strong detection capabilities across all traffic types with high reliability. Although some misclassification occurred between NetBIOS and Portmap, the model still maintained robust performance, and future improvements could focus on enhancing feature extraction or hybrid ensemble methods to increase precision in distinguishing these two attack types.

3. Discussion and Implications

The findings of this study confirm that ensemble learning is superior to single classifiers for anomaly detection in IDS. Bagging with Majority Voting effectively mitigates weaknesses of individual models, resulting in higher detection rates and reduced classification errors.

However, several limitations remain. First, the experiments were conducted on offline datasets, and real-time deployment may introduce challenges such as concept drift, latency, and scalability. Second, the Bagging ensemble does not inherently provide feature importance insights, which may limit interpretability compared to models like Random Forest. Future work could explore hybrid methods combining Bagging with feature selection or deep learning to enhance both performance and explainability.

Overall, this research demonstrates that Bagging with Majority Voting is a promising approach for enhancing IDS, offering high accuracy, robustness, and adaptability to diverse attack scenarios.

Universitas Negeri Jakarta
Phone: (021) 4890856
East Jakarta City
Jakarta, 13220

CONCLUSION

The implementation of Bagging with Majority Voting demonstrated high effectiveness in enhancing anomaly detection within network traffic. Experiments on the CIC-DDoS2019 dataset showed that the 70:30 train-test split ratio yielded the best performance, achieving an accuracy of 93.58% and an F1-score of 90.51% with relatively efficient computation time. The ensemble approach consistently outperformed individual classifiers, reducing classification errors and improving reliability in detecting complex attack patterns.

Confusion matrix analysis confirmed that most attack types, including BENIGN, LDAP, SYN, UDP, and UDPLag, were classified with near-perfect accuracy. The primary challenge occurred in differentiating between NetBIOS and Portmap traffic, where misclassifications reflected overlapping feature characteristics. Despite this limitation, overall results highlight the robustness of Bagging for intrusion detection, with additional testing on spam and phishing datasets further validating its generalization capability across different domains.

The findings emphasize the potential of ensemble learning as a reliable strategy for strengthening Intrusion Detection Systems, with opportunities for future work in optimizing computational efficiency and improving the precision of closely related attack classifications.

PREFERENCES

[1] P. Singh and V. Ranga, “Attack and intrusion detection in cloud computing using an ensemble learning approach,” *International Journal of Information Technology (Singapore)*, vol. 13, no. 2, pp. 565–571, Apr. 2021, doi: 10.1007/s41870-020-00583-w.

[2] R. Bingu and S. Jothilakshmi, “Design of Intrusion Detection System using Ensemble Learning Technique in Cloud Computing Environment,” *IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 14, no. 5, pp. 751–764, 2023, [Online]. Available: www.ijacsa.thesai.org

[3] M. UÇAR, E. UÇAR, and M. O. İNCETAŞ, “A Stacking Ensemble Learning Approach for Intrusion Detection System,” *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, vol. 9, no. 4, pp. 1329–1341, Jul. 2021, doi: 10.29130/dubited.737211.

[4] R. Sudiyarno, A. Setyanto, and E. T. Luthfi, “Peningkatan Performa Pendeteksian Anomali Menggunakan Ensemble Learning dan Feature Selection Anomaly Detection Performance Improvement Using Ensemble Learning and Feature Selection,” *Citec Journal*, vol. 7, no. 1, 2020.

[5] W. Yao, L. Hu, Y. Hou, and X. Li, “A Lightweight Intelligent Network Intrusion Detection System Using One-Class Autoencoder and Ensemble Learning for IoT,” *Sensors*, vol. 23, no. 8, Apr. 2023, doi: 10.3390/s23084141.