Matrik: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer
Vol. 25, No. 1, November 2025, pp. 83∼96
ISSN: 2476-9843, accredited by Kemenristekdikti, Decree No: 10/C/C3/DT.05.00/2025
DOI: 10.30812/matrik.v25i1.5382

Optimizing Random Forest for IoT Cyberattack Detection Using SMOTE: A Study on CIC-IoT2023 Dataset

Guntoro [1,2], Lisnawita [1], Loneli Costaner[1]
[1]Universitas Lancang Kuning, Pekanbaru, Indonesia
[2]Universiti Utara Malaysia, Kedah, Malaysia

Article Info

Article history:

Received July 21, 2025
Revised September 22, 2025
Accepted October 10, 2025

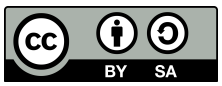
Keywords:

CIC-IoT2023;
Internet of Things;
Intrusion Detection System;
SMOTE;
Random Forest.

ABSTRACT

The growing number of Internet of Things devices has led to an increased risk of cyberattacks. A challenge is the imbalanced class distribution in Internet of Things datasets, which can cause classification algorithms to be biased towards the majority class and hinder effective threat detection. This study addresses this issue by leveraging the Random Forest algorithm optimized by the Synthetic Minority Oversampling Technique (SMOTE). The research aims to develop an effective model for detecting cyberattacks in Internet of Things (IoT) environments by addressing class imbalance within the CIC-IoT2023 dataset. The methodology involves data preprocessing and the application of SMOTE for data balancing. The balanced dataset was used to train a Random Forest model, and its performance was evaluated utilizing accuracy, precision, recall, F1-score, and Cohen's Kappa metrics. The results demonstrate the model's effectiveness, achieving an accuracy of 99.01%, an F1-score of 98.96%, and a Cohen's Kappa of 98.92%. This marks a performance improvement, particularly in detecting minority classes, compared to the model trained without SMOTE, which struggled to identify less common attack types. The outcomes suggest that combining Random Forest with SMOTE can significantly enhance intrusion detection systems by improving detection accuracy for all 33 attack types and reducing the risks associated with undetected threats. In conclusion, this study advances Internet of Things cybersecurity by presenting a method for addressing data imbalance in attack detection and calls for further evaluation of the model's robustness on more complex datasets and its performance in real-time deployment on resource-constrained Internet of Things devices.

Copyright ©2025 The Authors.
This is an open access article under the CC BY-SA license.



Corresponding Author:

Guntoro, +6285271220118,
Faculty of Computer Science, Informatic Engineering,
Universitas Lancang Kuning, Riau, Indonesia,
Email: guntoro@unilak.ac.id.

How to Cite:
G. Guntoro, L. Lisnawita, and L. Costaner, "Optimizing Random Forest for IoT Cyberattack Detection Using SMOTE: A Study on CIC-IoT2023 Dataset", *MATRIK: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer*, Vol. 25, No. 1, pp. 83-96, November, 2025.
This is an open access article under the CC BY-SA license (https://creativecommons.org/licenses/by-sa/4.0/)

1. INTRODUCTION

The rise of the Internet of Things (IoT) has led to a significant transformation in global automation and communication systems. By connecting billions of smart devices across sectors such as healthcare, manufacturing, transportation, and smart homes, IoT is reshaping how data is generated, shared, and utilized [1]. However, this rapid expansion also introduces significant security vulnerabilities. A wide variety of devices characterizes IoT technology, many of which have limited computing power, memory, and energy, making them particularly susceptible to a range of cyberattacks [2]. These limited resources often impede the deployment of conventional security systems, leaving IoT devices increasingly vulnerable to cyberattacks. According to a recent report by Kaspersky 2023 [3], there has been a more than 40% increase in attacks targeting IoT devices over the past two years. These outcomes are consistent with academic analyses that highlight a growing surge in cyber threats within the IoT ecosystem, comprising Distributed Denial of Service (DDoS) attacks and malware [4, 5]. This highlights the urgent need to strengthen IoT security to safeguard data and ensure system reliability. Machine learning (ML)-based intrusion detection systems have emerged as an effective solution for identifying suspicious activities inside IoT network traffic [6]. Among the widely used algorithms, Random Forest (RF) stands out for its ability to manage high-dimensional datasets, resist overfitting, and deliver consistently accurate results, making it a robust tool for data analysis [7]. However, a significant challenge in cybersecurity is the issue of imbalanced label distribution [8]. This imbalance often results in minority classes, such as SQL Injection or Command Injection attacks, being overlooked due to the dominance of more frequent classes like DDoS attacks. Consequently, the detection of high-risk threats becomes less effective [9].

To mitigate this problem, preprocessing techniques such as the Synthetic Minority Oversampling Technique (SMOTE) have been widely applied. SMOTE generates synthetic samples for minority classes, thereby improving model sensitivity toward rare yet critical attacks. Several studies have reported its effectiveness in IDS contexts [10–14], showing improved performance across classifiers such as Support Vector Machines, Neural Networks, and Random Forest. Other approaches, including hybrid sampling [15], adaptive resampling [16], and ensemble learning [17], also demonstrated promising improvements. Nevertheless, most of these works were conducted on older datasets such as NSL-KDD or CICIDS2017, with limited application to more recent and complex IoT traffic.

CIC-IoT2023 is the latest publicly available dataset, specifically designed to illustrate IoT traffic patterns and various real-world attack scenarios [12]. The dataset comprises 33 attack classes, encompassing a wide range of threats, including DoS, DDoS, reconnaissance, and web application attacks. However, the complexity and pronounced class imbalance within the CIC-IoT2023 dataset present significant challenges in building effective intrusion detection system (IDS) models using machine learning techniques. The difference between this research and previous research is that prior studies mainly focused on evaluating SMOTE or other balancing methods on older datasets or simplified binary class detection. In contrast, this study explicitly investigates how data imbalance affects the performance of Random Forest in a multi-class intrusion detection scenario using the CIC-IoT2023 dataset, with a particular emphasis on detecting minority classes. The novelty of this research is the explicit combination of SMOTE and Random Forest for multi-class IoT intrusion detection using CIC-IoT2023, supported by a comprehensive comparative analysis of classification metrics before and after balancing. This approach provides deeper insights into the trade-off between sensitivity and precision, offering both methodological contributions and practical implications for IoT security.

Building on the context mentioned earlier, this study aims to assess the SMOTE technique in comparison to the performance of the Random Forest algorithm in detecting cyberattacks on the CIC-IoT2023 dataset. This research focuses on enhancing the model's ability to identify minority classes without compromising accuracy. The main contributions of this study include evaluating the performance of SMOTE on a highly imbalanced IoT dataset and implementing and validating the SMOTE-Random Forest algorithm in a multi-class detection scenario on the CIC-IoT2023 dataset. The study also compares classification performance metrics before and after the balancing process to assess the impact on model sensitivity and precision. In contrast to prior studies that predominantly emphasized overall accuracy or binary detection schemes, this research highlights the importance of class balance in multi-class intrusion detection, where minority attacks are often underrepresented yet highly consequential. By addressing this gap, the study advances methodological approaches while offering practical implications for building more reliable and equitable intrusion detection systems in IoT environments.

2. RESEARCH METHOD

This study applies the Random Forest (RF) algorithm to detect cyberattacks in Internet of Things (IoT) environments, combined with the Synthetic Minority Oversampling Technique (SMOTE) [10, 11] to address the class imbalance problem. The methodological workflow was designed to ensure experimental validity and consists of six main stages. First, dataset collection and preprocessing were conducted, including feature extraction and labeling to guarantee data quality and relevance [5]. Second, the dataset was divided into training and testing subsets to enable an unbiased evaluation of model generalization [2]. Third, feature normalization was performed to maintain scale uniformity and improve classifier performance by mitigating the influence of heterogeneous feature

ranges [17]. Fourth, SMOTE was applied to generate synthetic samples of minority classes, thereby reducing bias toward majority classes [8]. Fifth, the Random Forest classifier was implemented with optimized hyperparameters to enhance predictive accuracy and robustness, leveraging its ensemble structure. Finally, model evaluation was conducted using confusion matrix-based metrics, including accuracy, precision, recall, and F1-score, to provide a comprehensive view of both overall performance and minority class detection. The workflow is illustrated in Figure 1, which presents a structured overview of the research methodology.

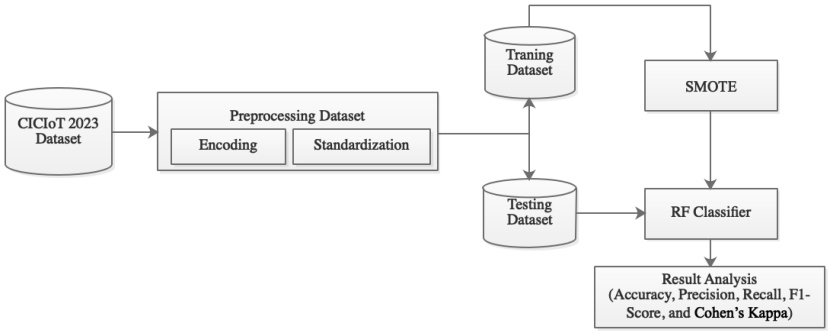


Figure 1. Conceptual view of the research methodology

2.1. Dataset

The dataset used in this study is CIC-IoT2023, developed by the Canadian Institute for Cybersecurity (CIC) as part of its ongoing effort to provide high-quality, publicly available resources for network security research. Recognized as one of the most recent and comprehensive benchmarks for evaluating IoT intrusion detection systems, it encompasses 33 distinct attack types spanning multiple threat categories, including Distributed Denial-of-Service (DDoS), Denial-of-Service (DoS), reconnaissance, malware, and web application attacks [12, 17]. The dataset was generated through controlled experimental setups involving a diverse range of IoT devices such as IP cameras, routers, smart sensors, and other connected endpoints, thereby capturing realistic and heterogeneous network traffic patterns. Each traffic instance is labeled according to its corresponding attack or benign category, ensuring precise ground-truth references for supervised machine learning tasks. This combination of breadth in attack coverage, realism in traffic generation, and precise labeling makes CIC-IoT2023 a reliable and representative benchmark for assessing the performance and generalization capabilities of intrusion detection models in real-world IoT environments. Importantly, these characteristics directly support the aim of this study, which is to evaluate the effectiveness of SMOTE and Random Forest in handling class imbalance while maintaining robust detection across diverse attack types.

2.2. Preprocessing Dataset

The data preprocessing phase is essential for converting raw inputs into a structured format suitable for machine learning models [13, 15, 15]. This stage involves two key procedures: feature encoding for categorical variables and standardization for numerical features [16]. Feature encoding ensures that categorical information is meaningfully represented in numerical form without introducing bias, while standardization places numerical features on a comparable scale to prevent attributes with larger ranges from dominating the learning process [15, 17].

2.2.1. Feature Encoding

Categorical features, such as flow duration, header length, and protocol type, are transformed into numerical representations utilizing one-hot encoding [13, 16]. This encoding approach preserves the semantic integrity of each categorical variable while avoiding the introduction of unintended ordinal relationships or biases that could mislead the learning process. In this method, each category is converted into a binary vector where a single bit is set to one, and all others are set to zero, thereby enabling algorithms such as Random Forest to process categorical data effectively without imposing any artificial hierarchy among feature values [16]. The use of one-hot encoding in this study ensures that the diversity of categorical attributes in the CIC-IoT2023 dataset is represented faithfully, which is crucial for maintaining the reliability and fairness of the intrusion detection model's training and evaluation phases [15]. Moreover, this procedure directly supports the study's objective of building a balanced and unbiased detection system that can generalize well across heterogeneous IoT traffic scenarios.

2.2.2. Data Standardization

Given that the IDS dataset contains features with diverse value ranges and varying scales, data standardization was applied to ensure consistency and comparability across all attributes [13, 15]. Standardization transforms the distribution of each feature from its original scale to a standard normal distribution, thereby centering the data around zero and scaling it based on its variability. In practical terms, this rescaling process ensures that each feature has a mean value of zero and a standard deviation of one, which is particularly important for machine learning algorithms that are sensitive to differences in feature magnitude. This transformation was performed using the standard score (z-score) method, in Equation 1.

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

where x denotes the observed sample value, μ is the mean of the feature, and σ is its standard deviation. This standardization centers each feature at zero and scales it to unit variance, which prevents variables with larger numerical ranges from dominating the learning process. As a result, optimization becomes more stable and the overall robustness of the intrusion detection model improves [18].

2.3. Imbalancing Data

The CIC-IoT2023 dataset exhibits a significant class imbalance, with the majority of samples concentrated in dominant attack types such as DDoS-ICMP_Flood and DDoS-UDP_Flood. To mitigate this issue, the Synthetic Minority Over-sampling Technique (SMOTE) is utilized, with the k_neighbors parameter set to 2 [10]. This configuration was chosen because preliminary experiments indicated that a smaller neighborhood yields more realistic synthetic instances for highly sparse minority classes. At the same time, larger k values tended to generate less representative samples. This aligns with prior studies that recommend lower k settings when dealing with extreme imbalance, as it reduces the risk of producing noisy or overlapping samples [11]. Importantly, SMOTE is applied exclusively to the training data to avoid information leakage into the testing set, thereby preserving the integrity of the model evaluation.

2.4. Classification

The classification model employed in this study is Random Forest (RF), an ensemble learning algorithm that constructs multiple decision trees utilizing different data subsets. The final prediction is determined through a majority voting mechanism, enhancing both accuracy and robustness [14]. Random Forest was selected for its strong capability to process large-scale datasets and its robustness against noisy data. In this study, the RF classifier was configured with key hyperparameters, including 100 decision trees (n_estimators = 100), a maximum tree depth of 10 (max_depth = 10), and the square-root rule for the number of features considered at each split (max_features = sqrt). These parameter settings were chosen based on preliminary experiments to achieve a balance between predictive performance and computational efficiency, while also preventing overfitting.

2.5. Evaluation

The performance of the proposed model was assessed using a confusion matrix, which provides a detailed summary of the classification outcomes based on four key components: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). These elements serve as the basis for calculating several important evaluation metrics, comprising. Equation 2 is used to compute the accuracy metric, which serves as a fundamental performance indicator for evaluating the effectiveness of the intrusion detection model. Accuracy is defined as the ratio of correctly predicted instances, including both attack and normal classes, to the total number of samples in the dataset. This metric reflects the overall proportion of correct classifications made by the model and is expressed mathematically as.

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

Equation 3 is used to calculate the precision metric, which measures the proportion of correctly predicted attack cases relative to the total number of instances classified as attacks by the model. Precision focuses on the quality of positive predictions, indicating how many of the instances labeled as attacks are indeed true attacks. It is mathematically expressed as.

$$Presion = \frac{TP}{TP + FP} \tag{3}$$

Equation 4 is used to calculate the detection rate (also referred to as recall or true positive rate), which reflects the proportion of correctly identified attack cases relative to the actual number of attack instances present in the dataset. This metric assesses the model's ability to identify malicious activities without overlooking genuine threats. It is mathematically expressed as.

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

The F-Score (also referred to as the F1-Score) measures the harmonic mean between the detection rate (also known as recall) and precision, thereby providing a single value that balances the trade-off between correctly identifying attacks and minimizing false alarms. This metric is particularly useful when the class distribution is imbalanced, as it penalizes extreme disparities between precision and recall. The F1-Score is calculated using Equation 5.

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

Equation 6 is used to assess the level of agreement between two raters while accounting for the possibility that some degree of agreement may occur purely by chance. This metric is commonly measured using Cohen's Kappa coefficient, which provides a more robust evaluation of inter-rater reliability compared to simple percentage agreement. Cohen's Kappa is mathematically expressed as.

$$k = \frac{P_o - P_e}{1 - P_e} \tag{6}$$

3. RESULT AND ANALYSIS

This section presents the experimental results and analysis of applying the Random Forest algorithm to the CIC-IoT 2023 dataset. The evaluation was carried out under two distinct scenarios: with and without the application of the Synthetic Minority Oversampling Technique (SMOTE) for data balancing. The objective of this evaluation was to assess the model's effectiveness in detecting various types of cyberattacks inside an Internet of Things (IoT) environment, under both imbalanced and balanced data conditions. Model performance was evaluated using multiple metrics, including accuracy, precision, recall, F1 Score, and Cohen's kappa. To gain deeper insights, a confusion matrix and detailed per-class performance reports are also given, offering a clearer understanding of the model's behavior across different attack categories. The test results are systematically organized to highlight the impact of data balancing on classification performance and are further compared with outcomes from previous research.

3.1. Experimental Setup

All experiments were conducted utilizing Google Colab Pro, a cloud-based platform that offers access to high-performance virtual machines. The experiments were implemented in Python 3.10, utilizing a range of essential libraries: Scikit-learn 1.3.2 for machine learning tasks, Imbalanced-learn for handling data imbalance through oversampling, Pandas and NumPy for data manipulation, and Matplotlib along with Seaborn for data visualization.

3.2. Experimental Results

The proposed approach, which combines the SMOTE oversampling technique with the Random Forest algorithm, was thoroughly evaluated using the CIC-IoT2023 dataset. The assessment employed key performance metrics, comprising accuracy, precision, recall, and F1-score, based on a 70:30 train-test split. As shown in Table 1, the results demonstrate a remarkably high classification performance. Table 1 presents the performance evaluation of the Random Forest model after applying SMOTE for data balancing. The model achieved an outstanding accuracy of 99.01%, reflecting a very high level of classification effectiveness. Furthermore, the precision and recall scores of 98.98% and 99.01%, respectively, demonstrate the model's strong capability to accurately detect attacks while minimizing both false positives and false negatives effectively. An F1-score of 98.96% further underscores the model's excellent balance among precision and recall, reinforcing its reliability in detecting real-world threats that demand high sensitivity. This impressive level of accuracy gives strong assurance of the model's effectiveness and robustness in practical applications.

Table 1. The Experimental Results

Metrics	Results (%)
Accuracy	99.01
Precision	98.98
Recall	99.01
F1-score	98.96
Cohen's Kappa	98.92

Moreover, the Cohen's Kappa score of 98.92% indicates a strong agreement among the model's predictions and the actual class labels, even after accounting for chance agreement. These results clearly demonstrate that integrating SMOTE with the Random Forest algorithm is highly effective in improving detection accuracy and mitigating the impact of class imbalance within the CIC-IoT2023 dataset. Consequently, this model can be considered a robust and reliable tool for detecting IoT-based cyberattacks, offering a high degree of confidence in its practical effectiveness.

3.3. The Impact of SMOTE on Class Imbalance

The label distribution in the CIC-IoT 2023 dataset reveals a pronounced class imbalance. Before the application of SMOTE (Synthetic Minority Oversampling Technique), a few classes were heavily overrepresented, most notably DDoS-UDP_Flood by 22,048 samples, DDoS-TCP_Flood by 18,553 samples, and DoS-TCP_Flood by 10,976 samples. In contrast, many of the remaining classes contained only a few hundred or even just a single sample. For instance, SQL Injection had 24 samples, XSS had 12 samples, Uploading Attack had six samples, and Recon-Ping Sweep had only five samples. This severe imbalance can lead to bias in the classification model, causing it to primarily learn patterns from the majority classes while neglecting those of the minority classes.

The application of SMOTE effectively balanced the dataset by generating synthetic instances for minority classes, ensuring an equal number of samples across all categories, resulting in 29,060 samples per class. This oversampling process addressed the inherent class imbalance problem in the CIC-IoT2023 dataset, which, if left uncorrected, could bias the learning algorithm toward majority classes and degrade detection performance for rare attack types. Table 2 provides a detailed summary of the sample distribution for each class before and after applying SMOTE, illustrating the transformation from an imbalanced dataset to a perfectly balanced one. The balanced distribution ensures that the classifier receives an equal representation of all classes during training, thereby enhancing its ability to detect both frequent and infrequent attack categories with comparable accuracy.

Table 2. Comparison of Sample Before and After SMOTE

Label	Before_SMOTE	After_SMOTE
DDoS-UDP_Flood	22048	29060
DDoS-TCP_Flood	18553	29060
DoS-TCP_Flood	10976	29060
DoS-UDP_Flood	13571	29060
DDoS-SYN_Flood	16491	29060
DDoS-ICMP_Flood	29060	29060
DDoS-PSHACK_Flood	17035	29060
DDoS-RSTFINFlood	16597	29060
BenignTraffic	4482	29060
DoS-SYN_Flood	8254	29060
DDoS-ACK_Fragmentation	1194	29060
Mirai-udpplain	3754	29060
Mirai-greeth_flood	4002	29060
DNS_Spoofing	738	29060
DDoS-SynonymousIP_Flood	14664	29060
Recon-OSScan	412	29060
Mirai-greip_flood	3005	29060
DDoS-ICMP_Fragmentation	1883	29060
Recon-HostDiscovery	544	29060
DictionaryBruteForce	53	29060
DDoS-UDP_Fragmentation	1175	29060
MITM-ArpSpoofing	1288	29060

(continued on next page)

Table 2 (continued)

Label	Before_SMOTE	After_SMOTE
DoS-HTTP_Flood	339	29060
SqlInjection	24	29060
VulnerabilityScan	154	29060
Recon-PortScan	350	29060
BrowserHijacking	27	29060
DDoS-SlowLoris	78	29060
DDoS-HTTP_Flood	131	29060
CommandInjection	23	29060
Backdoor_Malware	21	29060
XSS	12	29060
Uploading_Attack	6	29060
Recon-PingSweep	5	29060

Overall, the total number of samples in the dataset increased significantly—by 210,122 to 957,960—following the application of SMOTE. This substantial expansion enables the model to learn more effectively and recognize patterns associated with minority classes. Figure 2 below illustrates a comparison of the label distributions before and after applying SMOTE.

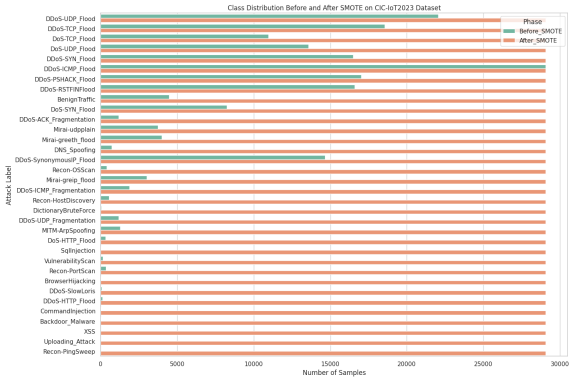


Figure 2. Class distribution before and after SMOTE

These enhancements have a direct and meaningful impact on the model's overall performance. As illustrated in Table 2, both recall and F1-score revealed significant improvement following the application of SMOTE. This indicates that the model was no longer biased toward majority classes and could accurately recognize a broader range of attack types. The balanced dataset achieved through SMOTE was instrumental in reducing misclassification errors, particularly false negatives, for previously underrepresented classes. This is especially critical in the realm of IoT cybersecurity, where failing to detect even minor attacks can result in severe consequences for critical systems. The results reinforce confidence in the model's reliability and robustness.

3.4. Model Comparison Analysis

To assess the impact of the SMOTE algorithm on the performance of the Random Forest-based cyberattack detection model, a comparative analysis was conducted between the baseline Random Forest model (without SMOTE) and the SMOTE-enhanced version. Tables 3 and 4 report detailed per-class performance metrics, comprising precision, recall, and F1-score for all 33 attack classes. Additionally, Figures 3 and 4 display the corresponding confusion matrices, offering a clear visual comparison of classification accuracy across all classes for both models.

Table 3 presents the performance of the Random Forest model devoid of SMOTE. While the model demonstrates high precision and recall for several majority classes—comprising DDoS-ICMP_Flood, DDoS-TCP_Flood, and DoS-SYN_Flood—it completely fails to detect several minority classes, such as Backdoor_Malware, BrowserHijacking, CommandInjection, DictionaryBruteForce, Recon-PingSweep, and SQLInjection, as reflected by an F1-score of 0.00 for these classes. This result clearly indicates a bias toward the majority classes, underscoring the challenges posed by class imbalance in the dataset. Figure 3 displays the confusion matrix for the model devoid of SMOTE. The visualization clearly reveals that the majority of misclassifications occur within the minority attack

classes, whereas the majority classes are classified with near-perfect accuracy. This further highlights the model's tendency to favor dominant classes, resulting in a poor detection of underrepresented threats.

Table 3. Model Classification Evaluation Random Forest without Smote

Label	Precision	Recall	F1-score
Backdoor_Malware	0.00	0.00	0.00
BenignTraffic	0.79	0.99	0.88
BrowserHijacking	0.00	0.00	0.00
CommandInjection	0.00	0.00	0.00
DDoS-ACK_Fragmentation	1.00	0.99	1.00
DDoS-HTTP_Flood	0.97	0.95	0.96
DDoS-ICMP_Flood	1.00	1.00	1.00
DDoS-ICMP_Fragmentation	0.98	0.98	0.98
DDoS-PSHACK_Flood	1.00	1.00	1.00
DDoS-RSTFINFlood	1.00	1.00	1.00
DDoS-SYN_Flood	1.00	1.00	1.00
DDoS-SlowLoris	0.85	0.82	0.84
DDoS-SynonymousIP_Flood	1.00	1.00	1.00
DDoS-TCP_Flood	1.00	1.00	1.00
DDoS-UDP_Flood	1.00	1.00	1.00
DDoS-UDP_Fragmentation	1.00	0.99	0.99
DNS_Spoofing	0.73	0.53	0.61
DictionaryBruteForce	0.00	0.00	0.00
DoS-HTTP_Flood	0.95	0.97	0.96
DoS-SYN_Flood	1.00	1.00	1.00
DoS-TCP_Flood	1.00	1.00	1.00
DoS-UDP_Flood	1.00	1.00	1.00
MITM-ArpSpoofing	0.89	0.74	0.81
Mirai-greeth_flood	1.00	1.00	1.00
Mirai-greip_flood	1.00	1.00	1.00
Mirai-udpplain	1.00	1.00	1.00
Recon-HostDiscovery	0.82	0.73	0.78
Recon-OSScan	0.74	0.25	0.37
Recon-PingSweep	0.00	0.00	0.00
Recon-PortScan	0.86	0.46	0.60
SqlInjection	0.00	0.00	0.00
Uploading_Attack	1.00	0.50	0.67
VulnerabilityScan	0.85	0.91	0.88
XSS	1.00	0.17	0.29

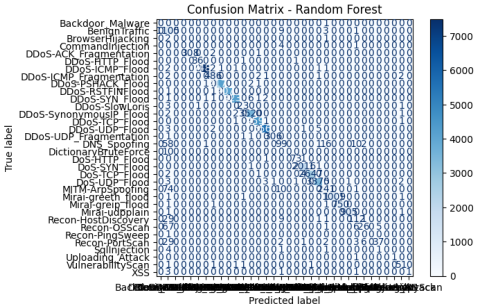


Figure 3. Confusion Matrix – Random Forest devoid of Smote

Table 4 presents the classification results of the Random Forest model enhanced by SMOTE. Following dataset balancing, the model demonstrated notable improvements in detecting minority classes. For instance, DictionaryBruteForce achieved an F1-score of 0.33, up by 0.00, while SQLInjection improved to an F1-score of 0.20. At the same time, the majority of classes continued to maintain high precision and recall, indicating that the model's performance on dominant classes was not compromised. These results

confirm that the SMOTE algorithm effectively enhances the model's ability to detect a broader range of IDS attacks, particularly those that are typically underrepresented in existing datasets.

Table 4. The Performance of Random Forest by Smote

Label	Precision	Recall	F1-score
Backdoor_Malware	0.00	0.00	0.00
BenignTraffic	0.84	0.96	0.89
BrowserHijacking	0.00	0.00	0.00
CommandInjection	0.00	0.00	0.00
DDoS-ACK_Fragmentation	0.99	1.00	0.99
DDoS-HTTP_Flood	1.00	0.97	0.99
DDoS-ICMP_Flood	1.00	1.00	1.00
DDoS-ICMP_Fragmentation	0.97	0.99	0.98
DDoS-PSHACK_Flood	1.00	1.00	1.00
DDoS-RSTFINFlood	1.00	1.00	1.00
DDoS-SYN_Flood	1.00	1.00	1.00
DDoS-SlowLoris	0.87	0.93	0.90
DDoS-SynonymousIP_Flood	1.00	1.00	1.00
DDoS-TCP_Flood	1.00	1.00	1.00
DDoS-UDP_Flood	1.00	1.00	1.00
DDoS-UDP_Fragmentation	0.99	0.99	0.99
DNS_Spoofing	0.65	0.60	0.63
DictionaryBruteForce	1.00	0.20	0.33
DoS-HTTP_Flood	0.94	0.97	0.95
DoS-SYN_Flood	1.00	1.00	1.00
DoS-TCP_Flood	1.00	1.00	1.00
DoS-UDP_Flood	1.00	1.00	1.00
MITM-ArpSpoofing	0.84	0.78	0.81
Mirai-greeth_flood	1.00	1.00	1.00
Mirai-greip_flood	0.99	0.99	0.99
Mirai-udpplain	1.00	1.00	1.00
Recon-HostDiscovery	0.79	0.77	0.78
Recon-OSScan	0.65	0.40	0.49
Recon-PingSweep	0.00	0.00	0.00
Recon-PortScan	0.74	0.46	0.57
SqlInjection	0.33	0.14	0.20
Uploading_Attack	1.00	0.50	0.67
VulnerabilityScan	0.87	0.93	0.90
XSS	1.00	0.17	0.29

Figure 4 presents the confusion matrix for the Random Forest model by SMOTE. The matrix demonstrates a more balanced classification across all classes, confirming that SMOTE significantly enhances the model's ability to detect both majority and minority attack types accurately. Overall, these outcomes underscore the crucial role of data balancing in enhancing the fairness, accuracy, and robustness of intrusion detection systems.

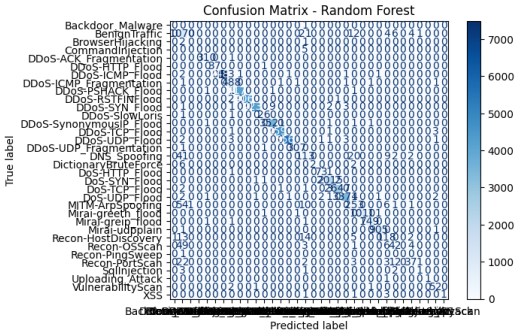


Figure 4. Confusion Matrix – Random Forest by Smote

Table 5. Comparison of Random Forest Performance with and without SMOTE

Metrics	Devoid of SMOTE	Random Forest + SMOTE
Accuracy	0.9898	0.9901
Precision	0.9896	0.9898
Recall	0.9898	0.9901
F1-score	0.9889	0.9896
Cohen's Kappa	0.9888	0.9892

The results presented in Table 5 reveal that, although the performance gains are relatively modest, the Random Forest model enhanced by SMOTE consistently outperforms the baseline across all evaluation metrics. Accuracy improved by 0.9898 to 0.9901, while the F1-score increased by 0.9889 to 0.9896. This balanced improvement in both precision and recall reflects the enhanced robustness and reliability of the model in detecting a wider range of attack types. The increase in Cohen's Kappa by 0.9888 to 0.9892—a statistical measure that evaluates observed accuracy against expected accuracy by chance—indicates an improvement in the model's predictive alignment by the actual labels, even after accounting for random agreement. This enhancement confirms that the application of SMOTE has a positive impact on the model's stability and reliability, particularly in detecting previously underrepresented cyberattack classes.

3.5. Comparative Analysis by Benchmark Studies

To further validate the effectiveness of the proposed Random Forest + SMOTE model, a comparison was conducted by several previous studies that also utilized the CIC-IoT2023 dataset—a widely recognized benchmark in IoT security research due to its comprehensive and diverse attack scenarios. Table 6 presents a concise comparative overview of the studies, highlighting differences in algorithm, dataset, accuracy, and data balancing approaches.

Table 6 further demonstrates that the proposed method achieves the highest accuracy among all compared studies, reaching 99.01%, and significantly outperforms the other models. For instance, a Study [12] A notable work in IoT security, employing CNN and CNN-BiLSTM architectures, achieved an accuracy of 98.00%. Meanwhile, the study [19] proposes a two-tier intrusion detection system (IDS) that uses deep learning models to identify DDoS attacks on IoT networks. Similarly, the Study [17], which explored the effectiveness of Recurrent Neural Networks (RNNs) on both the CIC-IoT2023 and TON_IoT datasets, achieved an accuracy of 96.56%. In contrast, the Study [20] The model, which introduced the DLMIDPSM model, attained a notably lower accuracy of just 85.00%.

What distinguishes the proposed model is its integration with SMOTE, which effectively addresses class imbalance without sacrificing the accuracy of the majority classes. Unlike several previous studies that achieved high accuracy at the expense of increased computational complexity, the Random Forest + SMOTE approach offers both high accuracy and computational efficiency, making it a practical and scalable solution for real-world IoT security applications.

The comparison presented in Table 6 clearly demonstrates the critical role of data balancing techniques, such as SMOTE, in significantly enhancing the detection capabilities of conventional classification models in IoT security. This insight is especially valuable for researchers and practitioners in the field, as it underscores the importance of incorporating such techniques to improve model performance and ensure more equitable detection across all attack classes.

Table 6. Comparison of The Proposed Approach with The Benchmark Study

Study	Dataset	Model	Accuracy	Balancing
[12]	CICIoT2023	CNN, CNN-BiLSTM	98.00%	-
[19]	CICIoT2023	Two Stage-(DNN, CNN, LSTM)	91.27%	-
[17]	CIC-IoT2023, TON_IoT	RNN	96.56%	-
[20]	CICIoT2023	DLMIDPSM	85%	-
Our Model	**CICIoT2023**	**RF + SMOTE**	**99.01%**	**SMOTE**

3.6. Discussion

The findings of this study demonstrate that integrating the Synthetic Minority Oversampling Technique (SMOTE) with the Random Forest algorithm effectively improves the detection of cyberattacks in IoT networks by addressing the severe class imbalance in the CIC-IoT2023 dataset. This outcome is consistent with prior studies highlighting the importance of data balancing in enhancing intrusion detection performance. Oversampling approaches, such as [10], and hybrid balancing methods, like SMOTE-ENN [11], have been shown to increase sensitivity to rare attack classes and improve classification stability. Similarly, [21] confirmed that

effective preprocessing strategies, when integrated with ensemble classifiers, contribute to more robust IDS performance. The present study strengthens these findings by providing empirical validation on the CIC-IoT2023 dataset, where the application of SMOTE enables Random Forest to achieve consistent precision, recall, and F1-scores across 33 attack types.

Beyond addressing imbalance, the results also reaffirm the competitiveness of Random Forest as a classifier in IoT security. Previous studies have shown Random Forest to be robust and interpretable for intrusion detection applications, particularly compared with more complex deep learning algorithms [6, 22]. Although CNNs, RNNs, and hybrid models have gained prominence in intrusion detection [2, 4, 19], their deployment in IoT systems is often limited by computational demands. In contrast, this study demonstrates that Random Forest, when enhanced with SMOTE, achieves state-of-the-art accuracy while remaining computationally efficient, making it a practical and scalable solution for real-world IoT environments.

4. CONCLUSION

This research has demonstrated the effectiveness of combining the Random Forest algorithm with the Synthetic Minority Oversampling Technique (SMOTE) in detecting cyberattacks in Internet of Things (IoT) networks, utilizing the CIC-IoT2023 dataset. First, the study successfully evaluated the performance of SMOTE on a highly imbalanced dataset, confirming its capability to improve the representation of minority attack classes. Second, it implemented and validated the SMOTE-Random Forest algorithm in a multi-class detection scenario, showing that the integration consistently achieved high performance across all 33 attack types. Third, by comparing classification performance metrics before and after the balancing process, the results demonstrated significant improvements in model sensitivity, precision, recall, and F1-score. The integration of SMOTE effectively resolved the significant class imbalance present in the original dataset, allowing the model to identify both majority and minority attack types more accurately. The test results indicate that the proposed model achieved an impressive accuracy of 99.01%, along with consistently high precision, recall, F1-score, and Cohen's Kappa across all 33 attack types. When compared to both the baseline Random Forest model without SMOTE and several recent deep learning-based approaches, the Random Forest model with SMOTE not only improves detection accuracy but also offers superior computational efficiency. As such, this study makes a valuable contribution to the development of reliable and practical intrusion detection systems (IDS) for IoT environments. For future work, future research could explore further enhancements by integrating hybrid balancing techniques, incorporating feature selection methods, and developing real-time, ensemble-based IDS solutions for deployment on edge devices, thereby improving detection reliability in real-world applications.

5. ACKNOWLEDGEMENTS

Thank you to Universitas Lancang Kuning for providing funding for this research.

6. DECLARATIONS

AI USAGE STATEMENT

During the preparation of this work, the authors used ChatGPT (OpenAI) to improve the language and clarity of the manuscript. After using this tool, the authors reviewed and edited the content as needed and took full responsibility for the publication's content.

AUTHOR CONTIBUTION

The first author, Guntoro, contributed to data collection and model creation. Lisnawita, the second author, reviewed the model created. Loneli Costaner, the third author, contributed to writing this article.

FUNDING STATEMENT

This research is funded by the Institute for Research and Community Service, Universitas Lancang Kuning.

COMPETING INTEREST

The author declares that the entire research, analysis, and manuscript preparation process was conducted without any conflict of interest that could affect the academic and scientific integrity of this article.

REFERENCES

[1] L. Babun, K. Denney, Z. B. Celik, P. McDaniel, and A. S. Uluagac, "A survey on IoT platforms: Communication, security, and privacy perspectives," vol. 192, p. 108040, June, 2021, https://doi.org/10.1016/j.comnet.2021.108040.

[2] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," vol. 50, p. 102419, Februari, 2020, https://doi.org/10.1016/j.jisa.2019.102419.

[3] C. M. Patterson, J. R. C. Nurse, and V. N. L. Franqueira, "Learning from cyber security incidents: A systematic review and future research agenda," *Computers and Security*, vol. 132, p. 103309, September, 2023, https://doi.org/10.1016/j.cose.2023.103309.

[4] I. Cvitic, D. Perakovic, B. B. Gupta, and K.-K. R. Choo, "Boosting-Based DDoS Detection in Internet of Things Systems," vol. 9, no. 3, pp. 2109–2123, Februari, 2022, https://doi.org/10.1109/JIOT.2021.3090909.

[5] O. Zorlu and A. Ozsoy, "A blockchain-based secure framework for data management," vol. 18, no. 10, pp. 628–653, June, 2024, https://doi.org/10.1049/cmu2.12781.

[6] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," vol. 32, no. 1, p. e4150, Januari, 2021, https://doi.org/10.1002/ett.4150.

[7] F.-S. Zamfir, M. Carbureanu, and S. F. Mihalache, "Application of Machine Learning Models in Optimizing Wastewater Treatment Processes: A Review," vol. 15, no. 15, p. 8360, July, 2025, https://doi.org/10.3390/app15158360.

[8] R. Ahsan, W. Shi, and J. Corriveau, "Network intrusion detection using machine learning approaches: Addressing data imbalance," vol. 7, no. 1, pp. 30–39, March, 2022, https://doi.org/10.1049/cps2.12013.

[9] P. Kaliyaperumal, S. Periyasamy, M. Thirumalaisamy, B. Balusamy, and F. Benedetto, "A Novel Hybrid Unsupervised Learning Approach for Enhanced Cybersecurity in the IoT," vol. 16, no. 7, p. 253, July, 2024, https://doi.org/10.3390/fi16070253.

[10] A. S. Tarawneh, A. B. A. Hassanat, K. Almohammadi, D. Chetverikov, and C. Bellinger, "SMOTEFUNA: Synthetic Minority Over-Sampling Technique Based on Furthest Neighbour Algorithm," vol. 8, pp. 59 069–59 082, March, 2020, https://doi.org/10.1109/ACCESS.2020.2983003.

[11] F. Omer Albasheer, R. Ramesh Haibatti, M. Agarwal, and S. Yeob Nam, "A Novel IDS Based on Jaya Optimizer and Smote-ENN for Cyberattacks Detection," vol. 12, pp. 101 506–101 527, July, 2024, https://doi.org/10.1109/ACCESS.2024.3431534.

[12] H. Q. Gheni and W. L. Al-Yaseen, "Two-step data clustering for improved intrusion detection system using CICIoT2023 dataset," *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, vol. 9, p. 100673, September, 2024, https://doi.org/10.1016/j.prime.2024.100673.

[13] J. Li, M. S. Othman, H. Chen, and L. M. Yusuf, "Optimizing IoT intrusion detection system: Feature selection versus feature extraction in machine learning," vol. 11, no. 1, p. 36, Februari, 2024, https://doi.org/10.1186/s40537-024-00892-y.

[14] S. Chen and W. Zheng, "RRMSE-enhanced weighted voting regressor for improved ensemble regression," vol. 20, no. 3, p. e0319515, March, 2025, https://doi.org/10.1371/journal.pone.0319515.

[15] X. Larriva-Novo, V. A. Villagrá, M. Vega-Barbas, D. Rivera, and M. Sanz Rodrigo, "An IoT-Focused Intrusion Detection System Approach Based on Preprocessing Characterization for Cybersecurity Datasets," vol. 21, no. 2, p. 656, Januari, 2021, https://doi.org/10.3390/s21020656.

[16] F. Bolikulov, R. Nasimov, A. Rashidov, F. Akhmedov, and Y.-I. Cho, "Effective Methods of Categorical Data Encoding for Artificial Intelligence Algorithms," vol. 12, no. 16, p. 2553, August, 2024, https://doi.org/10.3390/math12162553.

[17] S. Abbas, I. Bouazzi, S. Ojo, A. Al Hejaili, G. A. Sampedro, A. Almadhor, and M. Gregus, "Evaluating deep learning variants for cyber-attacks detection and multi-class classification in IoT networks," vol. 10, p. e1793, Januari, 2024, https://doi.org/10.7717/peerj-cs.1793.

[18] A. Alamleh, O. S. Albahri, A. A. Zaidan, A. S. Albahri, A. H. Alamoodi, B. B. Zaidan, S. Qahtan, H. A. Alsatar, M. S. Al-Samarraay, and A. N. Jasim, "Federated Learning for IoMT Applications: A Standardization and Benchmarking Framework of Intrusion Detection Systems," vol. 27, no. 2, pp. 878–887, Februari, 2023, https://doi.org/10.1109/JBHI.2022.3167256.

[19] S. Hizal, U. Cavusoglu, and D. Akgun, "A novel deep learning-based intrusion detection system for IoT DDoS security," vol. 28, p. 101336, December, 2024, https://doi.org/10.1016/j.iot.2024.101336.

[20] S. K. Erskine, "Real-Time Large-Scale Intrusion Detection and Prevention System (IDPS) CICIoT Dataset Traffic Assessment Based on Deep Learning," vol. 8, no. 2, p. 52, April, 2025, https://doi.org/10.3390/asi8020052.

[21] T. S. Naseri and F. S. Gharehchopogh, "A Feature Selection Based on the Farmland Fertility Algorithm for Improved Intrusion Detection Systems," vol. 30, no. 3, p. 40, July, 2022, https://doi.org/10.1007/s10922-022-09653-9.

[22] A. Hussain, K. Naseer Qureshi, K. Javeed, and M. Alhussein, "An Enhanced Intelligent Intrusion Detection System to Secure E-Commerce Communication Systems," vol. 47, no. 2, pp. 2513–2528, 2023, https://doi.org/10.32604/csse.2023.040305.

[This page is intentionally left blank.]