Indonesian Journal of Electrical Engineering and Informatics (IJEEI)
Vol. 12, No. 2, June 2024, pp. 384~396
ISSN: 2089-3272, DOI: 10.52549/ijeei.v12i2.5543

Classification of Darknet Traffic Using the AdaBoost Classifier Method

Rizky Elinda Sari[1], Deris Stiawan[2], Nurul Afifah[3], Mohd. Yazid Idris[4], Rahmat Budiarto[5]
[1,2,3]Faculty of Computer Science, Universitas Sriwijaya, Indralaya – Ogan Ilir 30662, Indonesia
[4]Faculty of Computing, Universiti Teknologi Malaysia, Johor Baru, Malaysia
[5]College of Computing and Information, Al-Baha University, Alaqiq, Saudi Arabia

Article Info

Article history:

Received Apr 19, 2024
Revised May 10, 2024
Accepted May 20, 2024

Keywords:

Darknet
Classification
AdaBoost
N_estimator
SMOTE-ENN

ABSTRACT

Darknet is famous for its ability to provide anonymity which is often used for illegal activities. A security monitor report from BSSN highlights that 290.556 credential data from institution in Indonesia have been exposed on the darknet. Classification techniques are important for studying and identifying darknet traffic. This study proposes the utilization of the AdaBoost Classifier in darknet classification. The use of variable estimator values significantly impact classification results. The best performance was obtained with an estimator value of 500 with an accuracy of 99.70%. The contribution of this research lies in the development of classification models and the evaluation of AdaBoost models in the context of darknet traffic classification.

Corresponding Author:

Deris Stiawan,
Faculty of Computer Science,
Universitas Sriwijaya,
Jalan Palembang – Prabumulih Km. 32 Indralaya – Ogan Ilir 30662, Indonesia.
Email: deris@unsri.ac.id

1. INTRODUCTION

The Internet provides the platform for an information system known as the World Wide Web (WWW). Internet users have access to shared information which is then widely exchanged throughout te world [1]. Sometimes internet users are unaware of the existence of hidden layers within the internet that provide anonymity and act as the primary platform for various illegal activities and cybercrimes. This layer is known as the Darknet [2], [3]. The darknet is a subset of deep web that cannot be accessed by regular search engines [4]. Darknet is an encrypted network technology that provides anonymity to senders and receivers [5], [6]. Darknet access to the darknet can only be done with anonymity technologies such as Virtual Private Network (VPN) or The Onion Router (Tor) [7], [8]. The darknet is a place where it is most likely thob e found stolen credential files of data sold freely [9]. The existence of crypto currency or bitcoin facilitates transactions on the darknet [10]. In addition to data, within the darknet can be found malware from unknown source, such as keyloggers, botnets, and ransomware [11].

Darknet have the potential to cause darknet exposure. Darknet exposure is a condition when data or account credential information at a particular institution is exposed on the darknet. Based on the report of the results of the BSSN cybersecurity monitor in August 2023 [12], 290,556 exposure data were found from 431 affected agencies in Indonesia. Classification can be considered the first step in addressing darknet problems as a basis for studying and differentiating darknet traffic. Evidenced by the success of research by Laskhari et al [13] who classified darknet on the CIC-Darknet2020 dataset with a CNN approach. As a result, the CNN approach was able to distinguish darknet and benign traffic with 86% accuracy. Based on the research, we propose a different approach using AdaBoosr Classifier. The goal is to improve the accuracy results of previous research.

AdaBoost Classifier is one of the currently popular boosting ensemble learning methods [14]. AdaBoost Classifier or Adaptive Boosting produces stronger learning by adding repeated weak learners [15], [16]. This method enhance model performance by adjusting the wights of incorrect observations in each iteration. Javed et al [17] demonstrated the effectiveness of this approach in detecting botnet attacks, achieving an accuracy of 99.1%.

In summary, the principal contributions of our research are detailed as follows :

- We employ a hybrid sampling technique, specifically SMOTE-ENN, to effectively address the data imbalances present in the CIC-Darknet2020 dataset. This approach enhances the robustness ande reliability of our model.
- We implement the AdaBoost Classifier to accurately differentiate between darknet and benign traffic.
- We provide a comprehensive evaluation of the AdaBoost model performance by varying the number of estimator (n_estimator). This thorough assessment highlights the impact of estimator variation on model accuracy.
- We analyze the optimal performance of AdaBoost based on the application different learning rates. This analysis identifies the most effective learning rate for achieving superior classification results, thereby enhancing the models predictive capability.

The remainder of the paper is organized as follows, section II describes the proposed method. Section III provides an analysis results and section IV concludes the contents of the paper.

2. METHOD

This research method describe the stages of the research process describe in a framework. The framework systematically describes the stages of research including dataset preparation, pre-processing and classification process using AdaBoost and validation of results.

2.1. Research Framework

The framework in this study aims to nake the research process more sequential and more structured. The first stage is a literature study to obtain information related to the research to be carried out. The second stage is to prepare the dataset by understanding the features of the dataset. At this stage the dasa is normalize, and balanced the dataset if needed. The last stage is the classification process using AdaBoost. From the classification results, analysis is carried out and then validated using confusion matrix calculations. Validation is performed to calculate accuracy, precision, recall, specificity, F1-score, MCC and BACC. The framework of this study can be seen in Figure 1.

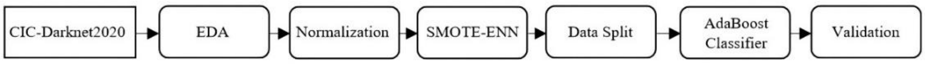


Figure 1. Research Stages

2.2. Dataset

The dataset used in this study is CIC-Darknet2020 which comes from the Canadian Institute for Cybersecurity [13]. The CIC-Darknet2020 dataset has 83 features with 2 label indicating classes. The dataset consists of 141,530 samples. Each is grouped by the type application used to generate traffic. This sub category of apps includes Audio-Streaming, Browsing, Chat, Email, File Transfer P2P Video-Streaming and VOIP. The samples are divided into classes that are benign and darknet. 92,926 samples were benign classes and 24,098 samples were darknet classes. The comparison of the number of benign and darknet classes in the dataset can be seen in Figure 2.

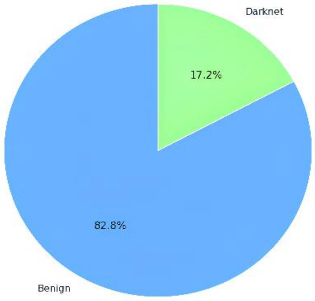


Figure 2. Class Distribution

2.3. Exploratory Data Analysis

Exploratory Data Analysis is the initial process carried out to understand and analyse datasets before the classification process is performed [18]. This stage carries out an initial review of the data, using visualization and descriptive statistics to identify pattern, relationship and clean the data from duplication. EDA is needed to find out data that is not relevant to research and find potential outliers to prepare data before conducting further analysis or model creation.

2.4. Data Encoding

In machine learning, data encoding is used to convert categorical variables into numerical representations [19]. Because data encoding converts data into numerical data, this technique is especially helpful when working with machine learning algorithms that can only process numerical data. There are many methods of data encoding, one of which is label encoding. This study applied label encoding to convert categorical data into numerical representations starting with 0 (zero).

2.5. Normalization

Before normalization or standardization, it is necessary to calculate the skewness value of each feature in the dataset [20]. This skewness value describes the extent to which the distribution of data tends to be skewed to one side or the other. Skewness can be positive, negative, or zero. A dataset allows varying skewness values. To overcome the variation in values from skewness, normalization becomes a better option than standardization. Data normalization aims to change the values of variables in a set to be on the same scale. In this study, robust scaler were used as a normalization method to reduce outlier values so that the normalization process was more controlled. Robust scaler was chosen because this method is more resistant to outliers compared to Min-Max scaler. Robust scaler seek to reduce the outlier effect by centering data around the median and scaling it according to the range between quartiles [21]. The calculation of the robust scaler is done with equation (1) with Q1 being the first quartile and Q3 being the third quartile.

$$X_{scale} = \frac{x_i - Q_1(x)}{Q_3(x) - Q_1(x)} \tag{1}$$

2.6. SMOTE-ENN

The CIC-Darknet2020 dataset is a class imbalance dataset. Class imbalance can occur when data is not evenly distributed and the number of minority classes is smaller then the majority class [22]. To overcome imbalance in the dataset, oversampling or undersampling techniques can be used [23]. The combination of oversampling and undersampling techniques is called hybrid sampling [24]. This study applies hybrid sampling in overcoming data imbalances. The hybrid sampling method implemented in this study is SMOTE-ENN.

Synthetic Minority Over-Sampling Technique Combine With Edited Nearest Neighbor (SMOTE-ENN) is a data pre-processing method that combines oversampling and undersampling techniques [25]. SMOTE is applied to create a new majority sample. Furthermore, ENN is applied to clean data from samples that are considered noise [26]. SMOTE-ENN has the advantage of reducing the potential for overfitting caused by synthetic sample generation. SMOTE-ENN can produce cleaner datasets by removing ambiguous samples after SMOTE implementation and preventing models from propensity to learn noise. Based on these advantages, SMOTE-ENN is considered more effective in improving the performance of classification models on unbalanced data than ordinary SMOTE.

The wat SMOTE-ENN works is to combine SMOTE and ENN. In the SMOTE stage, synthetic samples from minority classes are created by selecting random points from minority classes and new synthetic points between whose points and some of their closest neighbors. The goal is to add variety into minority classe. After that the points just crated from SMOTE are most likely misclassified as noise or outliers so they neet to be removed from the dataset. The process of removing such noise data is carried out using ENN by identifying points that have a majority of neighbours from the majority class and removing them. Based on this, SMOTE-ENN is a hybrid approach that combiner oversampling techniques (SMOTE) to deal with class imbalance and cleans samples that are considered noise with undersampling techniques (ENN). The pseudocode of SMOTE-ENN is as follows :

SMOTE-ENN Pseudocode

```
Import SMOTE-ENN from library imbalanced-leaarn.
Initialization of  SMOTE-ENN using parameter sampling-strategy=’auto’ and random_state=42.
Resample dataset using fit_resample from SMOTE-ENN
Save the resampled dataset into a Xresample variable and y_resample
```

2.7. Classification Using AdaBoost

AdaBoost Classifier is one of the ensemble learning algorithms used in machine learning [27]. AdaBoost generates strong learning by adding weak learning iteratively [15]. The advantage of AdaBoost is the ability of model to focus on samples that were difficult to classify by previous models. AdaBoost gives greater weight to samples that are misclassified, so the next iteration will focus on improving predictions for those samples. This process allows AdaBoost to produce models that have better accuracy than the relatively weak base model. The architecture of AdaBoost is as follows :

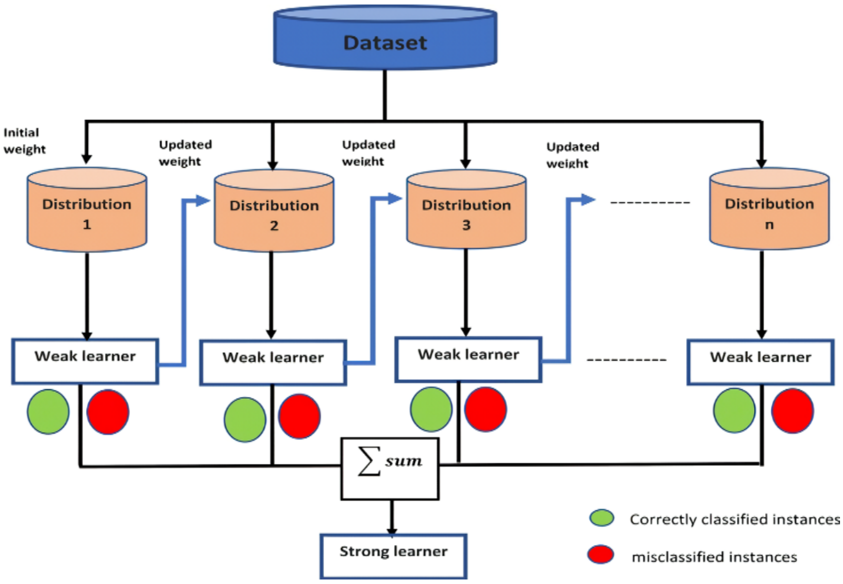


Figure 3. AdaBoost Architecture

AdaBoost has several parameters that can be used to improve model performance. In this study, apply some hyperparameter adjustments from AdaBoost. The following table shows the hyperparameter tuning applied in this study :

Table 1. Hyperparameter Settings

Parameter	Value
Number of Estimator (n_estimator)	100, 200, 300, 400, 500.
Base Learner	Decision Tree Classifier, with max depth = 1.
Algorithm	SAMME.R
Learning Rate	0.00001, 0.0001, 0.001, 0.01, 0.1
Random State	42

The pseudocode of the AdaBoost Classifier is as follows :

Pseudocode AdaBoost Classifier

```
1 : Import AdaBoostClassifier from sklearn_ensemble library.
2 : Initialize the base learner model, for example the Decision Tree Classifier.
3 : Initialize the AdaBoost model with base estimator, n_estimator, algorithms, learning rate, and random state.
4 : Train an AdaBoost model using training data (X_train, y_train).
5 : Test the model using test data (X_test).
6 : Displays confusion matrix from data testing results.
```

2.8. Validation of Results

After conducting the classification process, validation of classification results is required to measure the extent to which the model perform in identifying target classes. In this study, the validation method used was the Confusion Matrix. The confusion matrix was applied to evaluate the performance of classification models on datasets [28]. Confusion matrix is a matrix table that contains four main categories, namely True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) [29], [30]. In this study, a number of performances were measured. Performance that can be measured includes Accuracy, Precision, Recall, Specificity, F1-Score, Matthews Correlation Coefficient (MCC), and Balanced Accuracy (BACC) [31]–[34].

Table 2. Model Performance Measurement

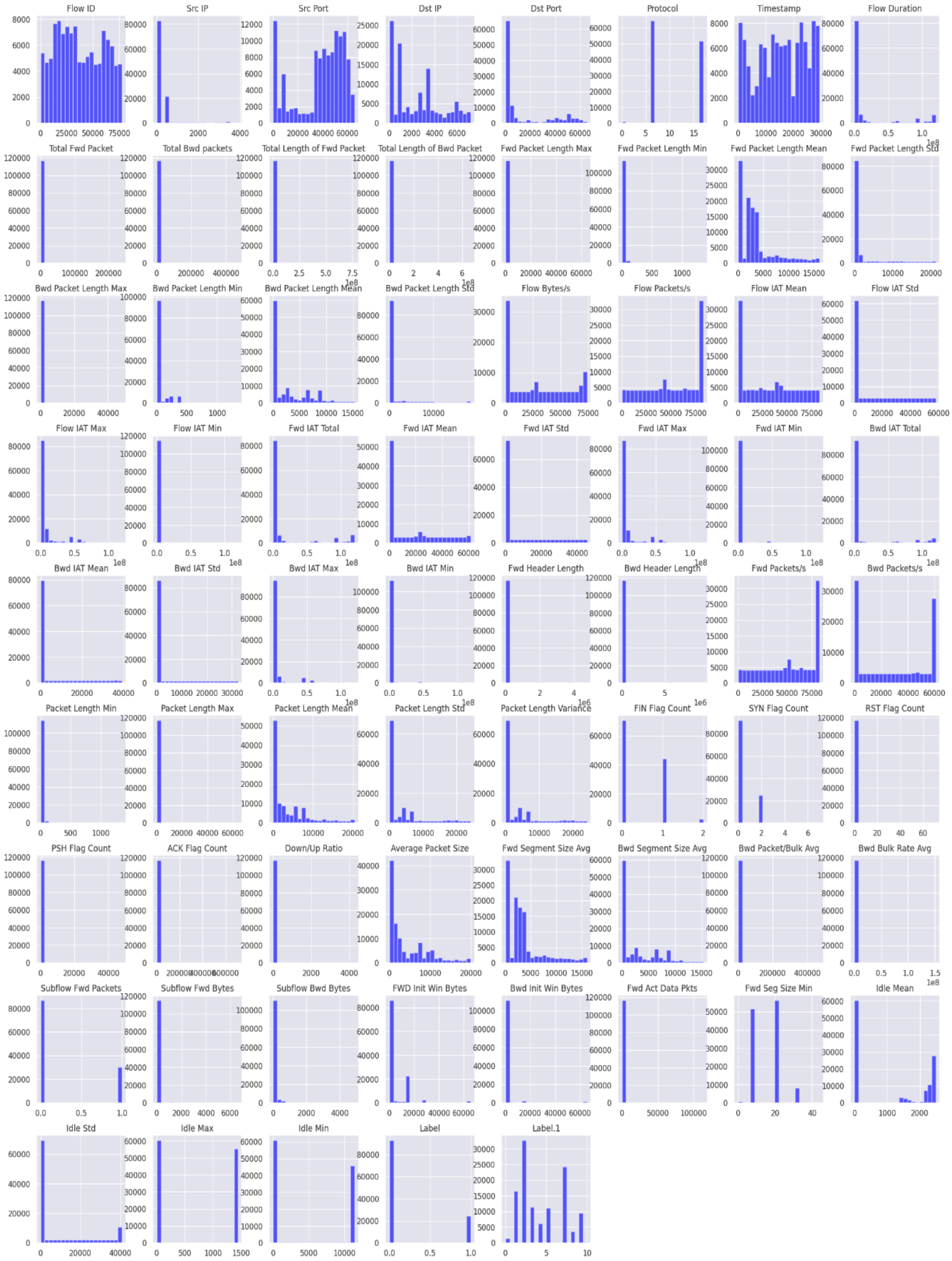
Performance	Formula	Details
Accuracy	TP + TN / TP + TN + FP + FN	Accuracy is used to measure how well the model can correctly identify classes.
Precision	TP / TP + FP	Precision is used to measure how accurately the model classifies positive samples.
Recall (Sensitivity)	TP / TP + FN	Recall is used to measure how accurately the model classifies a positive or a true positive class.
Specificity	TN / TN + FP	Specificity is used to measure how well the model identifies a negative class or true negative class.
F1-Score	2 * (Precision * Recall) / (Precision + Recall)	F1-Score is used to measure the balance between precision and recall.
MCC	(TP * TN – FP * FN) / √(TP + FN) * (TP + FN) * (TN + FP) * (TN + FN)	MCC provides information on how well the model predicts positive and negative classes.
BACC	(Sensitivity + Specificity) / 2	BACC provides a measure of how well a model performs in an unbalanced class.

3. RESULTS AND DISCUSSION

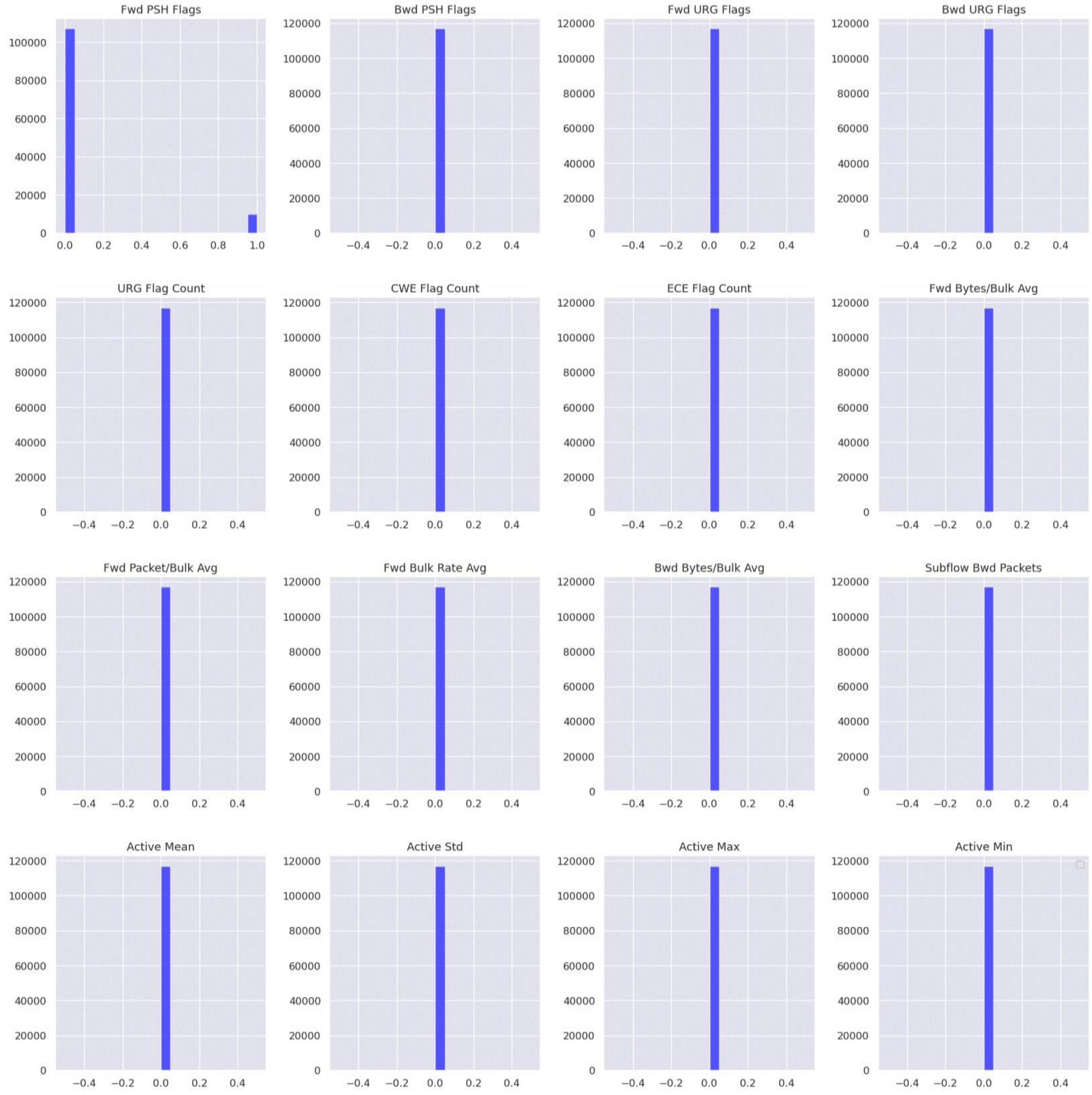
In this section, describes the results of darknet traffic classification by implementing the steps described in the previous section. Starting from dataset preparation to classification using AdaBoost. The classification process is done using the Python programming language and using the scikit-learn library. After the classification results are obtained, validation of the results is carried out with a confusion matrix to calculate the performance of the AdaBoost model.

3.1. Exploratory Data Analysis

Before classification, datasets need to be cleaned of duplicate data. The goal is to make the dataset cleaner. In thie CIC-Darknet2020 dataset there are 24,457 duplicate data, the data needs to be removed from the dataset. Before being cleared of duplication data, benign samples totaled 117,219 and darknet samples totaled 24,331. After the dataset was cleared of duplicate data, the number of benign samples owed to 92,926 and darknet samples owed to 24,098. After the data is cleared of duplication, the histogram visualization is then carried out. The goal is to find out the features that have high variability and low variability. Furthermore, features with low variability can be removed from the dataset in an attempt to improve model performance. The histogram visualization of the CIC-Darknet dataset can be seen in Figure 4.



(a). High variability features



(b). Low variability features

Figure 4. Histogram EDA

The results of histogram visualization based on figure 4 show feature variability. Section (a) shows the distribution of data that tends to be evenly distributed along the range of values. This shows that these features have high variation. Features with high variability provude relevant information in data analysis. Next in section (b) shows the distribution of data that is on the zero line or has no value. This suggests that these features hav low variability. Features with low variability tend to not to provide significant information in data analysis. Therefore, features with low variability may be considered for removal from the dataset.

3.2. SMOTE-ENN

The next process is to balance the unbalance dataset. The process of balancing datasets is indispensable. Without class balancing on the dataset, the model tends to have a significant bias towards minority classes, this cause the model to perform poorly in identifying minority classes. To balance the dataset, this study applied SMOTE-ENN. SMOTE oversamples the dataset. SMOTE takes excess samples from the majority data to produce synthetic samples. This led to an increase in the number of samples in minority classes. As a result, the minority and majority classes are evenly matched.

Once the data becomes balanced, an ENN is applied. ENN aims to remove samples that are perceived as noise from already balanced data. The result of the application of ENN is that the number of samples in the majority class is reduced, so that the number of minority samples is more than the majority. Despite the difference in numbers, the dataset becomes more balanced because the difference in the number of samples become closer compared to the number of samples before SMOTE-ENN was implemented. The difference in data before and after SMOTE-ENN is applied can be seen in Figure 5 and Figure 6.

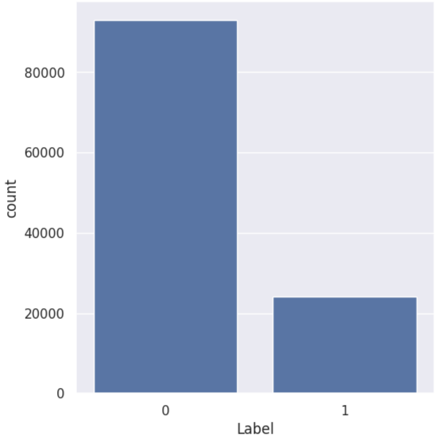


Figure 5. Amount of Data Before SMOTE-EN

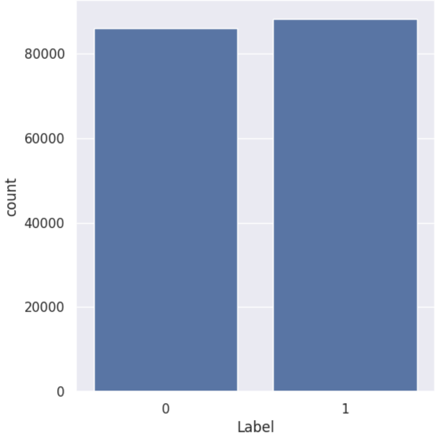


Figure 6. Amount of Data After SMOTE-ENN

3.3. Classification Using AdaBoost

In this study, a Decision Tree Classifier with a maximum depth of 1 from each tree was used as a base learner model. SAMME.R algorithm is used in the boosting process by initializing the random state value setting of 42. The n_estimator value used vary from 100 to 500. The goal is to measure the extent to which variation in the value of the estimator affect the performance of the model. The model is build with split data divisions of 50:50, 60:40, 70:30, 80:20, and 90:10. So there will be 5 scenarios for validating the results of each data split.

3.4. Result Validation

The confusion matrix is an important instrument in validating the results of darknet traffic classification. This matrix presents true positive, true negative, false positive and false negative value. This matrix is used for calculation such as Accuracy, Precision, Recall, Specificity, F1-Score, MCC, and BACC. The calculation is carried out based on 5 scenarios applied to the classification process. Next, in the sixth scenario, the best performance for each data split is tested using different learning rates.

3.4.1 Results of The First Scenario

The first scenario is classified with a 50:50 data split. The results listed in Table 3 show the model was able to achieve the highest accuracy of 99.68%. The precision value can reach 99.69% and the recall value can reach 99.52%. Furthermore, the specificity of the model was able to reach the highest value of 99.85% and the f1-score reached 99.69%. MCC and BACC values show an increase as n_estimator increase. MCC and BACC were able to achieve the highest value of 0.99.

Table 3. Result of Scenario 1

				Data Split 50:50			
n_estimator	Accuracy	Precision	Recall	Specificity	F1-Score	MCC	BACC
100	97.34%	99.56%	95.18%	99.56%	97.32%	0.94	0.97
200	99.00%	99.82%	98.18%	99.82%	99.00%	0.98	0.98
300	99.36%	99.84%	98.91%	99.83%	99.36%	0.98	0.99
400	99.66%	99.84%	99.48%	99.83%	99.66%	0.99	0.99
500	99.68%	99.69%	99.52%	99.84%	99.69%	0.99	0.99

3.4.2 Results of The Second Scenario

The second scenario is classified with 60:40 data split. The results listed in Table 4 show that the table is able to achieve the highest accuracy of 99.66%. The precision value on the model can reach 99.84% and the recall value can reach 99.48%. In addition, the specificity of the model reached a value of 99.84% and its f1-score reached a value of 99.66%. MCC and BACC in the second scenario were able to reach the highest value of 0.99.

Table 4. Results of Scenario 2

Data Split 60:40							
n_estimator	Accuracy	Precision	Recall	Specificity	F1-Score	MCC	BACC
100	97.18%	99.45%	94.94%	99.48%	97.15%	0.94	0.97
200	98.68%	97.80%	97.59%	99.80%	98.86%	0.97	0.98
300	99.35%	99.83%	98.88%	99.83%	99.35%	0.98	0.99
400	99.64%	99.83%	99.46%	99.83%	99.64%	0.99	0.99
500	99.66%	99.84%	99.48%	99.84%	99.66%	0.99	0.99

3.4.3 Result of The Third Scenario

The third scenario is classified with 70:30 data split. The results listed in Table 5 show that the model was able to achieve the highest accuracy of 99.68%. the model managed to achieve a precision value of 99.84% and recall of 99.52%. Model specificity was able to reach value of 99.85%, while f1-score reached a value of 99.68%. MCC and BACC in the third scenario were able to reach the highest value of 0.99.

Table 5. Result of Scenario 3

Data Split 70:30							
n_estimator	Accuracy	Precision	Recall	Specificity	F1-Score	MCC	BACC
100	96.48%	99.53%	93.50%	99.54%	96.42%	0.o3	0.96
200	98.78%	99.78%	97.81%	99.77%	98.78%	0.97	0.96
300	99.46%	99.80%	99.13%	99.79%	99.47%	0.98	0.99
400	99.60%	99.83%	99.38%	99.83%	99.61%	0.99	0.99
500	99.68%	99.84%	99.53%	99.85%	99.68%	0.99	0.99

3.4.4 Results of the Fourth Scenario

The fourth scenario is classified with 80:20 data split. The results listed in Table 6 show that the model was able to achieve accuracy of up to 99.70%. The precision value of the model was able to reach 99.87% and recall reached 99.53%. The specificity of the model was able to reach value of 99.87% and the f1-score of the model reached 99.70%. The MCC and BACC values in the fourth scenario managed to reach 0.99.

Table 6. Result of Scenario 4

Data Split 80:20							
n_estimator	Accuracy	Precision	Recall	Specificity	F1-Score	MCC	BACC
100	96.64%	99.49%	99.83%	99.51%	96.58%	0.93	0.97
200	96.64%	99.49%	93.83%	99.51%	96.58%	0.93	0.97
300	99.46%	99.81%	99.11%	99.81%	99.46%	0.98	0.99
400	99.62%	99.83%	99.42%	99.83%	99.62%	0.99	0.99
500	99.70%	99.87%	99.53%	99.87%	99.70%	0.99	0.99

3.4.5 Results of The Fifth Scenario

The fifth scenario is classified with 90:10 data split. The results listed in Table 7 show the model was able to achieve an accuracy of 99.66%. The precision and recall values reached 99.84% and 99.48% respectively. The specificity value of the model was able to reach 99.84% and the f1-score value reached 99.66%. The MCC and BACC values in the fifth scenario were able to reach a value of 0.99.

Table 7. Result of Scenario 5

Data Split 90:10							
n_estimator	Accuracy	Precision	Recall	Specificity	F1-Score	MCC	BACC
100	96.39%	99.45%	93.36%	99.48%	96.31%	0.92	0.96
200	98.85%	99.74%	97.96%	99.75%	98.84%	0.97	0.98
300	99.36%	99.76%	99.97%	99.76%	99.36%	0.98	0.99
400	99.59%	99.82%	99.36%	99.82%	99.59%	0.99	0.99
500	99.68%	99.86%	99.49%	99.86%	99.68%	0.99	0.99

3.4.6 Results of The Sixth Scenario

In the sixth scenario, we tested the best performance of each data split using different learning rates. This test aims to find out when the model achieves the best performance based on the learning rate applied. The results of these tests are presented in Tabel 8.

Table 8. Results of Scenario 6

Data Split	Learning rate	Accuracy	Precision	Recall	Specificity	F1-score	MCC	BACC
50 : 50	0.00001	81.1%	73.56%	97.84%	63.91%	83.98%	0.66	0.81
	0.0001	81.1%	73.56%	97.84%	63.91%	83.98%	0.66	0.81
	0.001	81.1%	73.56%	97.84%	63.91%	83.98%	0.66	0.66
	0.01	94.87%	98.35%	91.41%	98.43%	94.75%	0.9	0.95
	0.1	99.68%	99.86%	99.52%	99.85%	99.68%	0.99	0.99
60 : 40	0.00001	81%	73.48%	97.81%	63.81%	83.91%	0.66	0.81
	0.0001	81%	73.48%	97.81%	63.81%	83.91%	0.66	0.81
	0.001	81%	73.48%	97.81%	63.81%	83.91%	0.66	0.81
	0.01	94.8%	98.29%	91.32%	98.37%	94.68%	0.9	0.95
	0.1	99.66%	99.84%	99.48%	99.84%	99.66%	0.99	0.99
70 : 30	0.00001	81.13%	73.61%	97.84%	63.98%	84%	0.66	0.81
	0.0001	81.13%	73.61%	97.84%	63.98%	84%	0.66	0.81
	0.001	81.13%	73.61%	97.84%	63.98%	84%	0.66	0.66
	0.01	94.84%	98.28%	91.43%	98.36%	94.73%	0.9	0.95
	0.1	99.68%	99.84%	99.52%	99.84%	99.68%	0.99	0.99
80 : 20	0.00001	81%	73.43%	97.84%	63.84%	83.9%	0.66	0.81
	0.0001	81%	73.43%	97.84%	63.84%	83.9%	0.66	0.81
	0.001	81%	73.43%	97.84%	63.84%	83.9%	0.66	0.81
	0.01	94.99%	98.39%	91.59%	98.47%	94.87%	0.9	0.95
	0.1	99.7%	99.87%	99.53%	99.87%	99.7%	0.99	0.99
90 : 10	0.00001	81.22%	73.61%	97.75%	64.45%	83.98%	0.66	0.81
	0.0001	81.22%	73.61%	97.75%	64.45%	83.98%	0.66	0.81
	0.001	81.22%	73.61%	97.75%	64.45%	83.98%	0.66	0.81
	0.01	94.91%	98.54%	91.25%	98.63%	94.76%	0.9	0.95
	0.1	99.68%	99.86%	99.49%	99.86%	99.68%	0.99	0.99

The test results show that the learning rate has a significant influence on model performance. At very small learning rate values (0.00001 to 0.001), model performance tends to be consistent but less optimal. With an accuracy around 81%. However, by increasing the learning rate to 0.01, there was a significant jump in performance, with accuracy increasing to around 94%. The best performance was achieved at a learning rate of 0.1, where the model achieves highest accuracy 99.7% with 80:20 data split.

3.5. Discussion

From the five models that have been evaluated, it can be seen the AdaBoost model shows good performance in darknet traffic classification. There is consistently an increase in model performance along with an increase in n_estimator value. The best model is determined by the n_estimator and data division values that result in the highest accuracy, precision, recall, and f1-score. In this study, the model with 80:20 data sharing with a value of n_estimator 500 became the best model because it achieved the highest accuracy of 99.70% with 99.87% precision, 99.53% recall, and 99.70% f1-score.

Dividing the data by an 80:20 rasio provides a sufficient size of data testing to test the model and has large enough data to train the model. This helps the model to learn well from the data and make better predictions. The use of n_estimator of 500 provides considerable complexity in capturing complex pattern from data, allowing the model to learn better and produce better performance.

The use of SMOTE-ENN also has an impact on model performance. SMOTE-ENN helps the model to learn better from minority classes and reduces the risk of overfitting that may arise from synthetic sample making. SMOTE-ENN provides higher accuracy results compared to regular SMOTE because SMOTE-ENN removes ambiguous samples or irrelevant noise from the dataset.

The application of the learning rate also proves how the model performs. In this research we applu a learning rate from 0.00001 to 0.1. The best performance is obtained when the learning rate is 0.1. Increasing the learning rate allows the model to update weights more aggresively, which can speed up convergence and capture more complex pattern in the data.

With AdaBoost accuracy reaching 99.7%, this research shows higher accuracy compared to other approach in previous research. Comparison of the results with previous research can be seen in Table 9.

Table 9. Research Comparison

Research	Methods	Accuracy
Our Research	**AdaBoost**	**99.7%**
Al-Haija et al (2022)	BAG-DT	99.5%
Almomani (2022)	Ensemble Stacking	96.74%
Demertzis et al (2021)	WANN	92.7%
Habibi Laskhari et al (2020)	CNN	86%

4. CONCLUSION

Based on the result of the study, the highest accuracy rate obtained reached 99.70% while the lowest accuracy rate was 96.48%. This highest accuracy value is obtained when the data is divided by rasio of 80:20, n_estimator 500 and learning rate 0.1. This indicates that the model was able to provide the best results in these conditions. The AdaBoost model managed to achieve a value of 0.99 for MCC. This MCC value close to 1 indicates that the model has excellent capabilities and has a good balance between specificity and sensitivity. The model also achieved a value of 0.99 for BACC. This shows that the AdaBoost model was able to provide balanced and accurate prediction even then the sample in the dataset is unbalanced.

Varying n_estimator usage affect model performance. The grater the n_estimator value, the better the performance of the model. It is shown that the value of accuracy, precision, recall, specificity, f1-score, MCC, and BACC is increasing as the n_estimator value increases. Therefore, choosing the right n_estimator value is key in maximizing AdaBoost performance. Overall, AdaBoost is able to classify darknet traffic well. This AdaBoost model can be optimized for classification use in other cases.

For further research developement, there are several suggestions that can be used as a reference. Future research could implement other methods of classifying darknet traffic. To address the dataset imbalance, future research may apply techniques other than SMOTE-ENN. Further research can perform multiclass classification in distinguishing the types of application used in darknet traffic.

REFERENCES

[1] T. Pare, “Darknet and Black Market Activities Against the Cybersecurity : a Survey,” no. April, pp. 1–13, 2019, [Online]. Available: https://www.researchgate.net/publication/334126403_Darknet_and_black_market_activities_against_the_cybersecurity_A_Survey

[2] A. Anju *et al.*, “A Mysterious And Darkside Of The Darknet : A Qualitative Study,” *Webology*, vol. 18, no. 4, pp. 285–294, 2021.

[3] A. *Almomani, Darknet traffic analysis, and classification system based on modified stacking ensemble learning algorithms*, no. 0123456789. Springer Berlin Heidelberg, 2023. doi: 10.1007/s10257-023-00626-2.

[4] K. Demertzis, K. Tsiknas, D. Takezis, C. Skianis, and L. Iliadis, “Darknet traffic big-data analysis and network management for real-time automating of the malicious intent detection process by a weight agnostic neural networks framework,” *Electron.*, vol. 10, no. 7, 2021, doi: 10.3390/electronics10070781.

[5] M. Coutinho Marim *et al.*, “Darknet traffic detection and characterization with models based on decision trees and neural networks,” *Intell. Syst. with Appl.*, vol. 18, no. December 2022, p. 200199, 2023, doi: 10.1016/j.iswa.2023.200199.

[6] M. Alimoradi, M. Zabihimayvan, A. Daliri, R. Sledzik, and R. Sadeghi, “Deep Neural Classification of Darknet Traffic,” *Front. Artif. Intell. Appl.*, vol. 356, no. October, pp. 105–114, 2022, doi: 10.3233/FAIA220323.

[7] H. Mohanty, A. H. Roudsari, and A. H. Lashkari, “Robust stacking ensemble model for darknet traffic classification under adversarial settings,” *Comput. Secur.*, vol. 120, p. 102830, 2022, doi: 10.1016/j.cose.2022.102830.

[8] X. Tong, C. Zhang, J. Wang, Z. Zhao, and Z. Liu, “Dark-Forest: Analysis on the Behavior of DarkWeb Traffic via DeepForest and PSO Algorithm,” *C. - Comput. Model. Eng. Sci.*, vol. 135, no. 1, pp. 561–581, 2023, doi: 10.32604/cmes.2022.022495.

[9] Q. A. Al-Haija, M. Krichen, and W. A. Elhaija, “Machine-Learning-Based Darknet Traffic Detection System for IoT Applications,” *Electron.*, vol. 11, no. 4, 2022, doi: 10.3390/electronics11040556.

[10] K. Kanemura, K. Toyoda, and T. Ohtsuki, “Identification of Darknet Markets’ Bitcoin Addresses by Voting Per-Address Classification Results,” *ICBC 2019 - IEEE Int. Conf. Blockchain Cryptocurrency*, pp. 154–158, 2019, doi: 10.1109/BLOC.2019.8751391.

[11] F. Horasan and A. H. Yurttakal, “Darknet Web Traffic Classification via Gradient Boosting Algorithm,” *Int. J.*

Eng. Res. Dev., vol. 14, no. 2, pp. 794–798, 2022.

[12] BSSN, “Laporan Bulanan Agustus 2023,” 2023, [Online]. Available: www.idsirtii.or.id

[13] A. H. Lashkari, “DIDarknet : A Contemporary Approach to Detect and Characterize the Darknet Traffic using Deep Image Learning,” pp. 1–13.

[14] Y. Zhang, J. Liu, and W. Shen, “A Review of Ensemble Learning Algorithms Used in Remote Sensing Applications,” *Appl. Sci.*, vol. 12, no. 17, 2022, doi: 10.3390/app12178654.

[15] B. Thilagavathi, K. Suthendran, and K. Srujanraju, “Evaluating the AdaBoost Algorithm for Biometric-Based Face Recognition,” *Lect. Notes Data Eng. Commun. Technol.*, vol. 63, pp. 669–678, 2021, doi: 10.1007/978-981-16-0081-4_67.

[16] T. Toharudin *et al.*, “Boosting Algorithm to Handle Unbalanced Classification of PM2.5Concentration Levels by Observing Meteorological Parameters in Jakarta-Indonesia Using AdaBoost, XGBoost, CatBoost, and LightGBM,” *IEEE Access*, vol. 11, no. March, pp. 35680–35696, 2023, doi: 10.1109/ACCESS.2023.3265019.

[17] A. Rehman Javed, Z. Jalil, S. Atif Moqurrab, S. Abbas, and X. Liu, “Ensemble Adaboost classifier for accurate and fast detection of botnet attacks in connected vehicles,” *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 10, pp. 1–18, 2022, doi: 10.1002/ett.4088.

[18] P. Chakri, S. Pratap, Lakshay, and S. K. Gouda, “An exploratory data analysis approach for analyzing financial accounting data using machine learning,” *Decis. Anal. J.*, vol. 7, no. March, p. 100212, 2023, doi: 10.1016/j.dajour.2023.100212.

[19] M. X. Low *et al.*, “Comparison of Label Encoding and Evidence Counting for Malware Classification,” *J. Syst. Manag. Sci.*, vol. 12, no. 6, pp. 17–30, 2022, doi: 10.33168/JSMS.2022.0602.

[20] L. Huang, J. Qin, Y. Zhou, F. Zhu, L. Liu, and L. Shao, “Normalization Techniques in Training DNNs: Methodology, Analysis and Application,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 8, pp. 10173–10196, 2023, doi: 10.1109/TPAMI.2023.3250241.

[21] L. B. V. de Amorim, G. D. C. Cavalcanti, and R. M. O. Cruz, “The choice of scaling technique matters for classification performance,” *Appl. Soft Comput.*, vol. 133, pp. 1–37, 2023, doi: 10.1016/j.asoc.2022.109924.

[22] C. Kaope and Y. Pristyanto, “The Effect of Class Imbalance Handling on Datasets Toward Classification Algorithm Performance,” *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 22, no. 2, pp. 227–238, 2023, doi: 10.30812/matrik.v22i2.2515.

[23] M. S. Kraiem, F. Sánchez-Hernández, and M. N. Moreno-García, “Selecting the suitable resampling strategy for imbalanced data classification regarding dataset properties. An approach based on association models,” *Appl. Sci.*, vol. 11, no. 18, 2021, doi: 10.3390/app11188546.

[24] M. Alamri and M. Ykhlef, “Hybrid Undersampling and Oversampling for Handling Imbalanced Credit Card Data,” *IEEE Access*, vol. 12, no. November 2023, pp. 14050–14060, 2024, doi: 10.1109/ACCESS.2024.3357091.

[25] Y. Zhu, Y. Hu, Q. Liu, H. Liu, C. Ma, and J. Yin, “A Hybrid Approach for Predicting Corporate Financial Risk: Integrating SMOTE-ENN and NGBoost,” *IEEE Access*, vol. 11, no. October, pp. 111106–111125, 2023, doi: 10.1109/ACCESS.2023.3323198.

[26] F. Yang, K. Wang, L. Sun, M. Zhai, J. Song, and H. Wang, “A hybrid sampling algorithm combining synthetic minority over-sampling technique and edited nearest neighbor for missed abortion diagnosis,” *BMC Med. Inform. Decis. Mak.*, vol. 22, no. 1, pp. 1–14, 2022, doi: 10.1186/s12911-022-02075-2.

[27] J. M. Ferreira *et al.*, “Identification of Daily Activites and Environments Based on the AdaBoost Method Using Mobile Device Data: A Systematic Review,” *Electronics*, vol. 9, no. 1, p. 192, 2020, doi: 10.3390/electronics9010192.

[28] M. Hasnain, M. F. Pasha, I. Ghani, M. Imran, M. Y. Alzahrani, and R. Budiarto, “Evaluating Trust Prediction and Confusion Matrix Measures for Web Services Ranking,” *IEEE Access*, vol. 8, pp. 90847–90861, 2020, doi: 10.1109/ACCESS.2020.2994222.

[29] Y. Wang and S. Fan, “Deep Reinforcement Learning Based on Balanced Strati ed Prioritized Experience Replay for Customer Credit Scoring in Peer-to-Peer Lending,” 2023.

[30] D. Chicco, M. J. Warrens, and G. Jurman, “The Matthews Correlation Coefficient (MCC) is More Informative Than Cohen’s Kappa and Brier Score in Binary Classification Assessment,” *IEEE Access*, vol. 9, no. Mcc, pp. 78368–78381, 2021, doi: 10.1109/ACCESS.2021.3084050.

[31] M. K. S. Verma *et al.*, “On-Board State Estimation in Electrical Vehicles: Achieving Accuracy and Computational Efficiency through an Electrochemical Model,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 2563–2575, 2020, doi: 10.1109/TVT.2020.2966266.

[32] M. Heydarian and T. E. Doyle, “MLCM : Multi-Label Confusion Matrix,” pp. 19083–19095, 2022.

[33] D. Chen, Y. Lu, and C. Y. Hsu, “Measurement Invariance Investigation for Performance of Deep Learning Architectures,” *IEEE Access*, vol. 10, no. July, pp. 78070–78087, 2022, doi: 10.1109/ACCESS.2022.3192468.

[34] D. Chicco, V. Starovoitov, and G. Jurman, “The Benefits of the Matthews Correlation Coefficient (MCC) over the Diagnostic Odds Ratio (DOR) in Binary Classification Assessment,” *IEEE Access*, vol. 9, no. Mcc, pp. 47112–47124, 2021, doi: 10.1109/ACCESS.2021.3068614.

BIOGRAPHY OF AUTHORS

Rizky Elinda Sari, is an undergraduate student of Department of Computer System, Faculty of Computer Science, Universitas Sriwijaya. She can be contacted via email : 09011282025084@student.unsri.ac.id

Deris Stiawan, received the Ph.D. degree in computer science from the Universiti Teknologi Malaysia, Malaysia. He is currently a Professor at Department of Computer Engineering, Faculty of Computer Science, Universitas Sriwijaya. His research interests include computer networks, intrusion detection/prevention systems, and heterogeneous networks . He can be contacted via email : deris@unsri.ac.id

Nurul Afifah, She is Lecturer at the Faculty of Computer Science, Universitas Sriwijaya. She is also researcher at Comnets RG Unsri concentration of Infosec, especially in the field of malware. She can be contacted via email : nurul@unsri.ac.id

Mohd. Yazid Idris, received the M.Sc. degree in software engineering, in 1998, and the Ph.D. degree in information technology (IT) security, in 2008. In software engineering, he focuses on the research of designing and development of mobile and telecommunication software. He is currently an Associate Professor with the Faculty of Engineering and Faculty of Computing, Universiti Teknologi Malaysia. His main research interests includes IT security in the area of intrusion prevention and detection (IPD).

Rahmat Budiarto, received the B.Sc. degree in mathematics from Bandung Institute of Technology, Indonesia, in 1986, and the M.Eng. and D.Eng. degrees in computer science from the Nagoya Institute of Technology, Japan, in 1995 and 1998, respectively. He is currently a Full Professor with the College of Computer Science and Information Technology, Al-Baha University, Saudi Arabia. His research interests include intelligent systems, brain modeling, IPv6, network security, wireless sensor networks, and MANETs.