Optimizing Performance of AdaBoost Algorithm through Undersampling and Hyperparameter Tuning on CICIoT 2023 Dataset

Sahrul Yudha Fahrezi[1], Adhitya Nugraha[2], Ardytha Luthfiarta[3], Nauval Dwi Primadya[4]

Program Studi Teknik Informatika,
Fakultas Ilmu Komputer,
Universitas Dian Nuswantoro, Semarang.
[1]yudhafahrezi30@gmail.com, [2]adhitya@dsn.dinus.ac.id,
[3]ardytha.luthfiarta@dsn.dinus.ac.id, [4]primadya021@gmail.com

Abstrak

Peningkatan penggunaan *Internet of Things* (IoT) di berbagai sektor menimbulkan tantangan baru terkait keamanan dan perlindungan terhadap serangan Siber. Koneksi perangkat IoT ke jaringan Internet membuat perangkat IoT rentan terhadap berbagai jenis serangan. Salah satu pendekatan untuk mencegah serangan pada perangkat IoT adalah melakukan analisis jaringan menggunakan algoritma *Machine Learning*, seperti *AdaBoost*. Pada *paper* ini, dilakukan percobaan untuk mengoptimalkan algoritma *Adaboost* dalam melakukan pengklasifikasian. Optimalisasi dilakukan dengan cara menerapkan *Random Under Sampling* dan juga *GridSearchCV* untuk parameter *n_estimator* dan *Algorithm*. Hasilnya menunjukkan bahwa setelah melakukan *Random Under Sampling*, Akurasi meningkat menjadi 0.44. Setelah dilakukan *Hyperparameter Tunning*, akurasinya meningkat menjadi 0.78. Optimalisasi ini menunjukkan pentingnya penyetelan parameter dalam algoritma pembelajaran mesin untuk meningkatkan efektivitas langkah-langkah keamanan Siber untuk perangkat IoT.

Kata kunci: *AdaBoost, Grid SearchCV, IoT, Undersampling*

Abstract

The escalating utilization of the Internet of Things (IoT) in various sectors presents new challenges related to security and protection against cyberattacks. The connection of IoT devices to the Internet network makes them vulnerable to various types of attacks. One approach to preventing attacks on IoT devices is to perform analysis based on network traffic using machine learning algorithms, such as AdaBoost. In this paper, an experiment was carried out to optimize the Adaboost Algorithm to do a classification. Optimization was carried out by applying Random Under Sampling and also GridSearchCV for the n_estimator and Algorithm parameters. The results show that after carrying out Random Under Sampling the accuracy increased to 0.44. After doing Hyperparameter Tuning the accuracy increased to 0.78. This optimization shows the importance of parameter tuning in machine learning algorithms to improve the effectiveness of Cybersecurity measures for IoT devices.

Keywords: AdaBoost, GridSearchCV, IoT, Undersampling

Received: 6 Maret 2024 Revised: 13 Mei 2024 Accepted: 16 Mei 2024

1. Introduction

The increasing prevalence of the Internet of Things (IoT) in various sectors presents new challenges related to security and protection against cyberattacks [1]. The connection of IoT devices to the Internet network makes them vulnerable to various types of attacks. Some examples of common cyberattacks on IoT devices include Distributed Denial of Service (DDoS), Denial of Service (DoS), Recon, Web-based, Brute force, Spoofing, and Mirai [2], [3], [4], [5]. To prevent cyberattacks on IoT devices, important measures include regular firmware and software updates, the use of strong passwords, encryption of data, the use of firewalls, access restrictions, and network traffic monitoring [6]. To mitigate these risks, it is important to develop machine learning algorithms that are effective in detecting and preventing attacks on IoT [7]. Machine learning can be used to detect attacks by analyzing energy consumption on IoT devices, analyzing memory, and even detecting attacks through network traffic analysis [8], [9].

In research [10] machine learning was used to detect anomalies based on energy consumption in IoT devices. The research showed a change in energy consumption on devices affected by malware. The use of machine learning proved effective for detecting attacks such as DoS/DDoS with an accuracy of 0.99. Nugraha A. in his research, trained the decision tree algorithm with the CIC MalMem 2022 dataset to detect and classify malware based on memory usage on IoT devices. In this study, the Decision Tree got an accuracy of 0.99[11]. Tang et al, in their research, tried to detect LDoS attacks using the AdaBoost algorithm. In this study, a feature reduction algorithm was also used to optimize the accuracy value of the AdaBoost algorithm. The final result of this research, the AdaBoost algorithm has an accuracy of 0.97[12].

In research [8], Bot-IoT dataset is used to perform attack classification based on network traffic. In this research, the AdaBoost algorithm is one of the effective algorithms for classifying network traffic with an accuracy of 0.97. Nonetheless, in the research [9] Using the CICIoT 2023 dataset which is divided into 8 classes, the AdaBoost algorithm gets a fairly low accuracy of 0.35. This value is quite low when compared to other machine learning algorithms such as Logistic Regression with an accuracy of 0.83, Perceptron with an accuracy of 0.86, and Random Forest with an accuracy of 0.99.

One of the main advantages of AdaBoost is its ability to combine multiple weak models into a strong model [13]. This allows the AdaBoost algorithm to work in imbalanced data and can improve the accuracy of the classification model without overfitting the data[12]. Therefore, this algorithm is suitable for use in detecting malicious and benign network traffic. In his research, Bruno Guilherme managed to increase the accuracy of the AdaBoost algorithm by adjusting the hyperparameters. In this research, the parameters adjusted in the AdaBoost algorithm are n_estimator and algorithm. The Hyperparameter adjustment process is carried out using the Grid Search and Cross-Validation methods[14]. Due to these problems, this research aims to optimize the parameters of the AdaBoost algorithm using grid search to improve performance in predicting attacks on IoT devices. This research was conducted using a public dataset, namely the Canadian Institute for Cybersecurity IoT 2023 (CICIoT 2023). This research is expected to be a strong basis for decision-making regarding the classification of attacks on IoT devices.

2. Research Method

Based on Figure 1, exploration is done on the data used to identify the features present. The next stage involves preprocessing, where data balancing and normalization are performed to transform the data within a certain range. After that, the data is divided into Train and Test subsets. The next stage is Hyperparameter Tuning and Cross-Validation to optimize the AdaBoost classification model. In the final stage, model evaluation is performed using Test data.

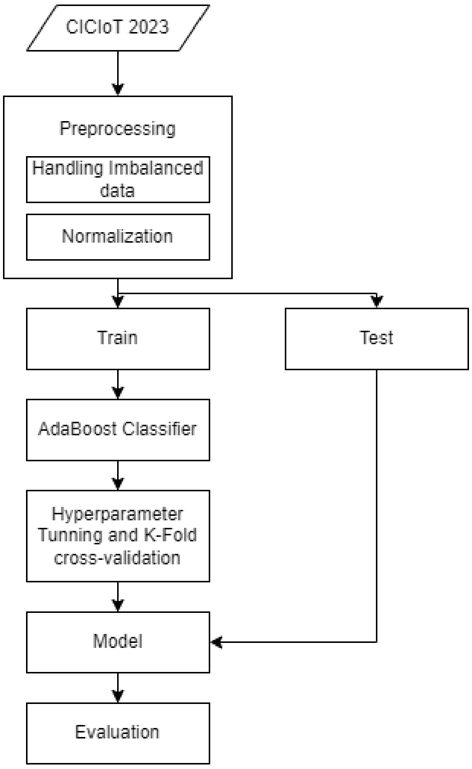

Figure 1 Proposed Method

2.1. Dataset

This research uses the CICIOT Attack 2023 Dataset. This dataset has 46 features and 46,686,579 rows which are grouped into 33 labels of malicious network traffic and 1 label of non-malicious network traffic. The 34 labels are grouped into 7 malicious network traffic labels namely DDoS, DoS, Recon, Web-based, Brute Force, Spoofing, and Mirai, and 1 harmless network traffic label namely Benign [9]. (source: https://www.unb.ca/cic/datasets/iotdataset-2023.html).

2.2. Pre-Processing

At this stage, data balancing is performed. By balancing the data, the unbalanced class proportion can be overcome. This can reduce accuracy errors because the data can only guess correctly in the majority class. Unbalanced data can also result in low recall values that eventually lead to overfitting. Random Under Sampling (RUS) works by selecting samples randomly. This is done until the number of classes which previously had the majority had the same number in each class. This process is carried out using the Random Under Sampling library from imbalanced-learn.

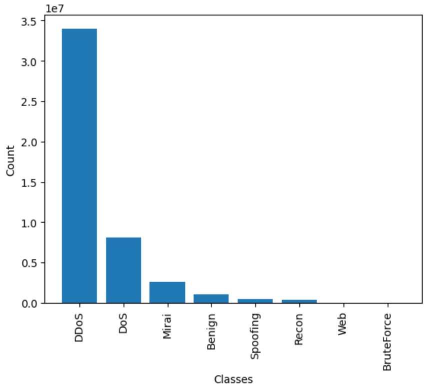

Figure 2 Before Undersampling

Figure 2. illustrates an imbalance in the data distribution among classes, with the DDoS class comprising 3.3 million instances, whereas the other classes contain fewer than 1 million instances each.

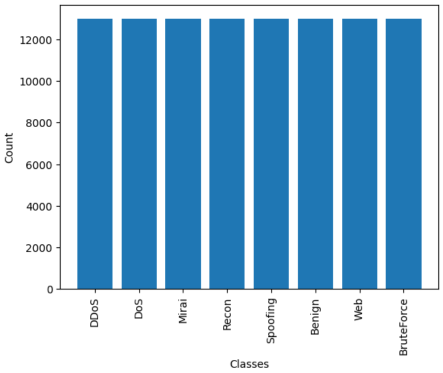

Figure 3 After Undersampling

Figure 3 is the final result of the data after under-sampling, with data that has been balanced for each class.

Normalization is performed using Standard Scaler. StandardScaler is a normalization method used in data preprocessing to convert numeric features to have a mean of 0 and a variance of 1. Using StandardScaler, each feature value is converted into a z-score, which is the distance from the mean in standard deviation units. The purpose of this method is to ensure that the different scales of the variables do not affect the performance of the classification model, thus facilitating comparison and interpretation of the feature coefficients in the model.

Tabel 1 Encoding Result

Before Encoding	After Encoding
Benign	0
BruteForce	1
DDoS	3
DoS	4
Mirai	5
Recon	5
Spoofing	6
Web	7

After encoding was applied, the Label column, which originally contains values with the string data type, has been effectively converted into an integer format. This conversion has an important role in converting categorical data into numerical data, thus enabling its utilization in various machine learning algorithms that specifically rely on numerical inputs. The encoding process can replace the string values in the Label column with corresponding integer values that accurately represent each category.

2.3. Train and Testing Set

In machine learning, it is important to divide data into a training set and a testing set to avoid overfitting when learning dependencies from data. The training set is used to train the model, and the testing set is used to measure the accuracy of the resulting model. Empirical studies show that the best results are obtained if use 20-30% of the data is for testing, and the remaining 70-80% of the data is for training. This is because using all available data points to determine model parameters often leads to overfitting, especially if it is not absolutely certain that the current model is adequate. By leaving some data for testing, the model can be evaluated for its performance on unseen data and ensure that the model can generalize well. In this study, the data is divided into two parts with the proportion of train data and test data being 80%: 20%. The division of data is done randomly with stratification based on the class in the dataset.

2.4. Hyperparameter Tuning

Hyperparameter tuning is important in improving the performance of the AdaBoost algorithm on the IoT Attack dataset because it enables the identification of optimal combinations of hyperparameters that can improve accuracy, precision, recall, or other relevant evaluation metrics. The IoT Attack dataset is a complex and diverse dataset containing different types of attacks and attack scenarios. Therefore, it is important to tune the hyperparameters of the AdaBoost algorithm to ensure that it can accurately classify the different types of attacks in the dataset.

In this research, the hyperparameter tuning algorithm used is GridSearch. Grid search is an optimization method that makes equally spaced grid points and then calculates the accuracy for each parameter so that the most optimal parameter point is found. In this research, the AdaBoost algorithm is optimized by varying the algorithm and n_estimator parameters. Then in its application combined with the cross-validation method. Cross-validation is a development method of the split validation model where the validation measures training error with test data. The cross-validation value used in this research is 5.

2.5. AdaBoost Classifier

Adaptive Boosting (AdaBoost) is a machine learning algorithm that combines multiple weak models to create a strong model. It works by repeatedly training a series of weak models on the same data set, with each subsequent model placing more emphasis on the misclassified data points of the previous model. The final model is a weighted combination of the weak models, with the weight of each model determined by its accuracy in classifying the data.

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right) \tag{1}$$

where $H(x)$ is strong classifier, $h_t(x)$ is weak classifier, α_t is weight.

Based on equation (1), In order to predict the class label for the input data x, the AdaBoost formula combines the predictions from each iteration ($h_t(x)$) with the corresponding weights (α_t). The sum of all these predictions is taken and then converted into binary class labels using the sign() function.

In the AdaBoost algorithm, there are several parameters that must be considered namely, the number of estimators or n_estimators and algorithm parameters. Number of estimators refers to the number of estimators used in the boosting process. The estimator is a decision tree model with only one level. A higher number of n_estimators tends to produce a better model but is prone to overfitting. In the AdaBoost model, the default value for this parameter is 50. The algorithm parameter is a parameter used to determine the algorithm that will be used to train the base estimator. SAMME.R and SAMME are commonly used values in the algorithm parameter.

2.6. Evaluation

When evaluating the performance of a machine learning model, it is important to define a performance measure that is appropriate for the task at hand. To evaluate the results, this study used the most important performance indicators for accuracy, precision, f1-Score, and recall as shown in the equations below:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$F1 - Score = 2 * \frac{Recall * Precision}{Recall + Precision} \tag{5}$$

3. Result

In this study, the first experiment conducted was to train the AdaBoost model on the CICIoT 2023 dataset after undersampling. The experiment was conducted using default values for the n_estimator and algorithm parameters.

Tabel 2 Result After Random Under Sampling

No	Research	Accuracy	Precision	Recall	F1-Score
1.	Neto et al [9]	0.35	0.46	0.48	0.36
2.	This research	0.44	0.40	0.45	0.45

Based on Table 2, it was found that the use of the AdaBoost algorithm for network attack classification on IoT devices resulted in an accuracy of 0.44 or 44%. In addition, the precision achieved is 40%, which indicates the algorithm's ability to correctly identify network attacks from the total positive predictions. The results also showed a recall of 45%, which illustrates the algorithm's ability to correctly detect network attacks from the total attacks. F1-Score, which reflects the balance between precision and recall, reached 45%.

The next step is training with the training data. The AdaBoost algorithm was optimized using GridSearchCV. The optimization is done by varying the algorithm parameters namely SAMME and SAMME.R and also the n_estimator parameters 5, 25, 50, 75, 100, 250, 500, 600, 700, 800, 1000.

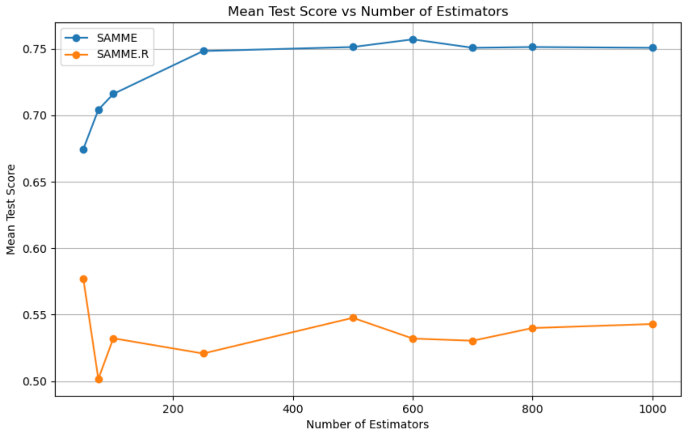

Figure 4 Comparison Results of SAMME and SAMME.R

For the SAMME algorithm increasing the n_estimator above 250 does not provide a high enough increase in accuracy, whereas for the SAMME.R algorithm changing the n_estimator provides different results. although the results are poor when compared with the SAMME algorithm.

The best results are when the n_estimator value reaches the maximum of 600. There was a significant improvement in accuracy (0.77) and precision (0.84), demonstrating the AdaBoost algorithm's ability to generate a few false positives. Recall remained high (0.77), demonstrating the algorithm's ability to detect most network attacks that occur on IoT devices. Can be seen from the heatmap in Figure 5 that the AdaBoost algorithm can classify correctly for several classes such as Benign, BruteForce, DDoS, DoS, and Mirai.

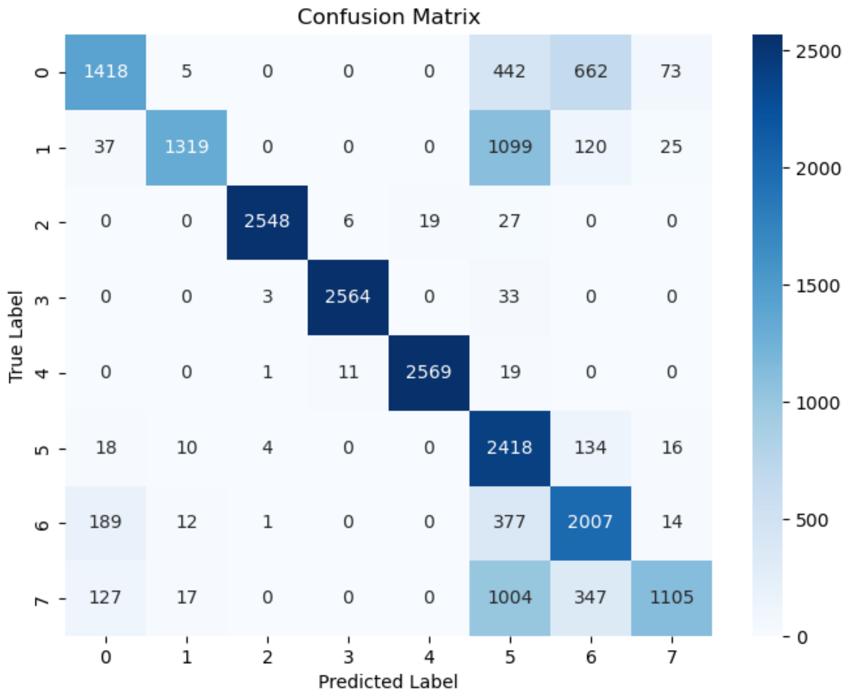

Figure 5 Heatmap of the Best Adaboost Models

4. Conclusions

Based on the research of AdaBoost algorithm implementation with hyperparameter tuning on the CICIoT 2023 dataset, it can be concluded that undersampling of the data provides an increase in the accuracy matrix. The use of hyperparameter tuning can also improve the performance of the AdaBoost algorithm. Based on the evaluation results, it can be concluded that with a comparison of 80% training data division and 20% test data, the most optimal algorithm value is SAMME and n_estimator 600 with an accuracy value of 76%, precision 84%, recall 0.77, F1-Score 0.77. This result succeeded in outperforming the use of the adaboost algorithm in the previous paper. Future research can consider other factors such as data preprocessing, the use of more appropriate features, or the use of other ensemble algorithms to achieve better results.

References

[1] C. Lee and G. Ahmed, "Improving IoT Privacy, Data Protection and Security Concerns," *International Journal of Technology, Innovation and Management (IJTIM)*, vol. 1, no. 1, pp. 18–33, Sep. 2021, doi: 10.54489/ijtim.v1i1.12.

[2] U. Inayat, M. F. Zia, S. Mahmood, H. M. Khalid, and M. Benbouzid, "Learning-Based Methods for Cyber Attacks Detection in IoT Systems: A Survey on Methods, Analysis, and Future Prospects," *Electronics (Basel)*, vol. 11, no. 9, p. 1502, May 2022, doi: 10.3390/electronics11091502.

[3] S. Kato, Tanabe Rui, K. Yoshioka, T. Matsumoto, "Adaptive observation of emerging cyber attacks targeting various IoT devices," in *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pp. 143–151, 2021.

[4] N. Agrawal, S. Tapaswi, "Defense mechanisms against DDoS attacks in a cloud computing environment: state-of-the-art and research challenges," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 4, pp. 3769–3795, Oct. 2019, doi: 10.1109/COMST.2019.2934468.

[5] A. Banitalebi Dehkordi, M. R. Soltanaghaei, F. Z. Boroujeni, "The DDoS attacks detection through machine learning and statistical methods in SDN," *Journal of Supercomputing*, vol. 77, no. 3, pp. 2383–2415, Mar. 2021, doi: 10.1007/s11227-020-03323-w.

[6] P. L. W. E. Putra, C. A. Sari, and F. O. Isinkaye, "Secure text encryption for IoT communication using affine cipher and diffie-hellman key distribution on arduino atmega2560 IoT devices," *Jurnal Teknik Informatika (Jutif)*, vol. 4, no. 4, pp. 849–855, Aug. 2023, doi: 10.52436/1.jutif.2023.4.4.1129.

[7] S. M. Tahsien, H. Karimipour, and P. Spachos, "Machine learning based solutions for security of Internet of Things (IoT): A survey," *Journal of Network and Computer Applications*, vol. 161, p. 102630, Jul. 2020, doi: 10.1016/j.jnca.2020.102630.

[8] N. Patil, "Using machine learning in detecting IoT cyber attacks," *Int J Res Appl Sci Eng Technol*, vol. 10, no. 11, pp. 472–478, Nov. 2022, doi: 10.22214/ijraset.2022.47365.

[9] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, A. A. Ghorbani, "CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment," *Sensors*, vol. 23, no. 13, p. 5941, Jun. 2023, doi: 10.3390/s23135941.

[10]K. Bobrovnikova, S. Lysenko, P. Popov, D. Denysiuk, A. Goroshko, "Technique for IoT cyberattacks detection based on the energy consumption analysis," in

International Workshop on Intelligent Information Technologies & Systems of Information Security, 2021. Accessed: Oct. 23, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:233432271

[11]A. Nugraha and J. Zeniarja, "Malware Detection Using Decision Tree Algorithm Based on Memory Features Engineering," *Journal of Applied Intelligent System*, vol. 7, no. 3, pp. 206–210, Dec. 2022, doi: 10.33633/jais.v7i3.6735.

[12]D. Tang, L. Tang, R. Dai, J. Chen, X. Li, J. J. P. C. Rodrigues, "MF-Adaboost: LDoS attack detection based on multi-features and improved Adaboost," *Future Generation Computer Systems*, vol. 106, pp. 347–359, May 2020, doi: 10.1016/j.future.2019.12.034.

[13]M. M. Kholil, F. Alzami, M. A. Soeleman, "AdaBoost based C4.5 accuracy improvement on credit customer classification," in *2022 International Seminar on Application for Technology of Information and Communication: Technology 4.0 for Smart Ecosystem: A New Way of Doing Digital Business, iSemantic 2022*, Institute of Electrical and Electronics Engineers Inc., pp. 351–356, 2022, doi: 10.1109/iSemantic55962.2022.9920463.

[14]B. G. Carvalho, R. E. V. Vargas, R. M. Salgado, C. J. Munaro, F. M. Varejão, "Hyperparameter tuning and feature selection for improving flow instability detection in offshore oil wells," in *IEEE International Conference on Industrial Informatics (INDIN)*, Institute of Electrical and Electronics Engineers Inc., 2021. doi: 10.1109/INDIN45523.2021.9557415.

This Page Intentionally Left Blank