Critical Feature Selection Technique for Improving Performance Classification Model in Adaptive Intrusion Detection System

Anggit Ferdita Nugraha[1,*], Yoga Pristyanto[1], Beti Wulansari[1], Dian Prasetya[1]

[1]Faculty of Computer Science, Universitas Amikom Yogyakarta, Indonesia

Emails: anggitferdita@amikom.ac.id; yoga.pristyanto@amikom.ac.id; bety@amikom.ac.id; dianprasetya772@students.amikom.ac.id

Abstract

A firewall is one of the devices that supports network security, especially at the organizational level. A Firewall's effectiveness in supporting network security is highly dependent on the capabilities and abilities of the Network Administrator. Unfortunately, the high complexity of creating rules and the process of configuring Firewall rules carried out statically by the Network Administrator weakens the effectiveness of the Firewall, and it cannot adapt to increasingly dynamic network pattern changes. Machine Learning is one of the potentials that can be used so that the Firewall can work adaptively. Adaptive Firewall configuration in recognizing various attacks in the network will undoubtedly increase the effectiveness of the Firewall in ensuring network security. The success of the machine learning model performance cannot be separated from the dataset used during the learning process. The dataset used in learning often has a large dimension, but various noises and attributes are irrelevant in representing one class of data. Therefore, it is necessary to support the feature selection technique, which will show the presence of relevant characteristics in the dataset and maximize the machine learning model's performance. This study will be conducted on adding feature selection techniques to develop machine learning models on the Benchmark dataset related to network security. Various popular feature selection techniques will be evaluated, and their performance will be compared based on scenarios between feature selection techniques or scenarios that only use a single classification.

Keywords: Network Security; Firewall; Machine Learning; Feature Selection; Classification

1. Introduction

The digital era that continues to develop massively has given rise to trending issues related to computer network security [1]. In an organizational environment, computer network security needs to be a top priority and receive more attention because it can threaten data integrity, information, and user privacy [2]. According to a global cyber security agency report [3], [4], until the second quarter of 2023, more than 65% of organizations in various sectors have experienced at least one incident related to cybercrime attacks [5], [6]. There have been 3 billion phishing attacks recorded in the past year, Distributed Denial of Service (DDoS) attacks that have increased by 50%, and other cybercrime threats that cause service disruptions to organizations and even cause significant financial losses for organizations and individuals [7].

A firewall is one of the efforts used to support the security of an organization's network. A firewall protects the organization's internal network from unauthorized access and malicious attacks. The Firewall works like a defence fortress that will monitor network traffic activity and then filter the activity. Firewalls will limit and deny access to network traffic activity, which is considered a threat and cyber-attack [8].

Firewalls need to be configured with rules to recognize cybercrime threats effectively. The effectiveness of a Firewall is highly dependent on the ability of the Network Administrator to determine its configuration rules [9]. Not

Doi: https://doi.org/10.54216/JCIM.160102
Received: October 23, 2024 Revised: January 11, 2025 Accepted: February 09, 2025

infrequently, due to the high complexity in determining the proper rules, limited workforce and monitoring time that cannot be done continuously, the existence of rule configurations that are prone to errors, and the configuration process that tends to be carried out statically, the effectiveness of the Firewall is getting weaker, which results in the Firewall being unable to adapt to changes in increasingly dynamic traffic patterns.

Machine Learning is one way that can be used to increase the effectiveness of a Firewall. Machine learning has excellent potential because of its ability to learn patterns from existing data and then make predictions or decisions through the data without explicit instructions [9]. With this ability, machine learning can analyse traffic patterns and make firewall configuration rules more adaptive.

The success of a machine learning model in determining adaptive Firewall configuration rules is highly dependent on the data used during the learning process. Feature selection techniques play an essential role in machine learning modelling because they recognize noise and irrelevant features in the data [10]. Therefore, there needs to be a study that discusses the proper feature selection techniques that will improve the machine learning model's performance so that it can work well in determining adaptive firewall configuration rules, especially in enhancing network security.

This study aims to develop a computer network security system through adaptive Firewall configuration based on machine learning models. The focus of this study is to examine various Feature Selection techniques that are popularly used in creating machine learning models. The study was conducted by comparing feature selection techniques with popular machine learning models to determine the performance improvement of the resulting model. The machine learning model with the best performance will be used to develop an adaptive Firewall configuration system to improve computer network security

2. Related Work

Related research shows great potential in using machine learning to improve computer network security. Research is dominated by machine learning [5]–[7], [9], [11] or deep learning [8], [9], [12] modelled on various datasets related to cyber security. Research [12] developed a machine-learning model for intrusion detection. The Support Vector Machine (SVM) algorithm used in modelling using NSL-KDD data showed a high accuracy value of 93.95%.

Research [13] also conducted machine learning modelling based on the SVM algorithm. Experiments conducted on three datasets, namely NSL-KDD, UNSW_NB15, and CICIDS2017, showed that the performance of SVM is worthy of consideration as an effective classification model. This is indicated by the accuracy value produced of 93.75% on the UNSW_NB15 dataset, 98.92% on the CICIDS2017 dataset, and 99.35% on the NSL-KDD dataset.

Other studies [14] also developed machine learning models for intrusion detection using various variations of popular classification algorithms such as Artificial Neural Network (ANN) [14]–[16], Decision Tree (DT) [4], [17]–[19], K-Nearest Neighbour (KNN) [17], [19], Naive Bayes (NB) [7], [14], [20], Random Forest (RF) [7], [15], [17], [18], [21], and Convolutional Neural Network (CNN) [9], [14]. Through experiments conducted on the UNSW-NB15, CICIDS2017, and NSL-KDD datasets, overall, they could show good performance based on the accuracy values produced. Research [14] also stated that the study could be used as a benchmark for subsequent research.

Most of the studies that examined the development of machine learning models on cyber security datasets were conducted based on a single classification model without considering the characteristics and features of the dataset. On the other hand, the Feature Selection method is known to improve model performance in various cases [16], [20]. This raises research opportunities to explore further the potential for improving the performance of machine learning models if feature selection techniques are added to the modelling process.

3. Materials and Methods

In general, this research was conducted using a machine learning model development methodology that was carried out sequentially, starting from the data acquisition stage, preprocessing, and Feature Engineering, which includes the Feature Extraction and Feature Selection stages, then continued with the modelling stage, and finally, the evaluation as shown in Figure 1.

Doi: https://doi.org/10.54216/JCIM.160102
Received: October 23, 2024 Revised: January 11, 2025 Accepted: February 09, 2025

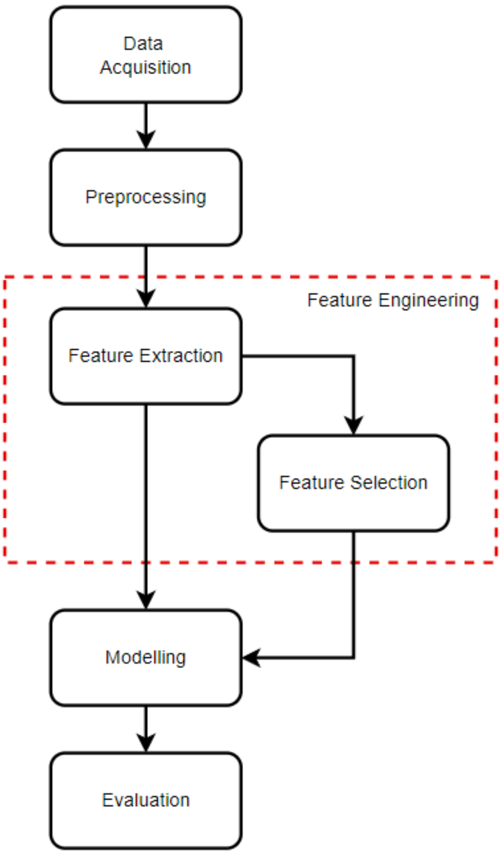

Figure 1. Research Flow

Figure 1. Shows the research methodology is carried out in sequential stages, starting from the data acquisition stage, the preprocessing stage, the Feature Engineering process, and the data modelling stage. Finally, the Evaluation process.

3.1. Data Acquisition

It is the first stage carried out which aims to collect datasets from various sources. There are three benchmark datasets (UNSW_NB-15, CICIDS 2017, and NSL-KDD) which will be the main datasets used in the study, but it does not rule out the possibility of datasets that are also obtained from other sources or primary data. The data that has been collected will then be reviewed and then continued to the preprocessing stage.

3.2. Preprocessing

Preprocessing is used to prepare the acquired dataset so that it is ready to be used in the subsequent processing stage. Some of the processes carried out in this preprocessing stage include handling missing data, handling data errors and outliers, and ensuring that no data is duplicate.

3.3. Feature Extraction

In the Feature Extraction process, the study scenario will be carried out according to the benchmark dataset, where all attributes in the dataset that have gone through the preprocessing stage will be used in the modelling process.

3.4. Feature Selection

Meanwhile, in the Feature Selection process, various popular feature selection techniques will be studied to determine relevant attributes that will improve the machine learning model's performance.

The feature engineering stage contributes to this research. The feature selection process, which is the focus, will be carried out in an experimental form using various popular feature selection techniques that will produce the best performance.

Doi: https://doi.org/10.54216/JCIM.160102
Received: October 23, 2024 Revised: January 11, 2025 Accepted: February 09, 2025

A. Information Gain

Information Gain is a widely used feature selection technique for improving machine learning model performance across various tasks [11]. It is popular due to its ability to identify the most relevant attributes based on their significance to the dataset as a whole [16]. The feature selection process using Information Gain involves measuring the change in entropy before and after the data is split. This process consists of three main steps: first, calculating the Information Gain value for each attribute in the dataset; second, determining a threshold to decide whether to retain or discard an attribute; and third, updating the dataset by removing attributes with low Information Gain values [22].

Mathematically, the entropy before data splitting is calculated using Equation (1):

$$Entropy\ (S) = -\sum_{i}^{k} p_i log_2\, p_i \quad (1)$$

Where S represents the entropy value before splitting, denotes the proportion of samples in class i, and k is the number of target classes. Next, after the data is split, the entropy of the attribute is recalculated using Equation (2):

$$Entropy\ (S, A) = \sum_{v \in A} \frac{|S_v|}{|S|} x\ Entropy\ (S_v) \quad (2)$$

$|S_v|$ represents the number of samples with v value for attribute A, while S de notes the total number of samples in the dataset. Once both entropy values (before and after splitting) are calculated, the next step is to determine the Information Gain $IG\ (S, A)$ by computing the difference between the entropy before splitting $Entropy\ (S)$ and the entropy after splitting $Entropy\ (S, A)$. This calculation is performed using Equation (3):

$$IG\ (S, A) = Entropy\ (S) - Entropy\ (S, A) \quad (3)$$

B. Gain Ratio

Gain Ratio is a feature selection technique developed to address the limitations of Information Gain, which tends to exhibit bias towards attributes with many unique values (high-cardinality). This bias arises because Information Gain naturally assigns higher preference to attributes with a greater number of distinct values, even if those attributes may not be entirely relevant [11]. Definitively, Gain Ratio is the ratio of Information Gain to the intrinsic information of the data. By using Gain Ratio, the method evaluates the space of high-dimensional features in relation to the target class of the dataset, providing a more balanced and intrinsic assessment of attribute relevance.

The general formula for calculating Gain Ratio is expressed as Equation (4):

$$Gain\ Ratio\ (S, A) = \frac{InformationGain\ (S,A)}{SplitInformation(S,A)} \quad (4)$$

$Gain\ Ratio\ (S, A)$ is the value calculated by normalizing the Information Gain (3) with the $SplitInformation(S, A)$, which represents the intrinsic value of an attribute that partitions the class information within the dataset. This relationship can be expressed mathematically as (5):

$$SplitInformation\ (S, A) = -\sum_{i}^{v} (\frac{|S_v|}{|S|} xlog_2 \frac{|S_v|}{|S|}) \quad (5)$$

Where:

$|S_v|$ is the number of samples with a specific value v for attribute A.
$|S|$ is the total number of samples in the dataset.

The value produced by Gain Ratio ranges from 0 to 1. A value closer to 1 indicates a strong relationship between the attribute and the class data, signifying that the attribute provides significant information for class differentiation. Conversely, a value closer to 0 suggests a weak or insignificant relationship, implying that the attribute contributes little to distinguishing the class data. This range allows Gain Ratio to effectively assess the relevance of attributes in feature selection.

C. Correlation Based Feature

In this study, a correlation-based feature selection approach is conducted using Pearson Correlation. Pearson Correlation is chosen as it is one of the most popular feature selection techniques, capable of identifying relationships and dependencies between attributes. Through this approach, the correlation between dataset attributes and the target class is measured to identify attributes that strongly influence the class. The Pearson Correlation values range from -1

Doi: https://doi.org/10.54216/JCIM.160102
Received: October 23, 2024 Revised: January 11, 2025 Accepted: February 09, 2025

to 1, where values approaching -1 indicate a strong negative correlation, values approaching 1 indicate a strong positive correlation, and values close to 0 suggest no significant correlation between the attributes and the target class [23].

The correlation between attributes is calculated using Pearson Correlation, as shown in Equation (6):

$$C = \frac{\sum_{i=1}^{n}(a_i-\bar{a})(b_i-\bar{b})}{\sqrt{\sum_{i=1}^{n}(a_i-\bar{a})^2 \sqrt{\sum_{i=1}^{n}(b_i-\bar{b})^2}}} \tag{6}$$

Here:

- C is the value of Pearson Correlation Coefficient.
- n is the sample size
- a_i and b_i represent the individual values of features a and b, respectively,
- A and b are the mean values of features a and, respectively.

The Pearson Correlation coefficient C obtained from Equation (6) falls within the range of -1 to 1, indicating the strength and direction of the correlation between the attributes.

D. Chi-Square

Chi-Square (Chi^2) is a feature selection technique that applies a statistical approach to measure the difference in distribution between attributes [23]. Chi2 evaluates the deviation in distribution under the assumption that the attribute in the data is independent of the target class values. This statistical method is widely used to assess the relevance of categorical attributes in relation to the class labels.

The general formula for calculating the Chi-square value from a dataset is given in Equation (7):

$$Chi2\ (a_i, y_j) = \frac{N(TZ-YX)^2}{(T+X)(T+Z)+(X+Z)+(Y+Z)} \tag{7}$$

In Equation (7), the term T represents the frequency of feature a_i occurring together with class label y_j in the dataset. The value X indicates the frequency of a_i occurring without the presence of y_j, while Y denotes the frequency of y_j occurring without a_i. The term Z corresponds to the frequency of neither a_i nor y_j appearing in the dataset. Lastly, N is he total number of records in the dataset, serving as a normalization factor to scale the Chi-square value relative to the dataset size. Together, these components allow the Chi-square formula to evaluate the strength of association between an attribute and the class label by comparing observed and expected frequencies under the assumption of independence.

E. Recursive Feature Elimination (RFE)

Recursive Feature Elimination (RFE) is a feature selection algorithm that adopts a wrapper approach [16]. RFE requires a base classifier to identify the dominant and relevant attributes within a dataset. It operates iteratively using a greedy algorithm approach to progressively reduce the data's dimensionality and produce a subset of attributes that yields optimal classification performance [11].

The RFE process is generally processed through the following steps:

$\boldsymbol{Input}: Dataset\ (X) = \{a_1, a_2, a_3, \ldots, a_n\}$

Process:

1. Define a base classifier f to be used (e.g., Decision Tree, Support Vector Machine, etc.).
2. Train the model f using all attributes a in the dataset X with the target class y.
3. Evaluate the importance of each feature by using (8)

$$I(a) = score(f(X, y)) \tag{8}$$

4. Eliminate the feature with the lowest importance score.

$$X \leftarrow X \backslash \left\{ \underset{a_i \in X}{\operatorname{argmin}}\, I(a) \right\} \tag{9}$$

5. Repeat steps 2–4 until all attributes have been processed, resulting in a ranked list of features R as a subset with the best performance.

$\boldsymbol{Output}: Feature\ Rank\ (R) = \{r_{a_1}, r_{a_2}, r_{a_3}, \ldots, r_{a_n}\}$

This iterative approach ensures that the final subset of features selected by RFE contains the most relevant attributes for the given classification task, maximizing the model's performance.

3.5. Modelling

The modelling stage is carried out after the feature extraction and feature selection stage. At this stage, data that has passed the feature engineering stage will be modelled using a popular classification model. In general, this study employs popular classification algorithms such as Decision Tree, Logistic Regression, Support Vector Machine, Naive Bayes, and Random Forest to evaluate model performance based on single classification tasks prior to the feature extraction process.

The model with the best performance will be used as the base classifier to identify the relevant features from the feature selection process that will be evaluated. Additionally, experiments are conducted using an AutoML approach to ensure that the model built with a single classifier can serve as a reliable benchmark. This benchmark will be used to compare and analyze the model's performance and impact after undergoing the feature selection process.

3.6. Evaluation

The evaluation stage is a stage that aims to evaluate the learning model created during the experimental process. In this study, accuracy (10) is emphasized as the primary evaluation metric to assess model performance. However, since the focus and contribution of the research lie in a comparative study on the application of feature selection methods, additional metrics are utilized to measure model performance, number of selected features as dominant or relevant features (n) and also processing time to computing training process.

$$accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative} \quad (10)$$

4. Result and Discussion

This study aims to conduct a comparative analysis of popular feature selection methods to identify the most effective approach for detecting network attacks. The experiments in this research are conducted using three benchmark datasets: NSL-KDD, UNSW-NB15, and CICIDS2017. These datasets are selected to ensure that the findings can be generalized, enabling their application in the development of various cyber-attack detection systems.

The following are some key findings from the study:

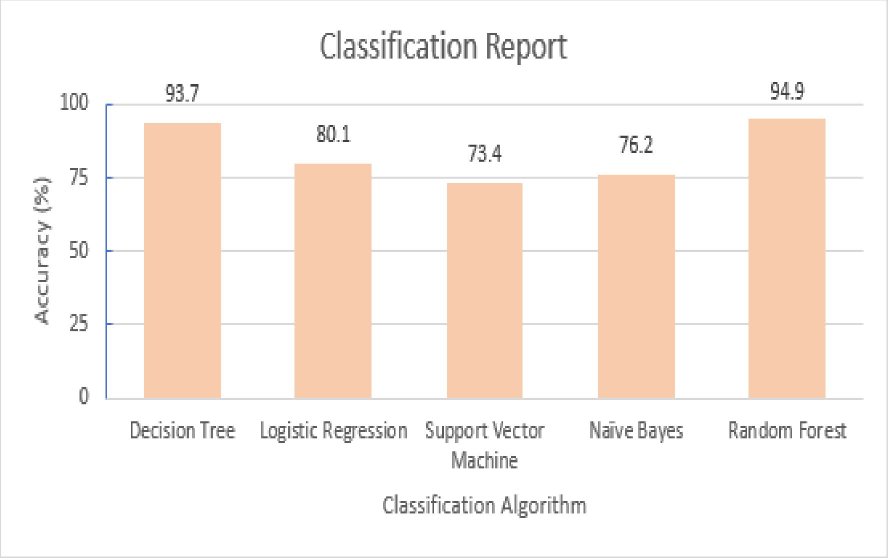

Figure 2. Comparison of single classification schema for UNSW-NB15 dataset.

Figure 2 shows the result of a comparison of the accuracy values of five (5) popular algorithms performed on the UNSW_NB-15 dataset. As a result, the Random Forest algorithm has the highest value when compared to other popular classification algorithms, with an accuracy value of 94.9%, followed by the Decision Tree with 93.7%, and then sequentially Logistic Regression with 80.1%, Naive Bayes 76.2%, and Support Vector Machine 73.4%.

Doi: https://doi.org/10.54216/JCIM.160102
Received: October 23, 2024 Revised: January 11, 2025 Accepted: February 09, 2025

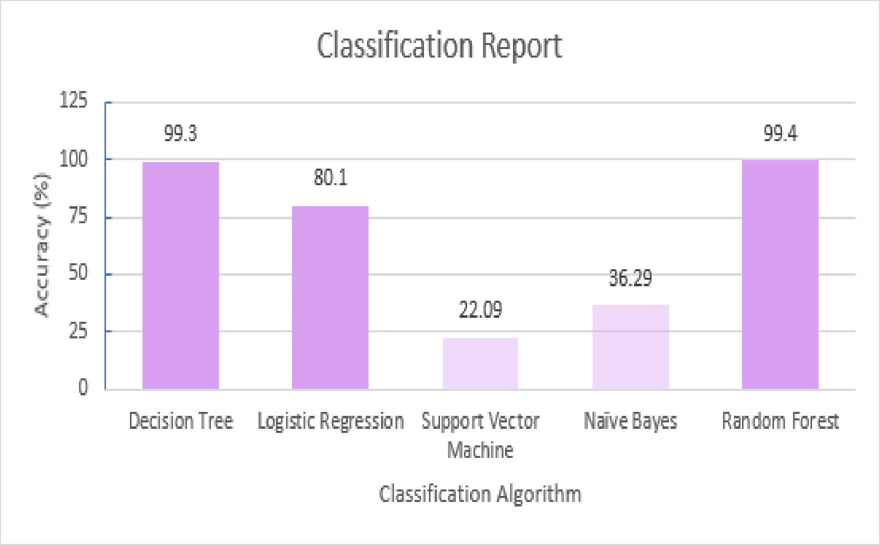

Figure 3. Comparison of single classification schema for NSL-KDD dataset.

Meanwhile, in Figure 3, where the first scenario experiment is applied to the NSL-KDD dataset, the resulting accuracy value has an unbalanced performance on the single classification algorithm tested in this study. However, when viewed in order, the Random Forest algorithm has the highest accuracy value of 99.4%, followed by the Decision Tree with a value of 99.3%, and the third place is Logistic Regression with 80.1%. Unfortunately, in this experiment, using the support vector machine and naive Bayes algorithms produced poor performance, namely 22.09% for SVM and 36.29 for using Naive Bayes.

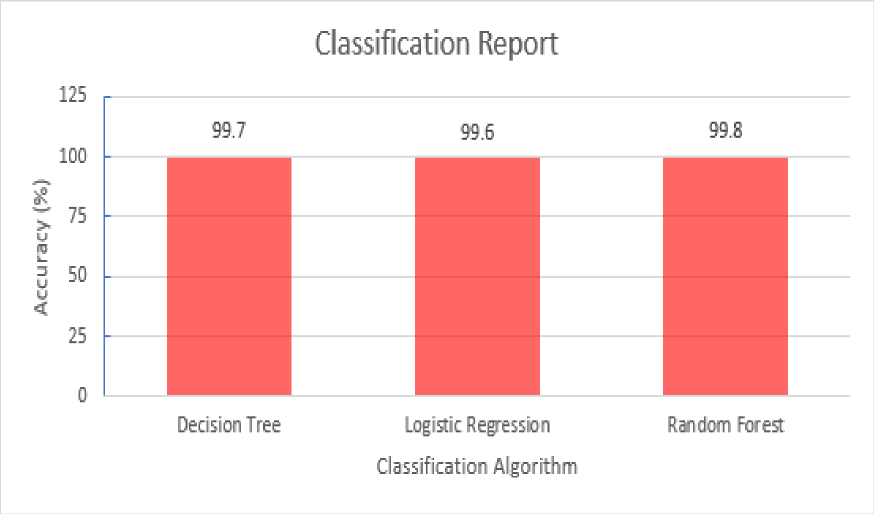

Figure 4. Comparison of single classification schema for CICIDS2017 dataset.

Similar to Figure 3, Figure 4 shows a performance comparison of using a single popular classification algorithm tested on the CICIDS 2017 dataset. Figure 4 shows the use of three popular classification algorithms, namely Decision Tree, Logistic Regression, and Random Forest, as algorithms used to evaluate the CICIDS 2017 dataset. This is because the large dimensions of the CICIDS 2017 dataset prevent the process of using other popular algorithms from producing performance values. Use on the CICIDS2017 dataset produces successive accuracy values of 99.8%, 99.7%, and 99.6%, where the first place uses Random Forest as its classifier, the second place is Decision Tree, and the third place is the use of the Logistic Regression algorithm.

Doi: https://doi.org/10.54216/JCIM.160102
Received: October 23, 2024 Revised: January 11, 2025 Accepted: February 09, 2025

Based on experiments conducted on the three datasets used in the study, Random Forest consistently ranks first regarding the accuracy values produced compared to other popular algorithms. Therefore, in the second scenario, Random Forest will be the base classifier to determine the model performance if a feature selection process is carried out to determine the relevant features in each dataset in recognizing attacks on computer networks.

Table 1: Comparison of model performance on the UNSW_NB-15 dataset

Method	***n- feature***	***Accuracy***	***Time Processing (ms)***
Single Classification (Scenario 1)	43	94.9	40.22
Information Gain	25	93.7	61.43
Gain Ratio	**38**	**95.11**	53.1
Chi2	26	94.91	40.99
Correlation Based	16	94.81	**29.8**
Recursive Feature Elimination (RFE)	**39**	**95.12**	50.05

Table 1 compares the accuracy values of using a single classification algorithm (Random Forest), as in scenario 1, with the addition of popular feature selection techniques to find the best performance with relevant features. As a result, using RFE-based wrapper feature selection shows a higher accuracy value than a single classification algorithm, which is 95.12%. The same happens with the Gain Ratio, which produces an accuracy value of 95.11%. From both results, it can be seen that the UNSW_NB-15 dataset has 38 to 39 relevant features.

Table 2: Comparison of model performance on the NSL-KDD dataset

Method	***n-feature***	***Accuracy***	***Time Processing (ms)***
Single Classification (Scenario 1)	42	99.4	**3.02**
Information Gain	26	**99.5**	12.6
Gain Ratio	15	93.0	4.51
Chi2	16	98.9	11.16
Correlation Based	18	98.4	10.45
Recursive Feature Elimination (RFE)	34	95.12	50.05

Doi: https://doi.org/10.54216/JCIM.160102
Received: October 23, 2024 Revised: January 11, 2025 Accepted: February 09, 2025

Table 2 is the result of the experiment on the NSL-KDD dataset. Information gain is at the top in accuracy but not far behind using a single classification with a difference of 0.1% when viewed from the accuracy value produced. However, when viewed from the processing time, using a single classification shows a better value than the high processing time using Information Gain.

Table 3: Comparison of model performance on the CICIDS2017 dataset

Method	***n-feature***	***Accuracy***	***Time Processing (ms)***
Single Classification (Scenario 1)	78	99.80	31.88
Information Gain	26	99.99	32.12
Gain Ratio	15	**99.98**	**8.78**
Chi2	16	**99.95**	**7.53**
Correlation Based	18	99.93	28.48
Recursive Feature Elimination (RFE)	34	99.99	22.19

Table 3 compares accuracy from the experiments conducted by using the CICIDS 2017 dataset. The accuracy appears balanced, with a significant difference of 0.1%. However, when viewed from the time processing side, the use of chi2 and gain ratio shows potential use.

5. Conclusion

This study aims to develop a computer network security system through adaptive Firewall configuration based on machine learning models. The focus of this study is to examine various Feature Selection techniques that are popularly used in creating machine learning models. Experiments in this study were conducted on three benchmark datasets, namely UNSW_NB-15, NSL-KDD, and UNSW, five single classification algorithms, namely Decision Tree, Logistic Regression, Support Vector Machine, Naive Bayes, and Random Forest, and popular feature selection techniques including Information Gain, Gain Ratio, Chi2, Correlation Based, and Recursive Feature Elimination (RFE). As a result, the use of RFE-based wrapper feature selection showed a higher accuracy value than a single classification algorithm, which was 95.12% on the UNSW_NB-15 dataset. On the NSL-KDD dataset, information gain was at the top in terms of accuracy, but it was not far behind using a single classification with a difference of 0.1%. However, when viewed from the processing time side, using a single classification shows a better value than the high processing time using Information Gain. In the CICIDS dataset, the accuracy values appear quite balanced; however, when viewed from the processing time side, the use of chi2 and gain ratio shows potential use

Funding the Directorate General of Higher Education, Research and Technology – Indonesia, number 107/E5/PG.02.00.PL/2024 June 1, 2024, Ministry of Education, Culture, Research, and Technology – Indonesia, funded this study.

Conflicts of Interest: “The authors declare no conflict of interest.”

References

[1] H. Liu and B. Lang, “Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey,” *Applied Sciences*, vol. 9, no. 20, 2019.

[2] A. Pinto, L. C. Herrera, Y. Donoso, and J. A. Gutierrez, “Survey on Intrusion Detection Systems Based on Machine Learning Techniques for the Protection of Critical Infrastructure,” *Sensors*, vol. 23, no. 5, pp. 1–18, 2023.

Doi: https://doi.org/10.54216/JCIM.160102
Received: October 23, 2024 Revised: January 11, 2025 Accepted: February 09, 2025

[3] B. B. Gupta, K. Yadav, I. Razzak, K. Psannis, A. Castiglione, and X. Chang, "A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment," *Computer Communications*, vol. 175, no. April, pp. 47–57, 2021.

[4] Z. Azam, M. M. Islam, and M. N. Huda, "Comparative Analysis of Intrusion Detection Systems and Machine Learning-Based Model Analysis through Decision Tree," *IEEE Access*, vol. 11, no. August, pp. 80348–80391, 2023.

[5] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, 2019.

[6] Q. A. Al-Haija and A. Ishtaiwi, "Machine Learning Based Model to Identify Firewall Decisions to Improve Cyber-Defense," *International Journal of Advanced Science, Engineering and Information Technology*, vol. 11, no. 4, pp. 1688–1695, 2021.

[7] F. Abdou Vadhil, M. Lemine Salihi, and M. Farouk Nanne, "Machine learning-based intrusion detection system for detecting web attacks," *IAES International Journal of Artificial Intelligence*, vol. 13, no. 1, p. 711, 2024.

[8] A. Khraisat and A. Alazab, "A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges," *Cybersecurity*, vol. 4, no. 1, 2021.

[9] H. Zainel and C. Koçak, "LAN Intrusion Detection Using Convolutional Neural Networks," *Applied Sciences*, vol. 12, no. 13, 2022.

[10] H. Ahmetoglu and R. Das, "A comprehensive review on detection of cyber-attacks: Data sets, methods, challenges, and future research directions," *Internet of Things (Netherlands)*, vol. 20, no. May 2022.

[11] S. S. Issa, S. Q. Salih, Y. D. Salman, and F. H. Taha, "An Efficient Hybrid Filter-Wrapper Feature Selection Approach for Network Intrusion Detection System," *International Journal of Intelligent Engineering Systems*, vol. 16, no. 6, pp. 261–273, 2023.

[12] J. Lu, L. Tan, and H. Jiang, "Review on convolutional neural network (CNN) applied to plant leaf disease classification," *Agriculture*, vol. 11, no. 8, pp. 1–18, 2021.

[13] A. H. A and K. Sundarakantham, "Machine Learning Based Intrusion," in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, 2019, pp. 916–920.

[14] Z. K. Maseer, R. Yusof, N. Bahaman, S. A. Mostafa, and C. F. M. Foozy, "Benchmarking of Machine Learning for Anomaly Based Intrusion Detection Systems in the CICIDS2017 Dataset," *IEEE Access*, vol. 9, pp. 22351–22370, 2021.

[15] B. A. Tama and S. Lim, "Ensemble learning for intrusion detection systems: A systematic mapping study and cross-benchmark evaluation," *Computer Science Review*, vol. 39, p. 100357, 2021.

[16] W. L. Al-Yaseen, A. K. Idrees, and F. H. Almasoudy, "Wrapper feature selection method based on differential evolution and extreme learning machine for intrusion detection system," *Pattern Recognition*, vol. 132, p. 108912, 2022.

[17] H. Y. Alshaeaa and Z. M. Ghadhban, "Developing a hybrid feature selection method to detect botnet attacks in IoT devices," *Kuwait Journal of Science*, vol. 51, no. 3, p. 100222, 2024.

[18] X. Gao, C. Shan, C. Hu, Z. Niu, and Z. Liu, "An Adaptive Ensemble Machine Learning Model for Intrusion Detection," *IEEE Access*, vol. 7, pp. 82512–82521, 2019.

[19] A. A. Alsulami, Q. Abu Al-Haija, A. Tayeb, and A. Alqahtani, "An Intrusion Detection and Classification System for IoT Traffic with Improved Data Engineering," *Applied Sciences*, vol. 12, no. 23, 2022.

[20] K. C. Khor, C. Y. Ting, and S. P. Amnuaisuk, "A feature selection approach for network intrusion detection," in *Proceedings of the 2009 International Conference on Information Management and Engineering (ICIME)*, 2009, pp. 133–137.

[21] T. Saranya, S. Sridevi, C. Deisy, T. D. Chung, and M. K. A. A. Khan, "Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: A Review," *Procedia Computer Science*, vol. 171, no. 2019, pp. 1251–1260, 2020.

[22] C. Kalimuthan and J. Arokia Renjit, "Review on intrusion detection using feature selection with machine learning techniques," *Materials Today: Proceedings*, vol. 33, pp. 3794–3802, 2020.

[23] A. K. Dey, G. P. Gupta, and S. P. Sahu, "Hybrid Meta-Heuristic based Feature Selection Mechanism for Cyber-Attack Detection in IoT-enabled Networks," *Procedia Computer Science*, vol. 218, pp. 318–327, 2022.

Doi: https://doi.org/10.54216/JCIM.160102
Received: October 23, 2024 Revised: January 11, 2025 Accepted: February 09, 2025