

A comprehensive evaluation of multiclass imbalance techniques with ensemble models in IoT environments

Januar Al Amien¹, Hadhrami Ab Ghani², Nurul Izrin Md Saleh³, Soni¹, Yulia Fatma¹, Regiolina Hayami¹

¹Department of Informatics Engineering, Faculty Computer Science, University Muhammadiyah Riau, Riau, Indonesia

²Department of Data Science, Faculty of Data Science and Computing, Universiti Malaysia Kelantan, Kelantan, Malaysia

³Department of Software Engineering, Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia

Article Info

Article history:

Received Nov 30, 2023

Revised Jan 8, 2024

Accepted Feb 7, 2024

Keywords:

Imbalanced ratio

Internet of things

Intrusion detection system

Machine learning

Multiclass

Oversampling

Weighted extreme gradient boosting

ABSTRACT

The internet of things (IoT) has revolutionized connectivity and introduced significant security challenges. In this context, intrusion detection systems (IDS) play a crucial role in detecting attacks in IoT environments. Bot-IoT datasets often face class imbalance issues, with the attack class having significantly more samples than the normal class. Addressing this imbalance is essential to enhance IDS performance. The study evaluates various techniques, including imbalance ratio techniques we call imbalance ratio formula (IRF) for controlling imbalance data, while also testing IRF to compare it with oversampling techniques like synthetic minority oversampling technique (SMOTE) and adaptive synthetic sampling (ADASYN). This research also incorporates the extreme gradient boosting (XGBoost) ensemble model approach to improve IDS performance in dealing with multiclass imbalance issues in Bot-IoT datasets. Through in-depth analysis, we identify the strengths and weaknesses of each method. This study aims to guide researchers and practitioners working on IDS in high-risk IoT environments. The proposed IRF, when integrated with the XGBoost algorithm has been demonstrated to achieve comparable accuracy of 99.9993% while reducing the training time to be on average at least two times faster than those achieved by the other state-of-the-art ensemble methods.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Januar Al Amien

Department of Informatics Engineering, Faculty Computer Science, University Muhammadiyah Riau
Riau, Indonesia

Email: januaralamien@umri.ac.id

1. INTRODUCTION

The internet of things (IoT) has brought about a profound transformation in various sectors, providing limitless connectivity between devices and data [1]. However, the benefits of the IoT revolution also come with significant security challenges [2]. Intrusion detection systems (IDS) play a crucial role in preserving the integrity and security of the IoT environment by detecting potential attacks [3]. The pervasive influence of information technology has magnified the role of networks in our daily lives, affecting various facets of society. Consequently, network security has become an intricate challenge as technology continues to proliferate [4]. The security landscape is not limited to fundamental components like vertical encryption; it also encompasses the monitoring of potential intruders within network traffic [5]. This expansion, driven by the widespread adoption of technology, bestows substantial benefits upon end-users, including time and

effort savings. In this context, the IDS emerges as a critical element capable of identifying and flagging suspicious activities within a network or system. When the IDS detects questionable behaviors associated with network traffic, it promptly dispatches alerts to network or system administrators [6]. Notably, the Bot-IoT dataset, the latest addition to intrusion detection datasets designed for IoT environments, has been meticulously crafted at the University of New South Wales (UNSW) Canberra Cyber Range Lab to create realistic testing environments for IoT scenarios. This dataset encompasses a range of attack types, including distributed denial of service (DDoS), denial of service (DoS), reconnaissance, normal, and theft, reflecting the evolving landscape of network security challenges [7].

IDS is used to detect hacking activities in computer systems and detect suspicious incoming and outgoing network traffic activities [8]. Many researchers use datasets for this IDS such as knowledge discovery in database-(KDD-CUP'99) [9], where this dataset is built based on data that has been taken in the DARPA'98 IDS evaluation program. Network security laboratory-knowledge discovery and data mining (NSL KDD) dataset [10] this dataset is a collaboration of several researchers in the KDD project and tries to improve the shortcomings of the KDD CUP99 dataset. UNSW-NB15 dataset [11], this dataset was developed by a team of researchers at the Australian center for cyber security (ACCS) within the UNSW. The Bot-IoT dataset, referenced in this study, represents an updated version that succeeds the UNSW-NB15 dataset, reflecting its more recent nature [12].

Addressing the complexity of multiclass imbalanced data, particularly in the Bot-IoT dataset, presents a critical challenge in IDS development. Various approaches have been explored, with the imbalance ratio technique gaining attention for its focus on balancing majority and minority classes through adjustments based on a calculated ratio. The suggested utilization of the imbalance ratio formula (IRF) aims to impart greater weight to classes with fewer instances during model training, specifically addressing the challenges in the Bot-IoT dataset [13]. Complementing this technique, oversampling methods synthetic minority oversampling technique (SMOTE) and adaptive synthetic sampling (ADASYN) have demonstrated efficacy in improving minority class representation. This research significantly contributes to managing multiclass imbalance by integrating the imbalance ratio technique and oversampling methods, providing valuable insights for IDS development.

- a. A comprehensive evaluation of the IRF technique, oversampling methods (SMOTE and ADASYN), and the extreme gradient boosting (XGBoost) ensemble model in a multiclass scenario to enhance IDS performance.
- b. A new class IRF controls the weights of each class, emphasizing more on classes with fewer instances during model training.
- c. A collaborative model for intrusion detection by integrating the proposed class IRF with XGBoost ensemble model for IoT environment with imbalance data.

This paper consists of four main sections: section 2 provides a related work on IDS and imbalance ratio, with a discussion on techniques to solve it. Section 3 outlines the method. Section 4 result and discussion presents the experimental results, including the evaluation metrics used. Finally, section 5 summarizes the study's findings, outlines limitations and future research directions, and highlights the practical implications of the proposed method.

2. RELATED WORK

IDS play a crucial role in identifying and preventing unauthorized access to computer networks. Class imbalance, characterized by a disproportionate number of normal instances compared to intrusion instances, poses a persistent challenge in the effectiveness of these systems. Researchers have actively investigated and implemented diverse strategies, with a notable emphasis on leveraging imbalance ratio techniques to enhance the accuracy and reliability of IDS in detecting potential security threats.

In parallel, oversampling methods have been widely employed to augment the representation of minority classes in the dataset. According to Popoola *et al.* [14] an algorithm for botnet attack detection based on deep learning that efficiently handles highly imbalanced network traffic data by utilizing the SMOTE to balance classes and employing the deep recurrent neural network (DRNN) for learning discriminative features from the balanced data. SMOTE is utilized to generate synthetic samples for the positive class with the aim of achieving a balanced training dataset [15]. However SMOTE is generally exposed to overfitting problems as it tempts to produce similar samples to existing instances. To address this issue and push the boundaries of current practices, this paper introduces a new oversampling technique called SMOTE-NaN-DE, which combines SMOTE-based synthetic sample generation with a natural neighbor-based error detection method to enhance class-imbalanced data [16]. Another algorithm proposed in literature that offers improved technique to cater imbalance problem without major overfitting effects is ADASYN, which has implemented to address the imbalanced datasets such as in NSL-KDD [17]. Using ADASYN balances sample distribution, preventing the model from favoring large samples and neglecting smaller ones

[18]. An IDS utilizing ADASYN oversampling and LightGBM is evaluated on the NSL-KDD, UNSW-NB15, and CICIDS2017 datasets [19], however ADASYN is computationally more extensive as compared to SMOTE. Furthermore, there are other studies focused on multiclass scenarios as well, such as in [20]. This study introduces and assesses an IDS model based on recurrent neural network (RNN), long short-term memory (LSTM), and gated recurrent unit (GRU), BotIDS, using a specific Bot-IoT dataset, which demonstrates promising results with a validation accuracy of 99.94%, a validation loss of 0.58%, and a prediction execution time under 0.34 ms. However, this proposed scheme also suffers from relatively higher computational complexity.

The imbalance ratio technique, as detailed by [13], utilizes a specific IRF to mitigate class imbalance within datasets. Additionally, research exemplified by [21] underscores the primary objective of imbalance ratio techniques, which is to alleviate bias towards the majority class, thereby enhancing system efficiency and average accuracy in the realm of imbalanced datasets. The application of these approaches has shown promising outcomes in elevating the overall performance of IDS when confronted with class imbalance scenarios.

Despite the notable progress made in prior research to tackle class imbalance in IDS, there exists an unexplored potential for comprehensive evaluations that integrate both imbalance ratio techniques and oversampling methods, harnessing the capabilities of XGBoost. This study seeks to fill this gap by conducting a thorough analysis of the strengths and weaknesses associated with the combined application of these approaches. The aim is to contribute valuable insights to the enhancement of IDS capabilities, particularly in navigating complex and high-risk environments, thus advancing the field's understanding and effectiveness in addressing security challenges.

3. METHOD

In response to the security challenges prevalent in the IoT environment, we present a tailored method designed explicitly for intrusion detection. Our proposed approach involves the implementation of an IDS customized to accommodate the unique characteristics of IoT. This method is strategically crafted to offer robust protection against the evolving and intricate security threats that arise within the dynamic IoT ecosystem. By addressing the specific challenges of the IoT landscape, our proposed methodology aims to enhance the overall security posture and resilience of IoT devices and systems.

In Figure 1, the implementation of the analysis using the Bot-IoT dataset is depicted, involving a series of preprocessing steps, including data cleaning, data transformation, and feature engineering. To address the issue of class imbalance, the application of oversampling techniques using ADASYN and SMOTE is employed, along with the utilization of class weight techniques to enhance the role of minority classes in the model. Data is then partitioned with an 80% allocation for training and 20% for testing, while the XGBoost model is chosen as the primary algorithm. The results of the analysis are evaluated using a confusion matrix, providing an in-depth insight into the model's ability to handle class imbalance in the Bot-IoT dataset.

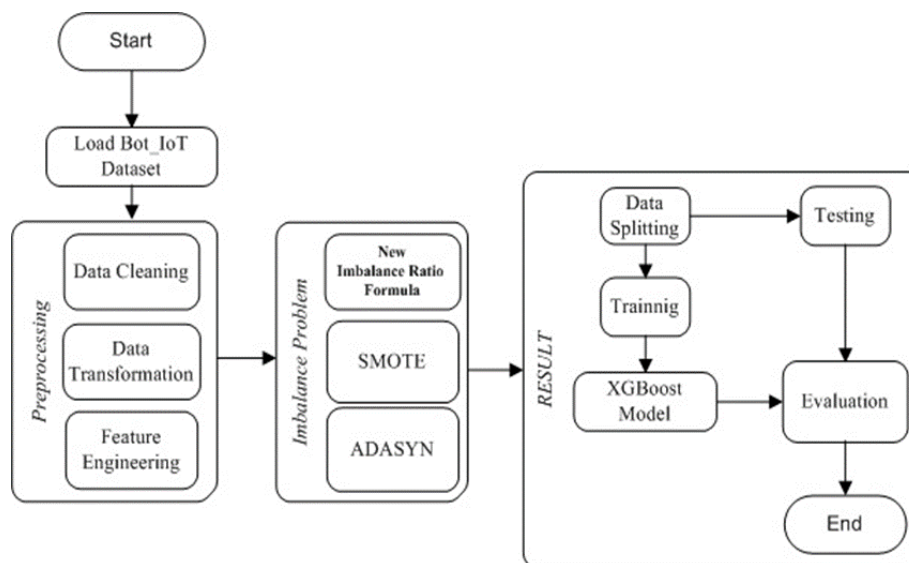


Figure 1. Process of constructing the model

3.1. Bot-IoT dataset preprocessing

The Bot-IoT dataset encompasses 46 network traffic features and incorporates three distinct label categories tailored for binary, 5-class, and 11-class classifications. A comprehensive description of these features and labels can be found in [15]. However, in the course of our analysis, it was discerned that only 37 out of the 46 features proved relevant for the identification of botnet attacks within IoT networks. Specifically, 'pkSeqID,' 'saddr,' 'daddr,' 'proto,' 'state,' 'flgs,' 'sport,' 'dport,' and 'subcategory' were notably excluded from the analysis.

The dataset presented in Table 1 consists of various classes, with a total accumulation of 3,659,522 data entries. The DDoS class is the largest class with 1,926,624 entries, reflecting the most common DDoS attacks. The DoS class, with 1,650,260 entries, signifies a significant DoS attack presence. The Reconnaissance class has 91,082 entries, indicating network scanning or reconnaissance activities commonly associated with attack planning. The normal class has the smallest count with only 477 entries, encompassing non-suspicious network traffic. Lastly, the Theft class has 79 entries, suggesting rare attempts at data or information theft. Data analysis in this context will aid in developing an effective IDS for identifying and safeguarding against various types of attacks.

Table 1. Distribution of multiclass target category frequencies in the dataset

No	Category	Class number
1	DDoS	1926624
2	DoS	1650260
3	Reconnaissance	91082
4	Normal	477
5	Theft	79
Total		3,659,522

The subcategory distribution in Table 2 includes a variety of classes, with a cumulative total of 3,659,522 data entries. The user datagram protocol (UDP) subcategory is the largest, with 1,981,230 entries, indicating the prevalence of UDP-based network traffic. The TCP subcategory follows with 1,593,180 entries, representing TCP-based traffic. Service_Scan has 73,168 entries, suggesting network scanning or service enumeration activities. OS_Fingerprint includes 17,914 entries, signifying efforts to identify the target system's operating system. HTTP has 2,474 entries, and highlighting web-related traffic. Normal comprises 477 entries and reflecting non-suspicious network activities. Keylogging has 73 entries, implying instances of keylogger-based attacks. Data_Exfiltration is the smallest subcategory, with only 6 entries, denoting rare attempts at data exfiltration. Analysis of these subcategories is crucial for developing a comprehensive IDS capable of identifying and mitigating various threat types.

Table 2. Distribution of multiclass target subcategory frequencies in the dataset

No	Subcategory	Class number
1	UDP	1981230
2	TCP	1593180
3	Service_Scan	73168
4	OS_fingerprint	17914
5	HTTP	2474
6	Normal	477
7	Keylogging	73
8	Data_exfiltration	6
Total		3,659,522

The combined dataset presented in Table 3 comprises various classes and subcategories, with a total accumulation of 3,659,522 data entries. These entries encompass both category and subcategory labels, reflecting the diversity of network traffic and activities. The categories include DoS-UDP, DDoS-TCP, DDoS-UDP, DoS-TCP, reconnaissance-service_Scan, reconnaissance-OS_Fingerprint, DoS-HTTP, DDoS-HTTP, normal-normal, theft-keylogging, and Theft-Data_Exfiltration [22], [23]. The combined analysis of these categories and subcategories is essential for the development of an effective IDS capable of accurately identifying and mitigating a wide range of potential threats in network traffic data.

Table 3. Combined dataset class and subcategory distribution

No	Label	Class number
1	DoS-UDP	1032975
2	DDoS-TCP	977380
3	DDoS-UDP	948255
4	DoS-TCP	615800
5	Reconnaissance-Service_scan	73168
6	Reconnaissance-OS_Fingerprint	17914
7	DoS-HTTP	1485
8	DDoS-HTTP	989
9	Normal-normal	477
10	Theft-keylogging	73
11	Theft-Data_Exfiltration	6
Total		3,659,522

3.2. A new imbalance ratio approach

In order to extend and improve the previous method proposed in binary classification [12], this paper has proposed a multi-class classification approach with a new imbalance ratio approach. The previous study was related to binary classification problems, where the goal was to address the differences between two classes. In this study, we have expanded its scope to explore multiclass classification issues. The steps to calculate the IRF for a dataset are:

- Find the number of samples in each class.
- For each class i , calculate the number of samples in the majority class (N_i) and the number of samples in the minority class (n_i).
- Calculate the imbalance ratio (IR_i) for each class i as (1):

$$IR_i = N_i/n_i \quad (1)$$

- Calculate the IRF value for the dataset as the maximum imbalance ratio across all classes:

$$IRF = \max(IR_1, IR_2, \dots, IR_k) \quad (2)$$

- Calculate the average of the values obtained in step b.
- Return the result.

Where k is the total number of classes in the dataset.

The algorithm used to calculate the Fisher's imbalance ratio IRF for a given dataset. This involves several steps: IRF, determining the number of samples in each class, then calculating the number of samples in both the majority class (N_i) and the minority class (n_i) for each class (i). The imbalance ratio (IR_i) for each class is then computed by dividing N_i by n_i . The IRF value for the entire dataset is determined by selecting the maximum imbalance ratio across all classes. Subsequently, the average of the values obtained in the second step is calculated, and the result is returned. This algorithm is applicable to datasets with k total classes.

3.3. Synthetic minority oversampling technique

SMOTE, is a method for addressing class imbalance by augmenting the minority class. This is achieved through the generation of synthetic data points created by replicating and modifying existing minority class instances [24]. In the SMOTE process, the over-sampling is carried out by identifying the k -nearest neighbors for each minority class instance and then generating synthetic instances that interpolate between the selected instance and its neighbors. This approach is advantageous as it helps mitigate excessive overfitting issues that may arise when simply replicating minority class instances. By adjusting the class distribution in this manner, SMOTE increases the amount of data in the minority class, making it a valuable tool for enhancing the performance of machine learning models on imbalanced datasets.

3.4. Adaptive synthetic sampling

ADASYN is grounded in the concept of adaptively sampling data for minority classes while considering the distribution of these classes. This approach allows ADASYN to dynamically produce synthetic data samples for minority classes, mitigating bias stemming from unbalanced data distribution [24]. Addressing the issue of imbalanced data can be achieved through various methods, with one effective approach being the utilization of the ADASYN method. This technique intelligently generates additional data points, particularly for the underrepresented classes, resulting in a more equitable dataset for machine learning models.

3.5. XGBoost

XGBoost [25], is a powerful and versatile machine learning algorithm that has gained widespread popularity in the data science and predictive modeling community. XGBoost is an ensemble learning method that is primarily used for regression and classification tasks. It excels at enhancing the predictive accuracy of models by combining the predictions from multiple weak learners (usually decision trees) to form a more robust and accurate final prediction. One of XGBoost's key strengths lies in its ability to handle complex datasets with a high degree of accuracy, making it a valuable tool for a wide range of applications, including image and text classification, as well as structured data analysis. XGBoost is known for its efficiency and scalability, often outperforming other algorithms in both speed and accuracy. Its adaptive gradient boosting approach, regularization techniques, and parallel processing capabilities make it a popular choice among data scientists for tackling various machine learning challenges.

$$\mathcal{L}^{(t)} = \sum_{i=1}^n I(\hat{y}_i, y_i^{(t-1)} + f_t(x_i)) + \Omega(f_t), \quad (3)$$

$$\mathcal{L}^{(t)} = \sum_{i=1}^n [l(\hat{y}_i, y_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t), \quad (4)$$

$$\mathcal{L}^{(t)} = \sum_{i=1}^n (g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)) + \Omega(f_t), \quad (5)$$

$$w_j^* = -\frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda}, \quad (6)$$

$$\mathcal{L}^{(t)}(q) = -\frac{1}{2} \sum_{j=1}^T \frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T, \quad (7)$$

$$\mathcal{L}_{split} = \frac{1}{2} \left[\frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{(\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda} \right] - \gamma, \quad (8)$$

The formula for the XGBoost model represents the objective function used in the training process, which aims to minimize the loss and construct a robust ensemble model. In this equation, $\mathcal{L}^{(t)}$ is the cumulative loss at a specific iteration (t), and it is composed of three main components. The IRFst component quantifies the difference between the current predicted values \hat{y} and the updated values from the previous iteration $y_i^{(t-1)}$ based on the model's predictions f_t . The second component introduces regularization terms (Ω) to control the complexity of the model and avoid overfitting. The third component combines gradient (g) and Hessian (h) statistics to guide the construction of individual trees in the ensemble. The optimal weights w_j^* for each leaf node are calculated, and the overall loss $\mathcal{L}^{(t)}(q)$ is computed based on these weights and other hyperparameters, such as the regularization term λ and a shrinkage factor γ , where I_L, I_R and I_j are the sets of instances of the left, right and j -th leaf respectively. Additionally, a splitting criterion \mathcal{L}_{split} is employed to determine the optimal tree structure while minimizing the overall loss. These equations demonstrate the mathematical foundation behind the XGBoost algorithm, which efficiently combines decision trees into a powerful ensemble model for predictive tasks.

3.6. Evaluating confusion matrix

According to Liu *et al.* [26], a confusion matrix is a fundamental tool in evaluating the performance of a classification model. It provides a comprehensive summary of the model's predictions by categorizing them into four different outcomes: true positives (correctly predicted positive instances), true negatives (correctly predicted negative instances), false positives (incorrectly predicted positive instances), and false negatives (incorrectly predicted negative instances). These metrics offer valuable insights into the model's performance, beyond accuracy alone, particularly in scenarios with imbalanced datasets where one class significantly outnumbers the other. Among these metrics, precision holds a special significance. It measures the reliability of the model's positive predictions and represents the proportion of true positive predictions relative to all instances classified as positive, regardless of their actual correctness. A high precision value indicates a system's ability to make trustworthy positive predictions, making it a critical metric to optimize in situations where the cost of false positives is high, and reliability is paramount. The model's performance was evaluated using the confusion matrix, accuracy, precision, recall, and F1 score for binary classification, and recall and false positive rate (FPR) for multiclass classification [8].

$$\text{Precision} = \frac{TP}{TP+FP} \quad (9)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (10)$$

$$F1 = 2 * \frac{\text{Presisi} * \text{Recall}}{\text{Presisi} + \text{Recall}} \quad (11)$$

$$FNR = \frac{FN}{TP+FN} \quad (12)$$

$$PR = \frac{FP}{FP+TN} \quad (13)$$

$$\text{Accuracy} = \frac{TP+TN}{TP+FN+FP+TN} \quad (14)$$

These formulas represent essential performance metrics for evaluating classification models:

- Precision: it measures the accuracy of positive predictions, specifically the ratio of true positives (correct positive predictions) to the total instances classified as positive. A high precision indicates fewer false positives.
- Recall: it gauges the model's ability to capture all positive instances. It's calculated as the ratio of true positives to the sum of true positives and false negatives.
- F1 Score: this is a harmonic mean of precision and recall, providing a balanced measure of a model's overall performance.
- False negative rate (FNR): it quantifies the ratio of false negatives (positive instances incorrectly predicted as negative) to the total actual positive instances.
- FPR or precision of negative predictions: it measures the ratio of false positives (negative instances incorrectly predicted as positive) to the total actual negative instances [27].

Accuracy: it represents the overall correctness of a model by calculating the ratio of correct predictions (true positives and true negatives) to the total number of predictions made.

4. RESULTS AND DISCUSSION

4.1. Hardware specifications

This section presents details regarding the hardware configuration employed in the experiments. The system utilized for the study was equipped with an Intel(R) Xeon(R) Gold 6134 CPU operating at 3.20 GHz, complemented by 65536 MB of RAM. The system operated within a virtual machine environment, employing the Anaconda application running Python version 3.9.16. Graphic experiments were executed on the Anaconda platform, leveraging various Python libraries.

4.2. Feature engineering

The process of feature engineering in the IDS dataset involves the transformation and adjustment of attributes to enhance the modeling and intrusion detection capabilities. Feature engineering encompasses various operations such as modifying, eliminating, or combining features, and even introducing new ones based on domain expertise or a deeper understanding of the problem. In this context, features categorized as Object type, including pkSeqID, flgs, proto, saddr, sport, daddr, dport, state, attack, category, and subcategory, are removed to streamline the dataset. Following this pruning, the classification now relies on a reduced set of 36 features. Additionally, the remaining categorical attributes will undergo Label Encoding for their use in the classification process.

4.3. Class imbalance handling

The Table 4 reflects efforts to address class imbalance through the implementation of three methods, namely IRF, SMOTE, and ADASYN. The "Class number" column provides insights into the distribution of sample numbers in each class, while the "IRF" column evaluates the performance of the IRF method in handling class imbalance. The IRF calculation uses the formula ($IR_i = N_i/n_i$), which represents the ratio between the number of samples in the majority class (N_i) and the minority class (n_i) for each class. The final IRF for the entire dataset is obtained by taking the maximum value from all imbalance ratios (IR_i). Thus, this table offers a comprehensive view of the effectiveness of various methods in addressing class imbalance, with a specific focus on the performance of the IRF method.

Table 4. Class imbalance handling IRF and oversampling results for label

No	Label	Class number	IRF	SMOTE	ADASYN
1	DoS-UDP	1032975	0,718422	1032975	1032975
2	DDoS-TCP	977380	0,733577	1032975	1033012
3	DDoS-UDP	948255	0,741516	1032975	1033006
4	DoS-TCP	615800	0,832139	1032975	1032993
5	Reconnaissance-Service_Scan	73168	0,980055	1032975	1033005
6	Reconnaissance-OS_Fingerprint	17914	0,995117	1032975	1032994
7	DoS-HTTP	1485	0,999595	1032975	1032973
8	DDoS-HTTP	989	0,999730	1032975	1032979
9	Normal-normal	477	0,999870	1032975	1032980
10	Theft-keylogging	73	0,999980	1032975	1032973
11	Theft-Data_Exfiltration	6	0,999998	1032975	1032976

4.4. Utilizing XGBoost models and performance evaluation

In this study, we utilized the Bot-IoT full dataset, which employs label encoding to convert the "category" and "label" columns into numerical values. We also applied the StandardScaler method to standardize the feature values. This dataset is then divided into training and testing sets, with 20% of the data allocated for testing, all while preserving a constant random state to ensure consistent reproducibility.

In our predictive modeling, we harness the power of the XGBoost algorithm, a robust and versatile tool known for its excellent performance in classification tasks. The XGBoost model allows us to achieve high predictive accuracy and generalization. To fine-tune the model's performance and adapt it to our specific dataset, we've employed various hyperparameters. Among these parameters, 'max_depth' plays a crucial role in controlling the depth of the individual decision trees within the ensemble, helping to balance model complexity and overfitting. Additionally Dhaliwal *et al.* [28], 'learning_rate' influences the contribution of each tree to the final prediction, while 'subsample' and 'colsample_bytree' regulate the sampling of training data and features, respectively. The 'gamma' parameter serves as a regularization term, controlling the reduction in loss required to make a further partition on a leaf node. Careful tuning of these parameters allows us to optimize the XGBoost model's performance and tailor it to the unique characteristics of our classification task. Furthermore, in the context of our dataset, we have applied the 'scale_pos_weight' parameter IRF calculation to further adapt the model to the imbalanced class distribution, ensuring that it effectively addresses the class imbalance challenge.

Table 5 presents a comprehensive breakdown of specific performance metrics for various methodologies aimed at mitigating the imbalanced data issue, encompassing Figure 2(a) IRF, Figure 2(b) SMOTE label, and Figure 2(c) ADASYN (see in Appendix). The metrics include accuracy, precision, recall, F1 score, and training time. Subsequently, Figure 2 visually compares the multiclass classification outcomes of these methodologies across six distinct scenarios, each configured with unique settings. The graphical representation emphasizes the efficacy of these methodologies in diverse contexts, accentuating noticeable distinctions in training times.

Table 5. Imbalanced problem: IRF, oversampling, and XGBoost results

No	Label	Accuracy	Precision	Recall	f1_score	Time (sec)
1	IRF XGBOOST label	0.999993	0.999993	0.999993	0.999993	540
2	SMOTE XGBOOST label	0.999999	0.999999	0.999999	0.999999	1494
3	ADASYN XGBOOST label	0.999999	0.999999	0.999999	0.999999	1716

The findings suggest that the integration of oversampling techniques such as SMOTE and ADASYN in conjunction with the XGBoost algorithm results in a successful resolution of the imbalanced data challenge across various contexts. It is imperative to highlight that the training duration varies based on the selected approach, with certain techniques necessitating prolonged training times while still yielding remarkable results. This underscores the adaptability and robustness of these combined methodologies in mitigating challenges associated with imbalanced data. Furthermore, IRF distinguishes itself for its efficiency, not only effectively addressing imbalanced data but also reducing classification time to less than 300 seconds, which is approximately at least twice as fast as those recorded when utilizing other conventional oversampling algorithms. This characteristic underscores the effectiveness of IRF as a solution for enhancing risk analysis across diverse contexts, ensuring both accuracy and temporal efficiency.

Table 6 presents a performance comparison among various methods, including IRF XGBOOST label, convolutional neural network (CNN) referenced from [20], RNN referenced from [29], and LS-DRNN referenced from [23], specifically in the context of imbalanced classes. IRF XGBOOST label exhibits exceptionally high results with an accuracy, precision, recall, and F1 score all around 0.999993, while requiring approximately 540 seconds for training. The CNN model, cited from [20], achieves a high accuracy of 0.99935 with a training

time of 1395 seconds. The RNN, cited from [29], shows an accuracy of 0.9820 without detailed precision, recall, and F1 score provided. On the other hand, LS-DRNN, referenced from [23], attains an accuracy of 0.9993 with precision 0.9687, recall 0.9975, F1 score 0.9822, and a training time of approximately 616.96 seconds.

Table 6. Performance comparison across 11 classes

Label	Accuracy	Precision	Recall	f1_score	Time (sec)
CNN [20]	0.99935	-	-	-	1395
RNN [29]	0.9820	-	-	-	-
LS-DRNN [23]	0.9993	0.9687	0.9975	0.9822	616.96
IRF XGBOOST	0.999993	0.999993	0.999993	0.999993	540

5. CONCLUSION

This study provides results that illustrate the comparison between the IRF approach and oversampling techniques, specifically SMOTE and ADASYN, in addressing class imbalance in Bot-IoT datasets. Evaluation results indicate that IRF significantly enhances the performance of IDS. Compared to oversampling techniques, IRF proves to be more effective in addressing data imbalance issues. Although oversampling, particularly using SMOTE and ADASYN, also yields improvements, IRF demonstrates superiority in terms of effectiveness. The integration of IRF with the XGBoost ensemble model produces excellent results, showcasing potential adaptability and higher robustness in enhancing IDS performance in IoT environments with imbalanced data. As a direction for future research, it is recommended to further explore the application of the imbalance ratio technique in various other IoT dataset imbalance scenarios to comprehensively understand its performance and applicability.

ACKNOWLEDGEMENTS

The authors wish to express their gratitude for the support and assistance provided by Universitas Muhammadiyah Riau, Indonesia, and Universiti Malaysia Kelantan, Malaysia, in making this article possible. Additionally, the authors extend their appreciation to fellow researchers who contributed, either formally or informally, to the preparation of this paper.

APPENDIX

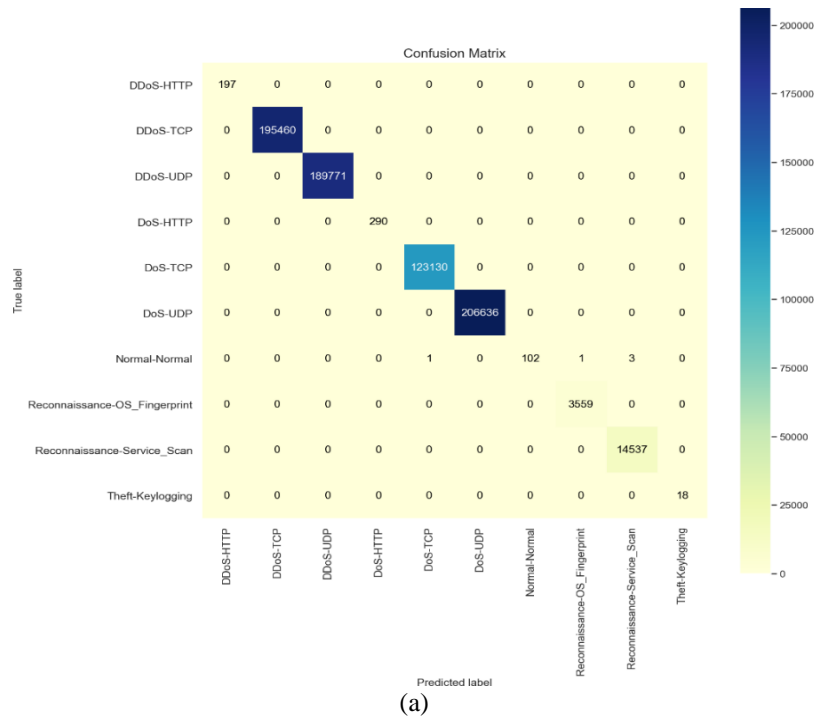


Figure 2. Comparing multiclass classification outcomes across class types: (a) IRF

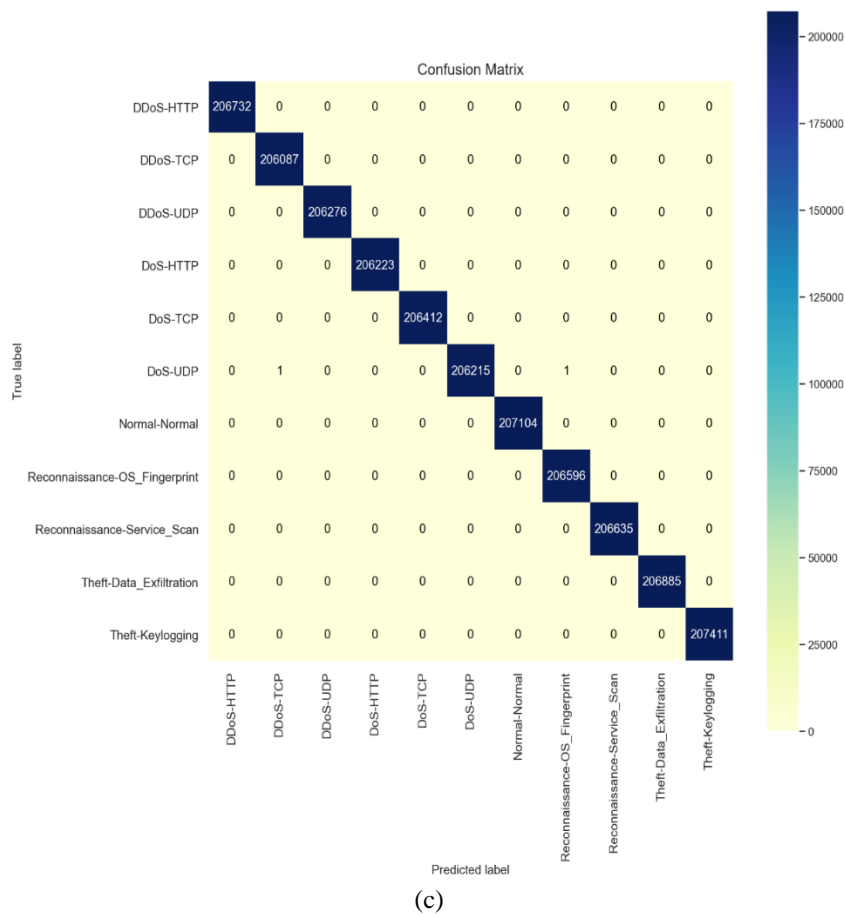
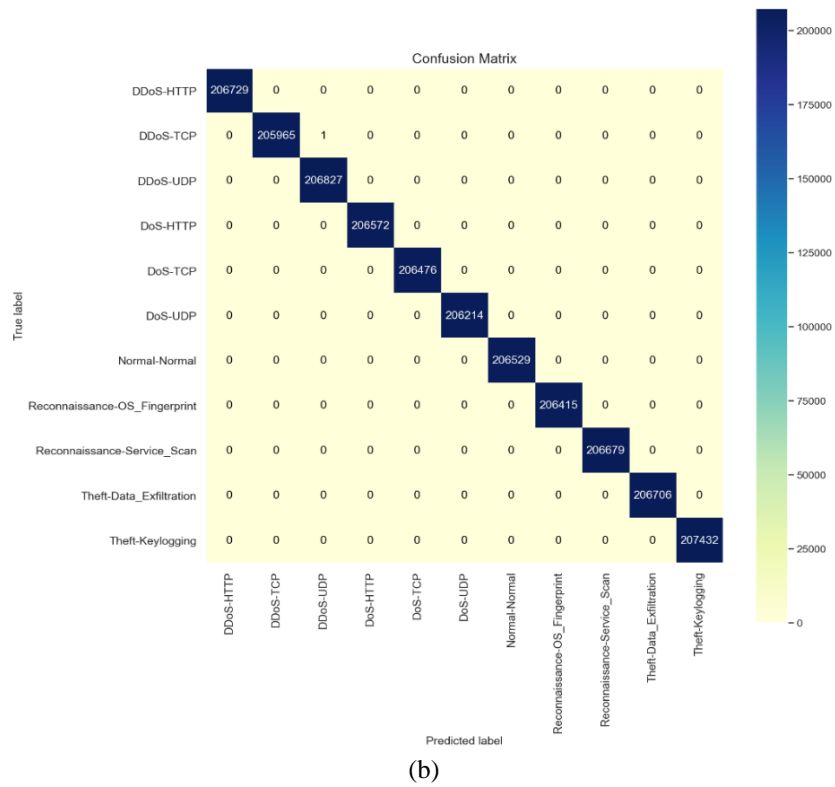





Figure 2. Comparing multiclass classification outcomes across class types: (b) SMOTE and (c) ADASYN (continue)

REFERENCES




- [1] N. Koroniotis, N. Moustafa, F. Schiliro, P. Gauravaram, and H. Janicke, "A Holistic Review of Cybersecurity and Reliability Perspectives in Smart Airports," *IEEE Access*, vol. 8, pp. 209802–209834, 2020, doi: 10.1109/ACCESS.2020.3036728.
- [2] N. Abdalgawad, A. Sajun, Y. Kaddoura, I. A. Zualkernan, and F. Aloul, "Generative Deep Learning to Detect Cyberattacks for the IoT-23 Dataset," *IEEE Access*, vol. 10, pp. 6430–6441, 2022, doi: 10.1109/ACCESS.2021.3140015.
- [3] L. Vigoya, D. Fernandez, V. Carneiro, and F. J. N6voa, "IoT dataset validation using machine learning techniques for traffic anomaly detection," *Electron.*, vol. 10, no. 22, 2021, doi: 10.3390/electronics10222857.
- [4] A. M. Bamhdi, I. Abrar, and F. Masoodi, "An ensemble based approach for effective intrusion detection using majority voting," *Telkonnika (Telecommunication Computing Electronics and Control)*, vol. 19, no. 2, pp. 664–671, 2021, doi: 10.12928/TELKOMNIKA.v19i2.18325.
- [5] Q. Meng, D. Li, and Y. Ma, "Research and Application Based on Network Security Monitoring Platform and Device," *2019 IEEE PES Innov. Smart Grid Technol. Asia, ISGT 2019*, pp. 716–719, 2019, doi: 10.1109/ISGT-Asia.2019.8881520.
- [6] R. Patil, H. Dudeja, and C. Modi, "Designing an efficient security framework for detecting intrusions in virtual network of cloud computing," *Comput. Secur.*, vol. 85, pp. 402–422, 2019, doi: 10.1016/j.cose.2019.05.016.
- [7] A. Derhab, A. Aldweesh, A. Z. Emam, and F. A. Khan, "Intrusion Detection System for Internet of Things Based on Temporal Convolution Neural Network and Efficient Feature Engineering," *Wirel. Commun. Mob. Comput.*, vol. 2020, 2020, doi: 10.1155/2020/6689134.
- [8] Y. Fu, Y. Du, Z. Cao, Q. Li, and W. Xiang, "A Deep Learning Model for Network Intrusion Detection with Imbalanced Data," *Electron.*, vol. 11, no. 6, p. 898, Mar. 2022, doi: 10.3390/electronics11060898.
- [9] O. Aslan and R. Samet, "A Comprehensive Review on Malware Detection Approaches," *IEEE Access*, vol. 8, pp. 6249–6271, 2020, doi: 10.1109/ACCESS.2019.2963724.
- [10] Y. Ding and Y. Zhai, "(CNN) Intrusion detection system for NSL-KDD dataset using convolutional neural networks," *ACM Int. Conf. Proceeding Ser.*, pp. 81–85, 2018, doi: 10.1145/3297156.3297230.
- [11] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," *2015 Mil. Commun. Inf. Syst. Conf. MilCIS 2015 - Proc.*, 2015, doi: 10.1109/MilCIS.2015.7348942.
- [12] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," *Futur. Gener. Comput. Syst.*, vol. 100, pp. 779–796, 2019, doi: 10.1016/j.future.2019.05.041.
- [13] J. A. Amien, H. A. Ghani, N. I. Md Saleh, E. Ismanto, and R. Gunawan, "Intrusion detection system for imbalance ratio class using weighted XGBoost classifier," *Telkonnika (Telecommunication Computing Electronics and Control)*, vol. 21, no. 5, p. 1102–1112, Oct. 2023, doi: 10.12928/telkonnika.v21i5.24735.
- [14] S. I. Popoola, B. Adebisi, R. Ande, M. Hammoudeh, K. Anoh, and A. A. Atayero, "Smote-drrn: A deep learning algorithm for botnet detection in the internet-of-things networks," *Sensors*, vol. 21, no. 9, 2021, doi: 10.3390/s21092985.
- [15] M. Gao, X. Hong, S. Chen, and C. J. Harris, "A combined SMOTE and PSO based RBF classifier for two-class imbalanced problems," *Neurocomputing*, vol. 74, no. 17, pp. 3456–3466, 2011, doi: 10.1016/j.neucom.2011.06.010.
- [16] J. Li *et al.*, "SMOTE-NaN-DE: Addressing the noisy and borderline examples problem in imbalanced classification by natural neighbors and differential evolution," *Knowledge-Based Syst.*, vol. 223, 2021, doi: 10.1016/j.knsys.2021.107056.
- [17] C. Liu, Z. Gu, and J. Wang, "A Hybrid Intrusion Detection System Based on Scalable K-Means+ Random Forest and Deep Learning," in *IEEE Access*, vol. 9, pp. 75729–75740, 2021, doi: 10.1109/ACCESS.2021.3082147.
- [18] Z. Hu, L. Wang, L. Qi, Y. Li, and W. Yang, "A novel wireless network intrusion detection method based on adaptive synthetic sampling and an improved convolutional neural network," *IEEE Access*, vol. 8, pp. 195741–195751, 2020, doi: 10.1109/ACCESS.2020.3034015.
- [19] J. Liu, Y. Gao, and F. Hu, "A fast network intrusion detection system using adaptive synthetic oversampling and LightGBM," *Comput. Secur.*, vol. 106, p. 102289, Jul. 2021, doi: 10.1016/j.cose.2021.102289.
- [20] I. Idrissi, M. Boukabous, M. Azizi, O. Moussaoui, and H. E. Fadili, "Toward a deep learning-based intrusion detection system for iot against botnet attacks," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 1, pp. 110–120, 2021, doi: 10.11591/ijai.v10.i1.pp110-120.
- [21] G. Karatas, O. Demir, O. K. Sahingoz, K. Sahingoz, O. K. Sahingoz, and O. K. Sahingoz, "Increasing the Performance of Machine Learning-Based IDSs on an Imbalanced and Up-to-Date Dataset," *IEEE Access*, vol. 8, pp. 32150–32162, 2020, doi: 10.1109/ACCESS.2020.2973219.
- [22] N. Islam *et al.*, "Towards Machine Learning Based Intrusion Detection in IoT Networks," *Comput. Mater. Contin.*, vol. 69, no. 2, pp. 1801–1821, 2021, doi: 10.32604/cmc.2021.018466.
- [23] S. I. Popoola, B. Adebisi, R. Ande, M. Hammoudeh, and A. A. Atayero, "Memory-efficient deep learning for botnet attack detection in iot networks," *Electron.*, vol. 10, no. 9, 2021, doi: 10.3390/electronics10091104.
- [24] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, no. 3, pp. 1322–1328, Jun. 2008, doi: 10.1109/IJCNN.2008.4633969.
- [25] T. Chen, C. Guestrin, and C. G. T. Chen, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 19, no. 6, pp. 785–794, Aug. 2016, doi: 10.1145/2939672.2939785.
- [26] X. Liu *et al.*, "NADS-RA: Network Anomaly Detection Scheme Based on Feature Representation and Data Augmentation," *IEEE Access*, vol. 8, pp. 214781–214800, 2020, doi: 10.1109/ACCESS.2020.3040510.
- [27] P. S. Muhuri, P. Chatterjee, X. Yuan, K. Roy, and A. Esterline, "Using a long short-term memory recurrent neural network (LSTM-RNN) to classify network attacks," *Information*, vol. 11, no. 5, p. 243, 2020, doi: 10.3390/INFO11050243.
- [28] S. S. Dhaliwal, A. Al Nahid, and R. Abbas, "Effective intrusion detection system using XGBoost," *Information*, vol. 9, no. 7, 2018, doi: 10.3390/info9070149.
- [29] M. A. Ferrag and L. Maglaras, "DeepCoin: A Novel Deep Learning and Blockchain-Based Energy Exchange Framework for Smart Grids," *IEEE Trans. Eng. Manag.*, vol. 67, no. 4, pp. 1285–1297, 2020, doi: 10.1109/TEM.2019.2922936.

BIOGRAPHIES OF AUTHORS






Januar Al Amien    completed education Bachelor's degree in the Department of Informatics Engineering, STMIK-AMIK Riau. Master's degree in Master of Information Technology at Putra Indonesia University Padang. Now working as a lecturer in the Department of Computer Science, University Muhammadiyah of Riau. With research interests in the field of machine learning algorithms and AI. He can be contacted at email: januaralamien@umri.ac.id.






Hadhrami Ab Ghani    received his Bachelor degree in Electronics Engineering from Multimedia University Malaysia (MMU) in 2002. In 2004, he completed his master's degree in Telecommunication Engineering at The University of Melbourne. He then pursued his Ph.D. at Imperial College London in intelligent network systems and completed his Ph.D. in 2011. He can be contacted at email: hadhrami.ag@umk.edu.my.






Nurul Izrin Md Saleh    obtained a Bachelor's degree in Information Technology from Multimedia University Malaysia (MMU). He completed his Master's degree in Computer Science at the University of Putra Malaysia. Then complete a Ph.D. at The University of Brunel London in the same field of study. She can be contacted at email: izrin@utem.edu.my.






Soni    received the Bachelor's degree in Department of Informatics Engineering from STMIK AMIK Riau, Indonesia and the Master's degree in Computer Science Islamic University of Indonesia. now works as a lecturer at the Faculty of Computer Science, University of Muhammadiyah Riau. His current research interests include machine learning algorithms and AI. He can be contacted at email: soni@umri.ac.id.



Yulia Fatma    completed education Bachelor's degree in the Department of Informatics Engineering, University of Amikom Yogyakarta. Master's degree in Master of Computer Science at Gadjah Mada University. Now working as a lecturer in the Department of Informatics, University of Muhammadiyah Riau. With research interests in the field of cryptography and AI. She can be contacted at email: yuliafatma@umri.ac.id.



Regiolina Hayami    graduated with a Bachelor's degree at Department of Informatics Engineering, State Islamic University of Sultan Syarif Kasim Riau, and Master's degree in Master of Information Technology at Putra Indonesia University Padang. Now works as a lecturer at the Faculty of Computer Science, University Muhammadiyah of Riau. With research interests in the field of machine learning algorithms and AI. She can be contacted at email: regiolinahayami@umri.ac.id.