Indonesian Journal of Artificial Intelligence and Data Mining (IJAIDM)
Vol. 7, No. 2, September 2024, pp. 220 – 228
p-ISSN: 2614-3372 | e-ISSN: 2614-6150

Optimizing Performance Random Forest Algorithm Using Correlation-Based Feature Selection (CFS) Method to Improve Distributed Denial of Service (DDoS) Attack Detection Accuracy

[1]Sopian Soim, [2]Sholihin, [3]Cahyo Bayu Subianto
[1,2,3]Department of Electrical Engineering, Study Program of Telecommunication Engineering,
Politeknik Negeri Sriwijaya, Indonesia
Email: [1]sopiansoim@gmail.com, [2]sholihin@polsri.ac.id, [3]cahyobayu48@gmail.com

Article Info
Article history:
Received Jan 12th, 2024
Revised Mar 20th, 2024
Accepted Apr 11th, 2024

Keyword:
Accuracy
Correlation
DdoS
Machine Learning
Random Forest Algorithm
Selection Feature

ABSTRACT
In the ever-evolving digital era, Distributed Denial of Service (DdoS) attacks have become a major threat to the security of networks and online services, making it important to develop effective strategies to detect and overcome such attacks.This research aims to improve the performance of Random Forest algorithm in dealing with DdoS attacks by using Correlation-Based Feature Selection method. This method can identify and select the most relevant features from the dataset used, in this case the CIC-DdoS2019 dataset, with respect to accuracy, precision, recall, and F1-score as evaluation metrics, so that this research achieves the best results in effectively detecting and preventing DdoS attacks, making an important contribution in strengthening the security of networks and online services.The results show that the application of the Correlation-Based Feature Selection method is able to improve DdoS attack detection in a complex network context using the Random Forest algorithm, increasing the detection accuracy rate to 99.89%. These findings highlight the potential of using the Random Forest algorithm with the CFS method in improving DdoS attack detection in complex network environments.This study recorded a significant improvement compared to the previous study, which only achieved an accuracy rate of 99.7% using the feature importance method.

Corresponding Author:
Cahyo Bayu Subianto,
Departement of Electrical Engineering, Study Program of Telecommunication Engineering,
Politeknik Negeri Sriwijaya
Jl. Srijaya Negara, Bukit Besar, Kecamatan Ilir Barat I, Kota Palembang, Sumatera Selatan.
Email: cahyobayu48@gmail.com

DOI: http://dx.doi.org/10.24014/ijaidm.v7i2.24783

1. INTRODUCTION

The use of information technology is currently rapidly increasing in various sectors due to its numerous benefits, including improved productivity, quality, better working conditions, process transparency, and profitable business models. In its application, IT can enhance operational efficiency through process automation, improve the quality of products or services offered, and create a better work environment through efficient collaboration tools and communication [1]. Behind the rapid increase in the use of information technology lies the threat of attacks by irresponsible individuals who aim to sabotage systems. According to the monitoring report by the National Cyber and Crypto Agency (BSSN) in 2021, there were a total of 5,940 recorded cases of web defacement attacks. The academic sector, particularly universities, experienced the highest impact with 2,217 cases, followed by private companies in second place with 1,483 cases. The third most affected sector was local governments with 1,097 cases, and the central government ranked fourth with 477 cases [2].

One of the most commonly encountered cyber attacks is DdoS (Distributed Denial of Service). Unlike data theft or leakage, the objective of DdoS offense is to disrupt systems by inundating the target web server with a massive volume of fake traffic. This flood of traffic overwhelms the server with service requests, leading to a degradation in performance or even complete server downtime [3]. Detecting DdoS attacks typically involves deploying an IDS. An IDS is a system employed for surveillance real-time network traffic, and detect and report cyber attacks or unwanted activities on a computer network. It provides notifications or warning alarms, enabling proactive or corrective actions to mitigate the attack [4].

Along with the rapid development of network infrastructure, DdoS attacks have become increasingly sophisticated, resulting in more powerful attacks with the use of Maximum Transmission Unit (MTU) [5]. Therefore, it is important to develop and implement advanced technologies to improve the accuracy of identifying DdoS attacks. Machine Learning [6] has proven to be an appropriate solution for this purpose. Machine Learning is a field of computer science that focuses on creating systems that can learn and make decisions on their own, without explicit programming instructions [7]. Several previous studies have explored the potential of Machine Learning in this regard. For example, in a study by Wani et al. [8] in 2019, experiments using the Random Forest method for DdoS detection achieved an accuracy rate of 98%. Another study conducted by Chen et al. [9] in 2020 also adopted the random forest approach to identify DdoS attacks. This research uses a dataset consisting of four categories, namely LLDoS1.0, LLDoS2.0.1, UDP Flood Attack, and ICMP Flood Attack. The results of the study show a high level of accuracy, reaching 99.41% in accurately detecting DdoS attacks.Similarly, Alduailij et al. [10] conducted a study in 2022 that focused on comparing the performance of five machine learning algorithms – Gradient Boosting, KNN, Logistic Regression, Random Forest, and using feature importance method in detecting DdoS attacks. Out of all the algorithms, Random Forest achieved the utmost accuracy rate of 99.7%, outperforming the rest.

This study aims to experiment with DdoS identification and detection using random forest. The random forest algorithm was chosen because of its high accuracy and efficiency on datasets [11]. The results of this research will optimize the random forest algorithm by using the correlation feature selection method. This method is used to eliminate features that are irrelevant and provide minimal contribution to the model. This optimization is expected to significantly increase the accuracy of DdoS attack detection to above 99.7%, compared to previous research [10].

2. RESEARCH METHOD

This research framework will be presented in the form of a comprehensive diagram that provides a clear overview of the research steps. The diagram serves as a visual representation of the research process, allowing researchers to gain a comprehensive understanding of the entire study. It offers a guide that outlines the logical and organized sequence of steps in the research.

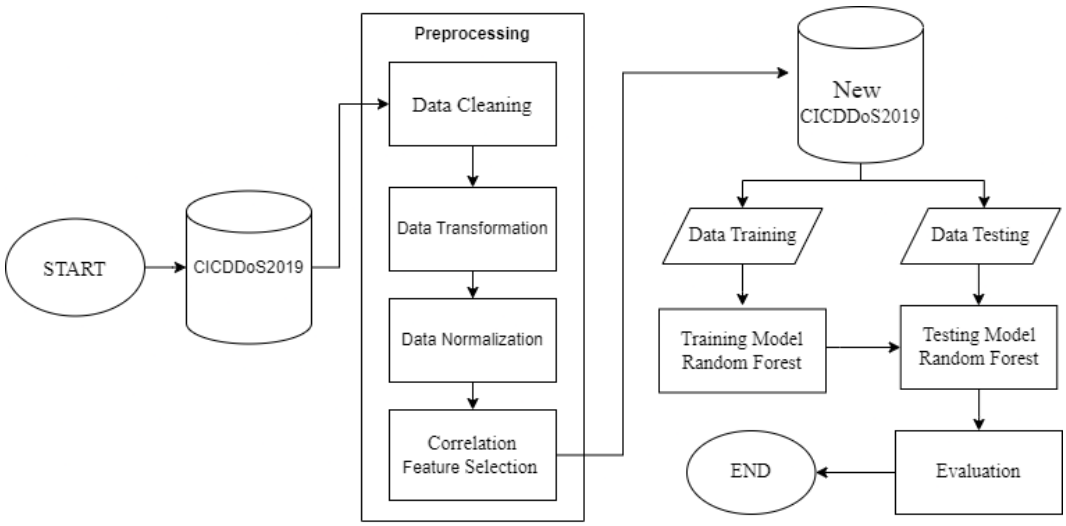

Figure 1. Research Methodology

2.2. Dataset

This research will utilize the CIC-DdoS2019 dataset obtained from the source [12]. This dataset addresses the limitations of previous datasets and offers a comprehensive set of network flow features that are crucial for detecting and classifying DdoS attacks. Additionally, it introduces a new taxonomy for categorizing DdoS attacks and shares feature sets with other NIDS CIC datasets, namely IDS2017, IDS2018, and DoS2017,

allowing for broader comparisons and analyses. It comprises 431,371 rows and 17 types of attacks. The dataset includes 97,831 instances of benign (non-malicious) traffic and 212,172 instances of DdoS attacks. More detailed information about the attack types can be found in Table 1.

Table 1. CIC-DdoS2019 Dataset Summary

Number	Type of Traffic	Number of Record
1.	DrDos_NTP	121,368
2.	TFTP	98,917
3.	Benign	97,831
4.	Syn	49,373
5.	UDP	18,090
6.	DrDoS_UDP	10,420
7.	UDP-lag	8,872
8.	MSSQL	8,523
9.	DrDoS_MSSQL	6,212
10.	DrDoS_DNS	3,669
11.	DrDoS_SNMP	2,717
12.	LDAP	1,906
13.	DrDoS_LDAP	1,440
14.	Portmap	685
15.	NetBIOS	644
16.	DrDoS_NetBIOS	598
17.	UDPLag	55
18	WebDDoS	51
	Total Traffic	431,371

2.2. Preprocessing

2.2.1. Data Cleansing

At this stage, the CIC-DdoS2019 dataset undergoes data cleansing. Data cleansing plays a vital role in enhancing the accuracy and effectiveness of machine learning. It helps eliminate noise, outliers, and errors in the data, reducing overfitting, and enhancing model efficiency [13]. We removed rows and columns that had duplicate values, NaN, infinite, and -infinite values. Then, the data were grouped based on the 'Label' column and filtered to include only the data groups with a data count exceeding 10,000. The purpose of this step was to obtain a meaningful subset of data with a sufficient quantity for further analysis or modeling.

2.2.2. Data Transformation

Data transformation refers to the process of converting or changing data from its original form to a different form, enabling it to be used more effectively or suitably for specific analysis or applications with the aim of improving data quality or meeting specific requirements in data analysis [14]. In the data transformation stage, I executed the conversion of int64 columns to int32 and float64 columns to float32 data types. This conversion was implemented to optimize memory utilization and enhance data processing speed. By converting the integer data to int32 and float data to float32, we achieved a significant reduction in memory usage and expedited operations on the transformed data types.

2.2.3. Data Normalization

Data normalization involves transforming the values within a dataset to a standardized range. Its purpose is to eliminate variations in scale among the features or variables, facilitating data comparison and analysis [15]. In this process, we employed undersampling using the RandomUnderSampler approach to handle class imbalance in the dataset through the use of reducing the number of samples from the majority class, achieving a balanced representation of the minority class. Subsequently, the undersampled variables underwent Z-score scaling, which normalizes the distribution of features in the dataset by centering them around a data set with a mean of 0 and a standard deviation of 1. The Z-score scaling formula is:

$$z = \frac{x-\mu}{\sigma} \quad (1)$$

Where (z) is the Z-score or the scaled score, (x) is the score in the data to be scaled, (μ) is the mean of the data, and (σ) is deviation standard of the data.

2.2.4. Correlation-based Feature Selection

This method, known as Correlation-based Feature Selection, assesses the significance of features by evaluating the correlation level between them. It is useful for reducing data dimensionality and eliminating irrelevant or correlated features, which can improve model performance and facilitate interpretation of results

[16]. The formula for Correlation-based Feature Selection to calculate the correlation between features in equation 2:

$$Corr(Fitur1, Fitur2) = \frac{\sum_{i=1}^{n}(X_{1i}-\overline{X_1}).(X_{2i}-\overline{X_2})}{\sqrt{\sum_{i=1}^{n}(X_{1i}-\overline{X_1})^2.\sum_{i=1}^{n}((X_{2i}-\overline{X_2})^2}} \quad (2)$$

Where X_{1i} is value represents the first feature in the i-th sample, $\overline{X_1}$ is the mean value of the first feature, X_{2i} is value corresponds to the second feature in the i-th sample, $\overline{X_2}$ is the mean value of the second feature, and n is the score of samples. The purpose of this formula is to measure the level of correlation or linear relationship between two features (feature1 and feature2) in a dataset. The correlation between these features can help in selecting highly related or correlated features. The result of this formula is a correlation value between feature1 and feature2, the correlation coefficient ranges from -1 to 1, where a score of 1 represents a perfect positive linear relationship between feature1 and feature2, a value of -1 indicates a perfect negative linear relationship, and a score of 0 suggests no linear relationship between the two features. The correlation coefficient ranges from -1 to 1, the higher the correlation between the features [17].

2.3. Split Data Training and Testing

The CIC-DDoS2019 dataset, which has undergone preprocessing, is divided into two separate parts: the training data and the test data, with a ratio of 70% and 30% respectively. The training data, consisting of 70% of the total dataset, the training data is employed to train the random forest model, while the test data, comprising 30% of the entire dataset, is utilized to assess the performance of the trained random forest model on previously unseen data. By splitting the dataset, we can obtain a more realistic understanding of how well the model can perform and prevent overfitting, which is a condition where the model memorizes and predicts the trained data well but fails to generalize well on new data [18].

2.4. Random Forest

Random Forest is a robust machine learning method that utilizes ensemble techniques by merging numerous decision trees. Each decision tree algorithm is trained using a randomly selected subset of the dataset. During the classification process, the ultimate prediction is derived by collecting the majority vote from the entire set of trees. Random Forest comprises several trees, and each tree follows the same construction approach. Trees with different variables are built as independently as possible, aiming to minimize redundancy. As the dataset grows, the trees adapt and evolve accordingly, capturing more complex patterns and improving performance. The spatial separation of trees signifies their distinct development and provides a diverse range of perspectives for enhanced accuracy [19].

2.5. Evaluation Model

During the model evaluation stage, an analysis and measurement of the performance of the trained Random Forest model in identifying and detecting Distributed Denial of Service is conducted. This evaluation aims to assess the extent to which the model can recognize and classify DDoS attacks with high accuracy. Evaluation metrics, including accuracy, precision, recall, and F1-score, are employed to assess the model's performance in accurately detecting DDoS attacks. Additionally, a confusion matrix is employed to provide a visual representation of the classification results and obtain information about correctly detected attacks, missed attacks, and false predictions. These evaluation measures help assess the effectiveness and reliability of the Random Forest model in DDoS attack detection, providing insights into its strengths and areas for improvement.

Table 2. Confusion Matrix

True Class	Predict Class	
	Positive	Negative
Positive	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP)	True Negative (TN)

Confusion matrix is an evaluation method for classification models that compares predictions with the actual values to calculate evaluation metrics such as accuracy, precision, recall, and F1-score. It provides detailed information about the model's performance in correctly or incorrectly classifying data [20].

Accuracy is an evaluation metric that measures the percentage of correct classifications made by a model. The basic formula for accuracy is [20].

$$\text{Accuracy} = \frac{(TP + TN)}{(Total)} \text{x} 100\% \quad (3)$$

Precision is an evaluation metric that measures the extent to which positive predictions made by a classification model are correct. The formula for calculating precision is shown in equation 4 [20].

$$\text{Precision} = \frac{TP}{TP+FP} \text{x}100\% \quad (4)$$

Recall is an evaluation metric that measures the extent to which a classification model can find or detect the total number of true positive data. The formula for calculating recall is shown in equation 5 [20].

$$\text{Recall} = \frac{TP}{TP+FN} \text{x}100\% \quad (5)$$

The F1 score is a performance metric that balances precision and recall, providing a comprehensive measure of the classification model's effectiveness. The F1 score is calculated by combining precision and recall using equation 6 [20].

$$\text{F1 Score} = 2 \times \frac{Precession \times Recall}{Precession + Recall} \text{x}100\% \quad (6)$$

3. RESULTS AND ANALYSIS

3.1. Data Preprocessing

Before training the model, the CICDDoS2019 dataset will undergo a preparation stage to verify the accuracy and reliability of the data This preparation includes removing duplicate values, NaN, infinite, and -infinite, as well as grouping the data based on the "Label" column and filtering only the data with a count greater than 10,000 that will be used.

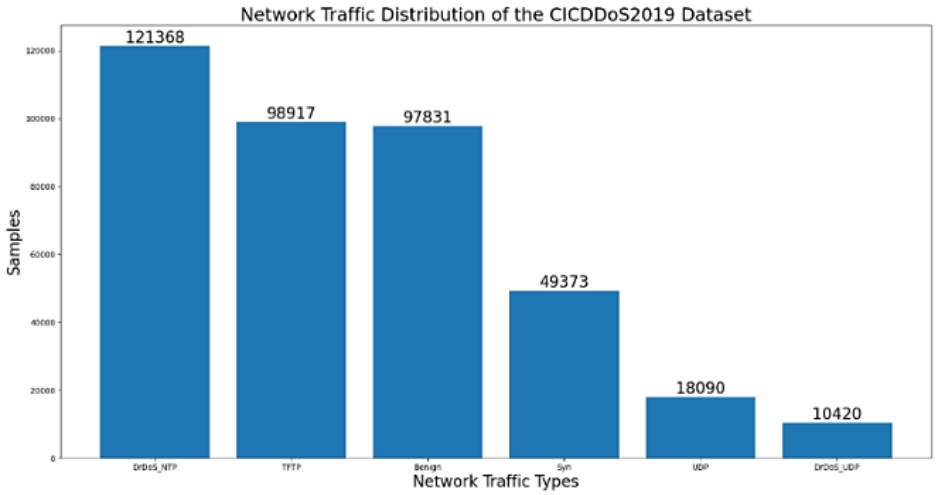

Figure 2. Result Data Cleansing

Then, the dataset undergoes a data type conversion, where columns of type int64 are converted to int32, and columns of type float64 are converted to float32. Next, data normalization is performed, which involves undersampling to address class imbalance in the dataset. After that, feature scaling is applied using the Z-Score scaling method, aiming to standardize the scale or distribution of features. The goal is to standardize the dataset, making sure that each feature has a mean of 0 and a standard deviation of 1.

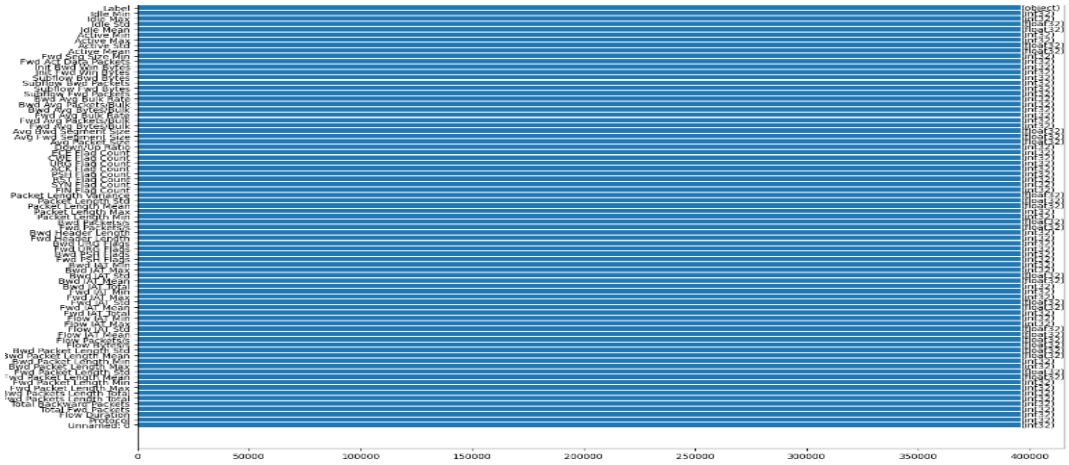

Figure 3. Result Data Tranformation and Normalization

3.2. Result Selection Feature

During process of feature selection, the correlation-based feature selection was applied to the CICDDoS2019 dataset, there are 78 features and 1 target variable 'Label'. This method calculates the correlation values between the features to be predicted using absolute values. Features that have high correlation with other features tend to have similar information and can cause multicollinearity issues in analysis or prediction models. The results of this calculation are visualized in the form of a heatmap, as shown in Figure 4.

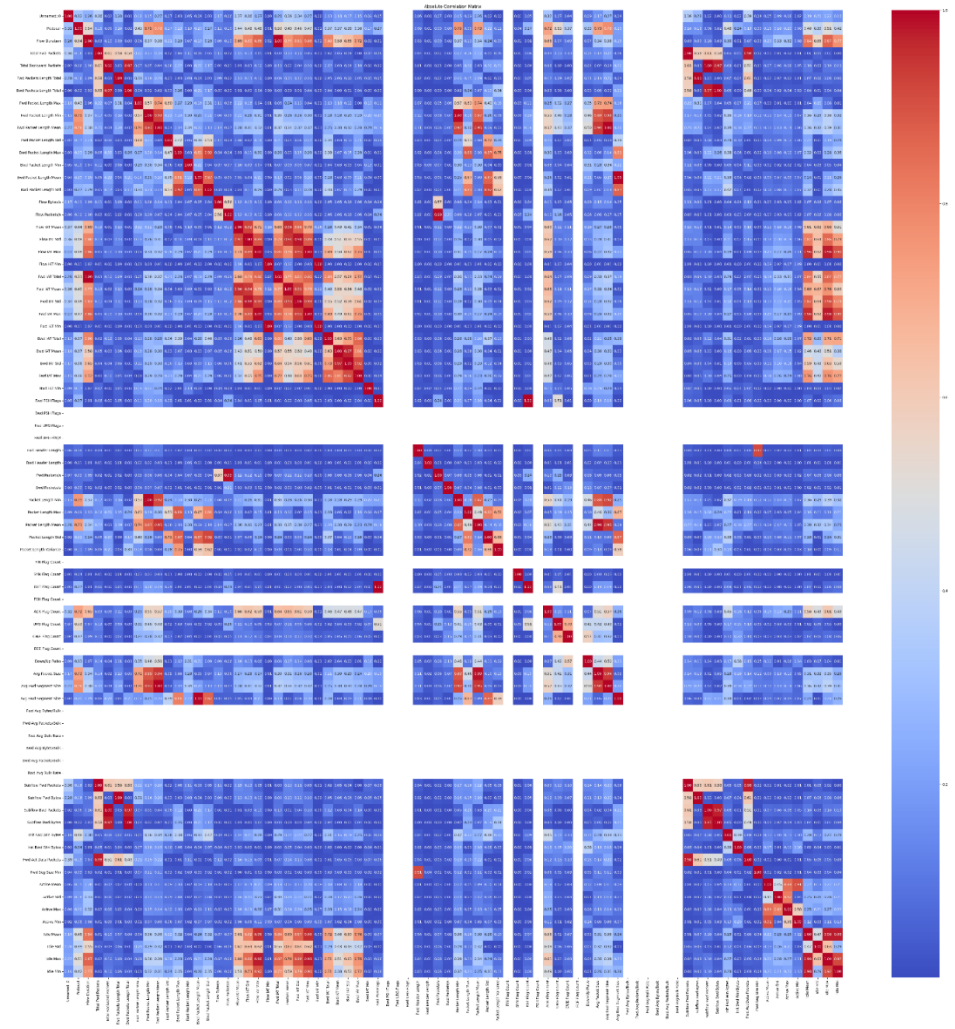

Figure 4. Heatmap Correlation Feature

In Figure 4, it can be observed that the subset of features that are colored in increasingly red shades have high correlation values. Therefore, in the correlation-based feature selection method, features that have correlations above the threshold value will be removed because their presence does not significantly contribute to improving the understanding or prediction of the target variable. On the other hand, features that have low correlation with other features are generally retained as they can provide unique and distinct information that can enhance the model's ability to classify DDoS attacks. The heatmap in Figure 3 visually represents the correlation between different features in the dataset. It indicates that highly correlated features have a strong linear relationship, which can lead to multicollinearity issues and redundant information.

3.3. Classification Testing Results

During the classification testing, the CICDDoS2019 dataset was preprocessed and split into training and testing data with a specific ratio of 70% for training and 30% for testing. To ensure reliable model evaluation, the cross-validation method with a 7-Fold Cross Validation. In this scheme, the dataset is divided into seven subsets, which are alternately used as training and testing data, helping to avoid potential biases in model performance evaluation. The algorithm employed in this model is the Random Forest algorithm, which is an ensemble model consisting of multiple decision trees trained independently and combining their predictions using voting techniques. Furthermore, this study also tests the classification results using the correlation-based feature selection method by varying the threshold parameter. I have tested several threshold values, namely 0.8, 0.6, 0.5, 0.4, 0.2, and 0.1. In this testing, features with correlations above the threshold value are eliminated because highly correlated features tend to have similar information and can lead to multicollinearity issues in analysis or prediction models. By varying the threshold value, researchers can observe its impact on the model's performance in classifying DDoS attacks. By selecting the optimal threshold value, the model's overall accuracy and performance are enhanced, leading to improved results in the classification process. The evaluation parameters include accuracy, precision, recall, and F1 Score. Furthermore, by testing various threshold values in the correlation-based feature selection method, researchers can evaluate the trade-off between the number of retained features and model performance. Choosing the appropriate threshold can result in an optimal subset of features, reducing data dimensionality, and improving the model's ability to classify DDoS attacks with high accuracy.

Table 3. Evaluation Test Results

Feature Weight	Feature Selected	Total of Trees	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
>0.8	34	10	99.83	99.83	99.83	99.83
		50	99.86	99.87	99.87	99.87
		100	99.88	99.88	99.88	99.88
		200	99.87	99.88	99.88	99.88
		500	99.87	99.87	99.87	99.87
>0.6	43	10	99.85	99.86	99.86	99.86
		50	99.87	99.87	99.87	99.87
		100	99.89	99.89	99.89	99.89
		200	99.88	99.88	99.88	99.88
		500	99.88	99.88	99.88	99.88
>0.4	52	10	99.75	99.75	99.75	99.75
		50	99.79	99.79	99.79	99.79
		100	99.79	99.79	99.79	99.79
		200	99.78	99.78	99.78	99.78
		500	99.80	99.80	99.80	99.80
>0.2	60	10	99.51	99.51	99.51	99.51
		50	99.50	99.50	99.50	99.50
		100	99.50	99.50	99.50	99.50
		200	99.52	99.52	99.52	99.52
		500	99.50	99.51	99.50	99.50
>0.1	63	10	89.35	89.39	89.36	89.36
		50	89.14	89.16	89.16	89.16
		100	89.21	89.22	89.22	89.22
		200	89.06	89.06	89.07	89.06
		500	89.09	89.09	89.10	89.09

The experiment evaluated the classification performance of DDoS attacks using various feature selection thresholds, the number of selected features, and the total decision trees. The most favorable outcome was achieved with a threshold value >0.6, 43 selected features, and 100 decision trees, resulting in an

impressive accuracy rate of 99.89%. This surpasses the performance of other parameter combinations. These findings highlight the effectiveness of the Correlation-based Feature Selection (CFS) method with a threshold >0.6, selecting 43 features, and utilizing 100 decision trees in the Random Forest algorithm for significantly improving DDoS offense detection accuracy. The study emphasizes the importance of proper feature selection and parameter optimization in enhancing the performance of DDoS attack detection systems. Further research can explore additional parameter combinations and validate the model's robustness on larger datasets to ensure the reliability of the obtained results.

4. CONCLUSION

This research provides evidence of the effectiveness of the Correlation Feature Selection (CFS) method in improving the accuracy of DDoS attack detection. In this study, by applying a threshold of 0.6, 43 relevant features were successfully selected from the dataset. When these features were combined with 100 decision trees in the Random Forest algorithm, an accuracy of 99.89% was achieved. This improvement in accuracy far surpasses the results of previous studies using the feature importance method, which only achieved an accuracy rate of 99.7%. These findings highlight the advantages of using the CFS method with a threshold of 0.6 and integration with 100 decision trees in the Random Forest algorithm for DDoS attack detection and identification.The recommendations of this study suggest the application of the CFS method with a threshold of 0.6 and the use of 100 decision trees to improve the effectiveness of DDoS attack identification and detection systems. In addition, the researchers recommend further research to optimize other relevant parameters and validate the results with larger datasets to ensure the generalizability and robustness of the model.

REFERENCES

[1] K. Nosalska and G. Mazurek, "Marketing principles for Industry 4.0 - a conceptual framework," *Eng. Manag. Prod. Serv.*, vol. 11, no. 3, pp. 9–20, 2019.

[2] Badan Siber dan Sandi Negara, "Laporan Tahunan Monitoring Keamanan Siber 2021," 2022, pp. 54–55.

[3] X. Chen, *Distributed denial of service attack and defense*, vol. 3. 2010.

[4] Amarudin, R. Ferdiana, and Widyawan, "A Systematic Literature Review of Intrusion Detection System for Network Security: Research Trends, Datasets and Methods," *ICICoS 2020 - Proceeding 4th Int. Conf. Informatics Comput. Sci.*, pp. 0–5, 2020.

[5] E. Osterweil, A. Stavrou, and L. Zhang, "20 Years of DDoS: a Call to Action," vol. 1, no. 1, pp. 1–11, 2019.

[6] M. Zamani, "Machine learning techniques for intrusion detection," *Handb. Res. Intrusion Detect. Syst.*, no. December 2013, pp. 47–65, 2020.

[7] B. Purnama, *Pengantar Machine Learning*. 2019.

[8] A. R. Wani, Q. P. Rana, U. Saxena, and N. Pandey, "Analysis and Detection of DDoS on Cloud Computing Environment using Machine Learning Techniques," *Commun. Comput. Inf. Sci.*, vol. 1076, pp. 260–273, 2019.

[9] Y. Chen, J. Hou, Q. Li, and H. Long, "DDoS attack detection based on random forest," *Proc. 2020 IEEE Int. Conf. Prog. Informatics Comput. PIC 2020*, pp. 328–334, 2020.

[10] M. Alduailij, Q. W. Khan, M. Tahir, M. Sardaraz, M. Alduailij, and F. Malik, "Machine-Learning-Based DDoS Attack Detection Using Mutual Information and Random Forest Feature Importance Method," *Symmetry (Basel).*, vol. 14, no. 6, pp. 1–15, 2022.

[11] A. M. Makkawi and A. Yousif, "Machine Learning for Cloud DDoS Attack Detection: A Systematic Review," *Proc. 2020 Int. Conf. Comput. Control. Electr. Electron. Eng. ICCCEEE 2020*, 2021.

[12] M. A. Talukder,Md Alamin; Uddin, "CIC-DDoS2019 Dataset," *Mendeley Data*, 2023. [Online]. Available: https://data.mendeley.com/datasets/ssnc74xm6r/1. [Accessed: 05-Jul-2023].

[13] F. Ridzuan and W. M. N. Wan Zainon, "A review on data cleansing methods for big data," *Procedia Comput. Sci.*, vol. 161, pp. 731–738, 2019.

[14] Á. Arnaiz-González, J. F. Díez-Pastor, J. J. Rodríguez, and C. García-Osorio, "Study of data transformation techniques for adapting single-label prototype selection algorithms to multi-label learning," *Expert Syst. Appl.*, vol. 109, pp. 114–130, 2018.

[15] E. K.I, "Data Transformation for Machine Learning," *Unversity Jean Monnet, Saint-Etienne, Fr.*, no. 4, pp. 1–8, 2018.

[16] H. A. Yanti, H. Sukoco, and S. N. Neyman, "Pemodelan Identifikasi Trafik Bittorrent Dengan Pendekatan Correlation Based Feature Selection (CFS) Menggunakan Algoritme Decision Tree (C4.5)," *CESS (Journal Comput. Eng. Syst. Sci.*, vol. 6, no. 1, p. 1, 2021.

[17] M. A. Hall, "Correlation-based Feature Selection for Machine Learning," no. April, 1999.

[18] B. Toleva, "The Proportion for Splitting Data into Training and Test Set for the Bootstrap in Classification Problems," *Bus. Syst. Res. J.*, vol. 12, 2021.

[19] C. Strobl, J. Malley, and G. Tutz, "An Introduction to Recursive Partitioning: Rationale, Application, and Characteristics of Classification and Regression Trees, Bagging, and Random Forests," *Psychol. Methods*, vol. 14, no. 4, pp. 323–348, 2009.

[20] M. D. Prasad, P. B. V, and C. Amarnath, "Machine Learning DDoS Detection Using Stochastic Gradient Boosting," *Int. J. Comput. Sci. Eng.*, vol. 7, no. 4, pp. 157–166, 2019.

BIBLIOGRAPHY OF AUTHORS

Sopian Soim is currently working as a faculty member at the State Polytechnic of Sriwijaya, where he serves as a lecturer in the Electrical Engineering Diploma 3 and Diploma 4 Study Program, specifically in the field of Telecommunication Engineering

Sholihin is presently employed as a faculty member at the State Polytechnic of Sriwijaya. He holds the position of lecturer in the Electrical Engineering Diploma 3 and Diploma 4 Study Program, specializing in Telecommunication Engineering.

Cahyo Bayu Subianto, a student in the Electrical Engineering Department, Applied Bachelor's Degree Program in Telecommunication Engineering at Sriwijaya State Polytechnic. I have a specific interest in Machine Learning, Deep Learning, and Data Science.