SINERGI Vol. 29, No. 3, October 2025: 779-792
http://publikasi.mercubuana.ac.id/index.php/sinergi
http://doi.org/10.22441/sinergi.2025.3.019

Optimizing intrusion detection with data balancing and feature selection techniques

Zulhipni Reno Saputra Elsi[1*], Ahmad Affandi Supli[2], Jimmie[1], Muhammad Ghozi Al-Faris[1], David Agustianto Rapel[1]
[1]Department of Information Technology, Faculty of Engineering, Muhammadiyah University of Palembang, Indonesia
[2]Digital Media Technology Department, Xiamen University Malaysia, Malaysia

Abstract
The rapid growth of IoT devices has brought significant security challenges, particularly in detecting various types of attacks within heterogeneous network environments. This study explores the effectiveness of data balancing techniques, including Random Under Sampling (RUS), Cost-Sensitive Learning (CSL), Synthetic Minority Oversampling Technique (SMOTE), and Randomized Combination Sampling (RCS). Feature selection methods, namely correlation (threshold 0.8) and mutual information (top 15 features), were employed to optimize feature sets. The Decision Tree (DT) and Linear Discriminant Analysis (LDA) classifiers were used to evaluate the performance of balanced datasets. The evaluation metrics included accuracy, precision, recall, F1-score, G-mean, and ROC curves. The results revealed that SMOTE and RCS outperformed other balancing methods, with SMOTE achieving the highest accuracy (98.7%) and RCS demonstrating robust G-mean values across both feature selection techniques. DT consistently showed better performance compared to LDA across all metrics, while feature selection significantly improved the classification results, particularly under mutual information criteria. However, the analysis highlighted limitations of LDA in handling imbalanced datasets and high-dimensional features. This study concludes that a combination of advanced data balancing and effective feature selection significantly enhances the accuracy of intrusion detection in IoT networks. Future work will focus on integrating real-time detection systems and exploring hybrid models to further improve the detection of complex attacks in dynamic IoT environments.

Keywords:
Correlation;
Imbalance;
IoT;
Load Balancing;
Mutual information;
RT-IOT22;

Article History:
Received: December 6, 2024
Revised: March 24, 2025
Accepted: April 10, 2025
Published: September 5, 2025

Corresponding Author:
Zulhipni Reno Saputra Elsi
Department of Information Technology, Muhammadiyah University of Palembang, Indonesia
Email: zulhipni_renosaputra@um-palembang.ac.id

INTRODUCTION

The development of Internet of Things (IoT) technology has brought great benefits to various aspects of life, including in the industrial sector [1], smart homes [2], and transportation [3]. However, the increasing use of IoT devices also expands the potential for cybersecurity attacks [4]. Attacks on IoT devices are increasing, so a reliable intrusion detection system (IDS) is needed to protect IoT networks [5][6]. To overcome this problem, the Intrusion Detection System (IDS) based on Machine Learning (ML) is increasingly being used in IoT network security [7]. ML-based IDS requires proper dataset management, especially in dealing with dataset imbalance, which is an unbalanced data distribution between the normal class and the attack class [8][9]. All of this often arises due to the dynamic nature of data collection in IoT networks and data distribution in the real world. Datasets such as IoT-23 [7] and IoTID20 [10] are examples of unbalanced IoT datasets, where the attack class is less than the normal class. This alignment can lead to bias in

ML models, which tend to favor majority class predictions, thus neglecting the detection of less frequent attacks. In addition, the data generated by IoT devices is dynamic and continuous, which further increases the complexity of anomaly detection [11].

Several previous studies have identified that imbalanced datasets in IoT networks pose a major challenge to the effectiveness of ML-based IDS. Approaches such as the Synthetic Minority Oversampling Technique (SMOTE) [12], Adaptive Synthetic Sampling (ADASYN) [13], Random Under Sampling (RUS) [14], Ensemble and ML methods have been proposed to address this issue. Researchers [7] showed that the combination of SMOTE and under sampling techniques successfully improved the accuracy to 96.81% on the IoT-23 dataset. Researchers [10] reported excellent results using a combination of deep learning and data balancing techniques on IoTID20, with an AUC reaching 99.93%.

However, the implementation of these techniques also has drawbacks, such as the risk of overfitting on synthetic data or removing important features in under sampling [15]. In addition, research [11] emphasizes the importance of handling the dynamic nature of IoT data to improve detection accuracy.

Managing IoT data integration requires solutions that are not only able to improve model accuracy but also consider computational efficiency and resilience to real-time data changes. This research aims to address these challenges by exploring various data balancing techniques, such as RUS, SMOTE, and Cost Sensitive Learning (CSL) [16], and Random Combination Sampling (RCS). This balancing is expected to reduce bias towards the majority class, improve accuracy on the minority class, and produce a more reliable IDS for IoT networks. This research will also provide an in-depth evaluation using metrics such as accuracy, precision, recall, F1 Score, and G-Men to ensure model performance on the highly imbalanced RT-IOT22 dataset [17].

This research contributes in several significant aspects to improve IoT network security through processing imbalanced datasets:

1. Data Balancing Strategy Development by implementing and comparing various techniques such as RUS, SMOTE, CSL, and RCS.
2. Optimization of Machine Learning Model for IoT by using Mutual information-based feature selection (MIFS), Correlation-based feature selection (CFS) and performing classification with Decision Tree (DT) [18] and Linear Discriminant Analysis (LDA) [19].
3. Evaluation with Metrics such as accuracy, precision, recall, F1 Score, and G-Men, this study ensures that the evaluation of model performance is more representative of the needs of attack detection on imbalanced IoT data.

This research proposes an optimized intrusion detection framework for IoT networks by integrating feature selection methods with hybrid sampling techniques and lightweight classifiers, evaluated on protocol-specific datasets to address data imbalance and computational constraints in real-world scenarios.

RELATED WORK

Recent studies have significantly addressed challenges in intrusion detection systems (IDS) for IoT networks and data imbalance in machine learning. This section reviews key works, focusing on their methods, contributions, and implications for IDS and other ML applications.

Researchers [20] proposed an automated myocardial infarction detection system using CNN and a hybrid CNN-LSTM with SMOTE-Tomek Link approach to handle imbalanced datasets. Their study showed that data balancing significantly improved the model accuracy up to 99.89%, which is relevant for clinical applications. This underscores the importance of data balancing techniques in healthcare and other domains facing class imbalance issues.

Researchers [21] proposed an IoT-specific IDS using ensemble methods like RF, Extreme Gradient Boosting (XGBoost), and Light Gradient Boosting Machine (LGBM) on the imbalanced DS2OS dataset. Their LGB-IDS model achieved 99.92% accuracy, excelling in speed and threat detection, showing strong potential for real-world IoT IDS applications.

Researchers [22] analyzed the impact of class imbalance on the performance of machine learning-based IDS using KNN, Gradient Boosting, and SVM algorithms on the BoT-IoT dataset. By applying SMOTE and random under sampling, they reported a significant improvement in the F1 score, highlighting the importance of balancing techniques in improving the reliability of IDS in IoT networks. Researchers [23] addressed the challenge of class imbalance in IDS datasets, which often reduces detection performance for rare attacks. Karatas used the CSE-CIC-IDS2018 dataset and applied SMOTE with six ML algorithms to improve detection rates. Researchers [24] evaluated ML models with various resampling strategies using F1-score and G-mean, demonstrating that proper integration enhances IDS robustness and accuracy in

identifying minority class intrusions within imbalanced network traffic. And researchers [25] developed a Collaborative Intrusion Detection System (CIDS) using Weighted Ensemble Averaging Deep Neural Network (WEA-DNN). This system achieves high accuracy and adaptability in detecting coordinated cyberattacks in heterogeneous networks, demonstrating the effectiveness of collaborative approaches in handling complex attack patterns.

Research on handling data imbalance in IDS has been growing rapidly, with various methods proposed to improve the accuracy and reliability of classification models, which have been summarized in Table 1.

Table 1. Summary of Approaches and Research Results on Imbalanced Data

Ref	Methodology	Dataset	Measurement Indicators	Key Results
[26]	Resampling techniques: Random Oversampling, Random Under sampling , SMOTE, and Adaptive Synthetic Sampling.	KDD99, UNSW-NB15, UNSW-NB17, UNSW-NB18	Macro Precision: 98%, Macro Recall: 96%, Macro F1-Score: 97%	Oversampling improves Macro Precision and Macro Recall, especially on minority classes. Resampling helps detect more minority data but increases training time.
[27]	A combination of a Deep Neural Network (DNN) with a Bagging Classifier approach. Further experiments using CNN and hybrid CNN+LSTM.	NSL-KDD, KDDCUP99, UNSW-NB15, Bot-Io	Accuracy: 99.8%, Precision: 99.5%, Recall: 99.6%, F1-Score: 99.6%	The DNN model with bagging produces high accuracy (99.8%), with low False Positive Rate. The combination of CNN+LSTM is more effective on IoT datasets such as Bot-Io.
[28]	Cluster-SMOTE + K-Means algorithm for preprocessing and Two-Layer CNN for classification.	UNSW-NB15, CICIDS2017	Accuracy: 98.77%, Recall: 98.3%, Precision: 98.9%, F1-Score: 98.6%, AUC: 99.2%	CSK-CNN provides the highest AUC (99.2%) and F1-Score 98.6%, demonstrating the model's ability to handle imbalanced data with high accuracy and generalization on both datasets.
[29]	Hybrid feature selection (filter + wrapper); Two-level IDS (normal vs. attack, then attack type); SMOTE for class imbalance; ML algorithms: Decision Tree, Random Forest, GNB, KNN	BoT-IoT, TON-IoT, CIC-DDoS2019	Accuracy: 99.82-100% Precision: 98.65-99.99% Recall: 98.56-100% F1-Score: 98.7-99.9% Detection Time: 0.02-0.15s	Decision Tree achieved highest accuracy and lowest detection time, outperforming other algorithms and prior works
[30]	SMOTE, Gaussian Distribution, SVM, RF methods.	MQTT-IOT-IDS2020	Accuracy: 98.7%, Precision: 96.5%, Recall: 95.8%, F1-Score: 96.1%	Significant improvement in model performance when using oversampling techniques.
[31]	RO, DT, RF, and SVM techniques.	Specific IoT dataset (unspecified).	Accuracy: 97.3%, Precision: 94.2%, Recall: 92.7%, F1-Score: 93.4%	RO is able to improve model performance with a more balanced data distribution.
[32]	SMOTE, ADASYN and XGBoost.	IoT dataset (unspecified).	TPR: 92.5%, FPR: 5.3%, Accuracy: 95.4%, Precision: 93.1%, Recall: 92.8%, F1-Score: 92.9%	Oversampling techniques have been shown to be helpful in increasing the sensitivity of the model to minority attacks.
[33]	Federated Learning (FL), SMOTE, ADASYN, and Generative Adversarial Networks (GANs).	TON_IoT and DS2OS IoT datasets	F1 score: up to 0.91; Precision: up to 0.89; Recall: up to 0.92; Accuracy: up to 95%	Data augmentation improves performance by up to 22.9% in detecting anomalies compared to the baseline without data augmentation.
[34]	Feature engineering with mRMR + SMOTE; CatBoost classifier; Optuna for hyperparameter tuning; Tested on binary and multi-class	NSL-KDD, UNSW-NB15, CICIDS-2017	Accuracy: 98.41-99.59% Precision: 97.36-99.44% Recall: 97.71-99.55% F1-Score: 97.52-99.49%	Optimized CatBoost with mRMR + SMOTE consistently outperformed baseline methods across all datasets
[35]	SMOTE, ADASYN, and BoostedEnML	CSE-CIC-IDS2018 and CIC-IDS2017 datasets	Precision: 1.0; Recall: 1.0; F1 score: 1.0; AUC: 1.0	BoostedEnML with SMOTE/ADASYN achieves 100% accuracy on multiclass classification on IDS dataset with reduced False Positives and False Negatives.

Table 1 summarizes the various approaches that have been applied, including machine learning methods, deep learning, and data preprocessing techniques such as oversampling and under sampling. Each approach is evaluated using various performance metrics on different benchmark datasets, demonstrating its effectiveness in handling imbalanced data.

Although efforts to address data imbalance and optimize model architecture in machine learning-based IDS have been made, there is still a lack of specific approaches for IoT environments, especially those using communication protocols such as MQTT. Most studies still focus on traditional datasets such as NSL-KDD, BoT-IoT, and CICIDS2017, without considering the specific characteristics of IoT traffic. In addition, the balancing methods used are generally limited to oversampling and ensemble, while real-time adaptation and federated learning approaches are still rarely explored. Therefore, more comprehensive research is needed to develop more effective and adaptive IDS for the IoT ecosystem.

METHOD

This section describes the methods used in data processing and the process for generating the IDS model.

Raw Dataset

The RT_IOT2022 dataset is obtained from real-time IoT infrastructures from ThingSpeak-LED, Wipro-Bulb, and MQTT-Temp, and then extracted to obtain useful features for attack detection. The data consists of 85 features with 12 classes.

Table 2 presents the attack types included in the RT_IOT2022 dataset along with the number of recorded packets for each type. The dataset encompasses a variety of attack categories, reflecting different intrusion techniques in IoT environments.

Table 2. Attack type dataset RT_IOT2022

Attack_type	Packets
DOS_SYN_Hping	94659
Thing_Speak	8108
ARP_poisioning	7750
MQTT_Publish	4146
NMAP_UDP_SCAN	2590
NMAP_XMAS_TREE_SCAN	2010
NMAP_OS_DETECTION	2000
NMAP_TCP_scan	1002
DDOS_Slowloris	534
Wipro_bulb	253
Metasploit_Brute_Force_SSH	37
NMAP_FIN_SCAN	28

Proposed Model

Machine learning-based Intrusion Detection Systems (IDS) for IoT networks consist of several main stages, namely data ingestion, storage, feature engineering, model training, evaluation, deployment, monitoring, and retraining [36]. These stages form a continuous learning cycle to improve the accuracy of threat detection in the IoT ecosystem.

The proposed architecture is divided into several parts, processes as shown in Figure 1. Figure 1 illustrates the process flow in applying machine learning techniques for network intrusion detection, which is divided into several important stages. Here is an explanation for each part:

1. Data Preprocessing: This stage consists of two main sub-stages, namely:
   a) Preparation: Includes the process of data cleaning, data labeling, and data normalization to prepare the data before being used in model training.
   b) Balancing: Using various data balancing techniques, such as RUM, SMOTE, CSL, and RCS, to address class imbalance issues in the data.

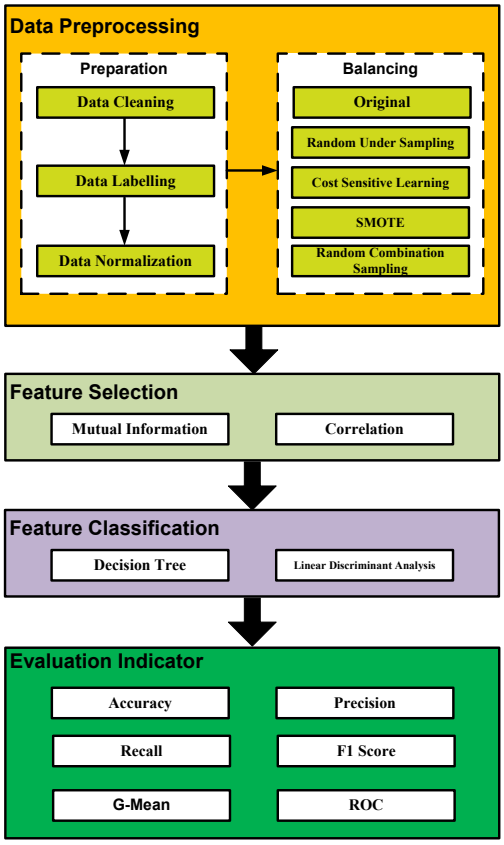


Figure 1. Machine learning architecture of our proposed model

2. Feature Selection: At this stage, relevant features are selected using techniques such as MIFS and CFS to ensure that only the most informative features are used in the model.
3. Feature Classification: Here, a classification model is applied using algorithms such as DT and LDA to classify data based on the selected features.
4. Evaluation Indicator: The results of the classification model are evaluated using several performance indicators, including Accuracy, Precision, Recall, F1 Score, and G-Mean, to assess the effectiveness of intrusion detection.

Balancing

Balancing in the context of machine learning refers to techniques for dealing with imbalanced datasets, where one class has a much larger number of samples than the other classes [37][38]. This imbalance can affect model performance because the algorithm tends to prioritize predictions for the majority class and ignores the minority class, which is often more important to analyze [39][40]. The relatedness parameter, especially in the data distribution, has a significant impact on the model performance, especially in classification problems. If the dataset is highly imbalanced, the model tends to be biased towards the majority class, which leads to misclassification of the minority class and compromises the performance of standard learning algorithms [41][42][43]. In many cases, an imbalanced dataset occurs when one class is much smaller than the other classes. This imbalance can result in high accuracy, even though the model is not able to detect the minority class well, which may be more important in the context of a particular application [44][45].

In addition to data synchronization issues, the performance of machine learning-based IDS in IoT is also influenced by several other factors, such as real-time processors, where IDS must be able to detect threats instantly without high latency, so that Edge Computing and Federated Learning-based approaches can be used to accelerate detection without having to send all data to a central server [46]. Another factor is scalability, because IoT networks have a very large number of devices, so the IDS model must be able to handle the growth in the number of devices without experiencing a decrease in performance [47]. In addition, resource limitations on IoT devices, which often have limited computing power and memory, make IDS need to use lightweight models, such as DT or AB based ensemble learning, to increase efficiency [48].

This study uses data balancing techniques in the following ways: 1) RUS, a technique for randomly reducing the number of samples from the majority class so that the number is comparable to the minority class; 2) SMOTE is a popular over-sampling technique where synthetic samples from the minority class are created based on interpolation between existing samples; 3) CSL is a technique that does not change the data distribution but adapts the learning algorithm by giving greater weight to prediction errors in the minority class; 4) RCS is a combination of RUS and SMOTE, this technique balances the dataset by reducing some of the majority class samples while adding synthetic samples to the minority class.

Selection Feature

This study uses selection features for MIFS and CFS, MIFS looks for the best 15 feature values from the MI Score while CFS selects features based on the correlation value of 0.8. MIFS gets 15 different features for original data, RUM, SMOTE, CSL, and RCS. While CFS produces a different number of features for each original data, RUM, CSL, SMOTE, and RCS. Original data produces 53 features, RUM data produces 58 features, CSL data produces 64, SMOTE data produces 61 features, and RCS data produces 51 features.

Classification

Classification is an important process in the workflow that aims to build an ML or DL model that is able to predict or classify data based on previously selected features [49]. In this study, there are two methods used, namely DT and LDA. DT is one of the most widely used models due to its simplicity and high interpretability [50][51]. This method works by building a DT from a dataset, where each node represents a feature, a branch represents a feature value, and a leaf represents a class or final result [52][53]. While LDA is a statistical classification method that seeks a linear projection of the data to maximize the separation between classes [54].

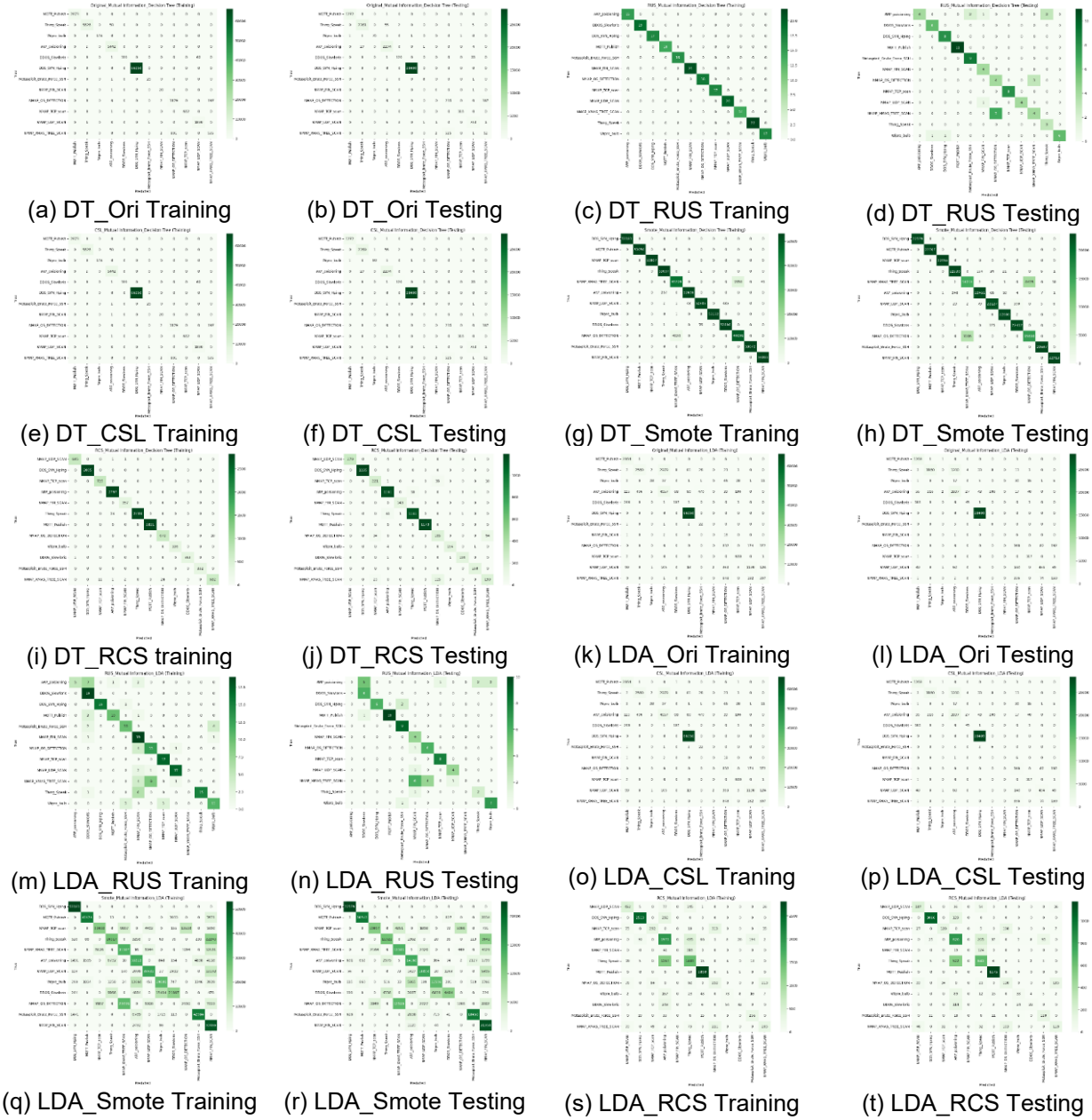
RESULTS AND DISCUSSION

In this study, we implemented several techniques to handle data imbalance and improve the performance of the attack detection system. We compared the original data and four data balancing techniques, namely RUS, CSL, SMOTE, and RCS. Each technique was followed by two feature selection methods, namely CFS

with a threshold of 0.8 and MIFS to select the best 15 features. After the features were selected, we used two different classification techniques, namely DT and LDA. Table 3 illustrates the distribution of attack data before and after the balancing technique was performed.

Table 3. Distribution of RT_IOT2022 Dataset before and after balancing

Attack_type	Original	RUS	CSL	SMOTE	RCS
DOS_SYN_Hping	94659	25	94659	75762	4000
Thing_Speak	8108	25	8108	75762	4000
ARP_poisioning	7750	25	7750	75762	4000
MQTT_Publish	4146	25	4146	75762	4000
NMAP_UDP_SCAN	2590	25	2590	75762	1000
NMAP_XMAS_TREE_SCAN	2010	25	2010	75762	1000
NMAP_OS_DETECTION	2000	25	2000	75762	1000
NMAP_TCP_scan	1002	25	1002	75762	1000
DDOS_Slowloris	534	25	534	75762	534
Wipro_bulb	253	25	253	75762	500
Metasploit_Brute_Force_SSH	37	25	37	75762	500
NMAP_FIN_SCAN	28	25	28	75762	500



(a) DT_Ori Training (b) DT_Ori Testing (c) DT_RUS Traning (d) DT_RUS Testing

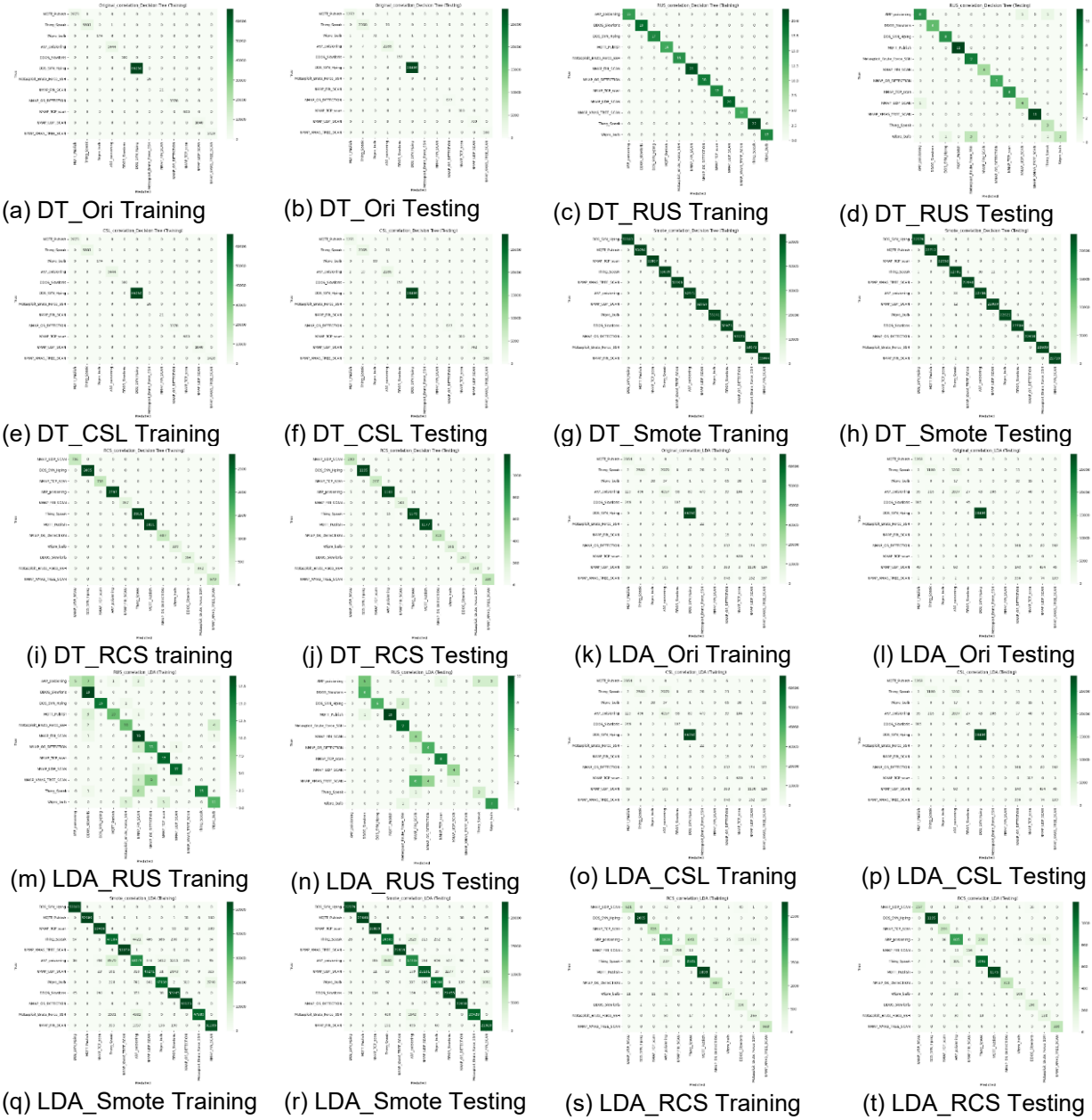
(e) DT_CSL Training (f) DT_CSL Testing (g) DT_Smote Traning (h) DT_Smote Testing

(i) DT_RCS training (j) DT_RCS Testing (k) LDA_Ori Training (l) LDA_Ori Testing

(m) LDA_RUS Traning (n) LDA_RUS Testing (o) LDA_CSL Traning (p) LDA_CSL Testing

(q) LDA_Smote Training (r) LDA_Smote Testing (s) LDA_RCS Training (t) LDA_RCS Testing

Figure 2. Confusion Matrix MIFS



(a) DT_Ori Training (b) DT_Ori Testing (c) DT_RUS Traning (d) DT_RUS Testing

(e) DT_CSL Training (f) DT_CSL Testing (g) DT_Smote Traning (h) DT_Smote Testing

(i) DT_RCS training (j) DT_RCS Testing (k) LDA_Ori Training (l) LDA_Ori Testing

(m) LDA_RUS Traning (n) LDA_RUS Testing (o) LDA_CSL Traning (p) LDA_CSL Testing

(q) LDA_Smote Training (r) LDA_Smote Testing (s) LDA_RCS Training (t) LDA_RCS Testing

Figure 3. Confusion Matrix CFS

Table 3 shows the distribution of the amount of data for each attack type (Attack Type) based on the application of various data balancing techniques: Original (without balancing), RUS, CSL, SMOTE, and RCS. SMOTE is best suited to ensure a uniform data distribution, while RCS provides more flexibility in determining the amount of data. RUS is effective in creating a balanced data distribution, but risks reducing important information. CSL is a safe choice because it does not modify the original data but only modifies the training approach. This study produces a Confusion Matrix that can be used to calculate various performance metrics, such as Accuracy, Precision, Recall, F1-Score, and G-Mean to visualize the trade-off between True Positive Rate (TPR) and False Positive Rate (FPR). Figure 2 and Figure 3 illustrate the classification results with the DT and LDA algorithms.

Figure 2 is the Confusion Matrix of MIFS results, while Figure 3 is the Confusion Matrix of CFS results. This Confusion Matrix illustrates the results of training data and testing data with DT and LDA classifications. From the confusion matrix, the Precision, Recall, F1-Score values are obtained which are displayed in Table 4, Table 5, Figure 4 and Figure 5.

Table 4 compares the performance of DT and LDA models on data that has been balanced with various methods (Original, RUS, CSL, SMOTE, RCS) using the evaluation metrics Precision, Recall, and F1 Score. The DT model consistently outperforms LDA in all balancing methods and evaluation metrics, both on training and testing data. The RUS and CSL Balancing methods produce perfect precision and recall on training data for DT, but performance decreases on testing data and SMOTE is the best balancing method on testing data, providing the highest precision, recall, and F1 Score performance for DT. while RCS also produces good performance, but is still slightly below SMOTE for testing data. Performance on training data tends to be higher than on testing data. This is an indication that some methods such as RUS may cause the model to overfit on training data due to overly simple data.

Table 5 compares the performance of DT and LDA models with a correlation selection feature of 0.8. Overall, the DT model consistently outperforms LDA in terms of precision, recall, and F1 score, both on training and testing data. While LDA shows the best performance on the RUS method compared to other methods. RUS provides perfect performance for DT on training data, but its generalization to testing data is poor. And SMOTE is the best method for testing, producing the highest precision, recall, and F1 score for DT, indicating better generalization ability while RCS has almost comparable results with SMOTE, but still slightly lower especially in precision. All methods show a decrease in performance from training to testing data, especially on LDA. This indicates that LDA is more susceptible to generalization challenges than DT.

Figure 4 compares the accuracy of DT and LDA classification models based on two feature selection methods: MIFS and CFS. DT outperforms LDA in all balancing techniques and feature selection approaches, with consistently higher accuracy, CFS is more effective than MIFS, especially for SMOTE and RCS, producing near-perfect accuracy on testing data. In the Balancing Technique, SMOTE and RCS provide the best results in both feature selections, demonstrating their ability to improve the distribution of the minority class without sacrificing model performance, while RUS is less effective especially on testing data, because accuracy decreases drastically for both models, indicating poor generalization and CSL does not provide significant improvement compared to the original data, both in DT and LDA.

Table 4. Performance Comparison with MIFS

Classification with balancing	Training			Testing		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
DT_Ori	0.921	0.914	0.909	0.812	0.804	0.805
LDA_Ori	0.588	0.581	0.522	0.568	0.547	0.509
DT_RUS	**1.000**	**1.000**	**1.000**	0.805	0.824	0.786
LDA_RUS	0.725	0.702	0.684	0.594	0.738	0.637
DT_CSL	0.921	0.914	0.909	0.811	0.803	0.804
LDA_CSL	0.588	0.581	0.522	0.568	0.547	0.509
DT_Smote	0.980	0.980	0.980	0.937	0.937	0.937
LDA_Smote	0.632	0.618	0.605	0.63	0.616	0.603
DT_RCS	0.989	0.990	0.989	0.893	0.897	0.894
LDA_RCS	0.557	0.503	0.506	0.553	**0.497**	0.503

Table 5. Performance Comparison with CFS

Classification with balancing	Training			Testing		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
DT_Ori	**1.000**	**1.000**	**1.000**	0.931	0.954	0.938
LDA_Ori	0.588	0.581	0.522	0.568	0.547	**0.509**
DT_RUS	**1.000**	**1.000**	**1.000**	0.911	0.915	0.895
LDA_RUS	0.725	0.702	0.684	0.594	0.738	0.637
DT_CSL	**1.000**	**1.000**	**1.000**	0.937	0.951	0.942
LDA_CSL	0.588	0.581	0.522	0.568	0.547	**0.509**
DT_Smote	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
LDA_Smote	0.944	0.943	0.943	0.944	0.943	0.943
DT_RCS	**1.000**	**1.000**	**1.000**	0.992	0.995	0.993
LDA_RCS	0.879	0.889	0.878	0.878	0.887	0.878

Figure 5 presents the evaluation results of the classification model's performance based on the G-Mean, which reflects the balance between recall and specificity. G-Mean is particularly important for imbalanced datasets, as it provides an overview of the model's ability to handle both majority and minority classes simultaneously. DT is superior to LDA due to its higher G-Mean value in all balancing techniques and feature selection approaches. CFS is more effective than MIFS in improving G-Mean, especially in DT with SMOTE and RCS. The SMOTE and RCS balancing techniques provide the best results for DT, with almost perfect G-Mean, while LDA fails to produce adequate G-Mean values, especially with MIFS, although there is a slight increase in CFS. Balancing with RUS is ineffective, especially in LDA, where G-Mean remains zero in all scenarios.

Table 6 presents a comparative analysis of accuracy and G-Mean across various classifier methods used in intrusion detection. The comparison includes previously proposed methods and the newly developed models.

In the proposed model, the use of DT and LDA with various balancing techniques showed mixed results. Several DT variants, such as DT_Ori_CFS, DT_RUS_MI, DT_RUS_CFS, DT_CSL_CFS, and DT_Smote_CFS, achieved 100% accuracy, indicating that the model is very good at recognizing patterns in the data. However, despite the high accuracy, the G-Mean of some models, such as DT_Ori_MI was only 49.21%, indicating that the model is less able to handle class precision. Meanwhile, the LDA method performed much worse, with some variants such as LDA_RCS_MI and LDA_RCS_CFS having a G-Mean of 0.00%, meaning the model failed to recognize a single class at all.

Overall, although some models have high accuracy, the low G-Mean indicates that the model is less effective in handling data smoothness. The best models are those that have a balance between high accuracy and G-Mean, such as DT_RUS_CFS and DT_Smote_CFS, which achieve 100% accuracy and G-Mean close to 100%. This shows that the Decision Tree method with balancing techniques such as SMOTE and CFS is a more reliable choice than other methods, especially for applications in IDS in IoT Smart Home, where precision in detecting attacks from various classes is very important.

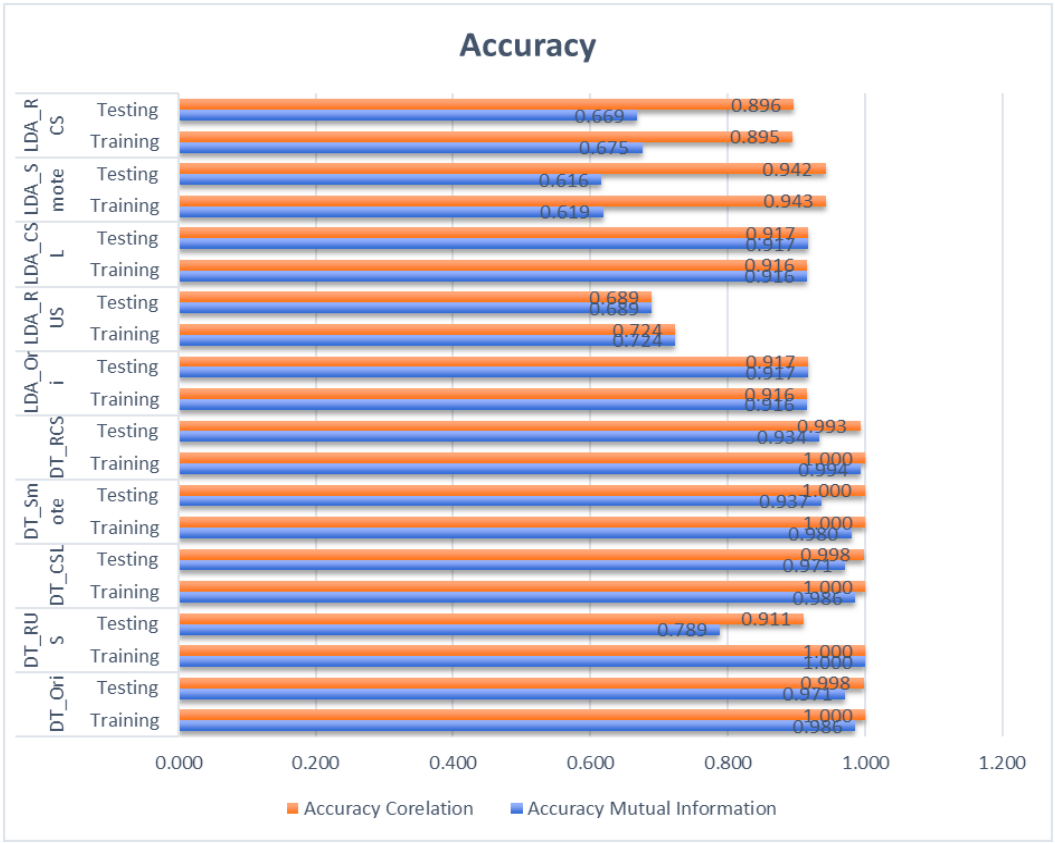


Figure 4. Comparison of Accuracy values

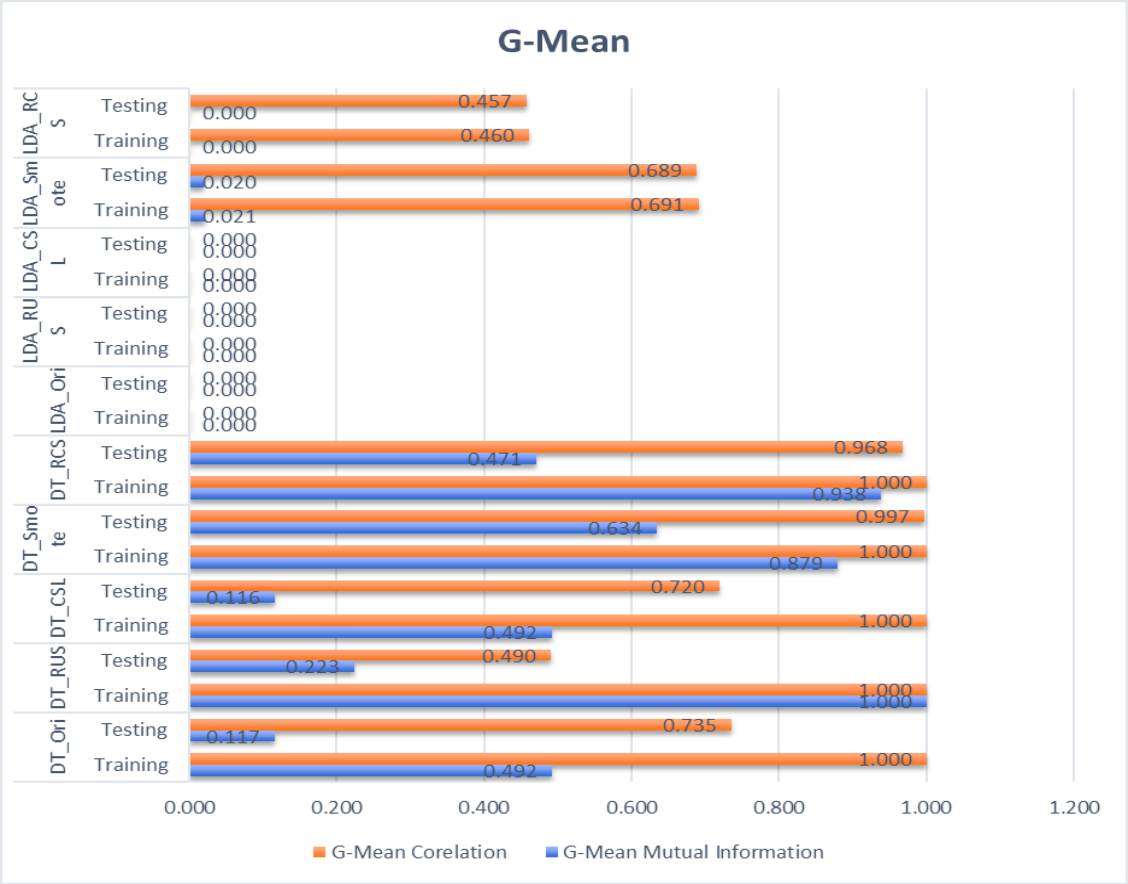


Figure 5. Comparison of G-Mean values

Table 6. Comparison of Accuracy and G-Mean

Ref.	Classifier Method	Accuracy	G-Mean
[55]	DL ensemble	99.3%	99%
[56]	mLSTM	99.9%	97.1%
[57]	SMOTE	94.81%	92.93%
	ADASYN	89.42%	91.54%
[58]	SVM-SMOT	93.51%	94.53%
	Borderline1-SMOTE	92.22%	92.60%
	Borderline2-SMOTE	90.23%	92.03%
Proposed Model	DT_Ori_MI	98.58%	49.21%
	DT_Ori_CFS	100.00%	99.99%
	DT_RUS_MI	100.00%	100.00%
	DT_RUS_CFS	100.00%	100.00%
	DT_CSL_MI	98.58%	49.21%
	DT_CSL_CFS	100.00%	99.99%
	DT_Smote_MI	97.98%	87.94%
	DT_Smote_CFS	100.00%	100.00%
	DT_RCS_MI	99.35%	93.83%
	DT_RCS_CFS	99.99%	99.98%
	LDA_Ori_MI	91.58%	0.00%
	LDA_Ori_CFS	91.58%	0.00%
	LDA_RUS_MI	72.38%	0.00%
	LDA_RUS_CFS	72.38%	0.00%
	LDA_CSL_MI	91.58%	0.00%
	LDA_CSL_CFS	91.58%	0.00%
	LDA_Smote_MI	61.86%	2.07%
	LDA_Smote_CFS	94.31%	69.11%
	LDA_RCS_MI	67.55%	0.00%
	LDA_RCS_CFS	89.50%	46.04%

CONCLUSION AND FUTURE WORKS

This study evaluates the impact of various data balancing techniques, RUS, CSL, Smote, RCS, combined with feature selection methods MIFS, CFS, and classification algorithms DT, LDA. This study concludes: 1) DT consistently outperformed LDA across all balancing methods, achieving higher accuracy, G-Mean, and other performance metrics; 2) CFS proved more effective than Mutual Information, especially when combined with SMOTE and RCS balancing techniques. These combinations resulted in nearly perfect G-Mean and accuracy, indicating excellent handling of imbalanced data; 3) Among balancing techniques, SMOTE and RCS showed the best performance, particularly for DT, as they effectively addressed class imbalance while maintaining generalization to testing data; 4) RUS was the least effective balancing method, often leading to poor generalization and significant performance drops, especially with LDA; 5) LDA demonstrated limitations in handling imbalanced datasets, failing to produce meaningful G-Mean and accuracy, even with advanced balancing techniques.

Based on the results of this study, there are several future works that can be done to further improve the effectiveness of intrusion detection systems for IoT networks: 1) Combining Advanced Balancing Techniques with more sophisticated oversampling and under sampling methods, such as Adaptive Synthetic Sampling (ADASYN) or generative adversarial networks (GANs) for synthetic data generation; 2) Performing feature engineering and dimensionality reduction with additional feature selection or extraction methods, such as Principal Component Analysis (PCA) or autoencoders, to improve model performance and reduce computational overhead.

REFERENCES

[1] J. Arents and M. Greitans, "Smart Industrial Robot Control Trends, Challenges and Opportunities within Manufacturing," *Applied Sciences*, vol. 12, no. 2, p. 937, 2022, doi: 10.3390/app12020937.

[2] A. A. Zaidan and B. B. Zaidan, "A review on intelligent process for smart home applications based on IoT: coherent taxonomy, motivation, open challenges, and recommendations," *Artificial Intelligence Review*, vol. 53, pp. 141–165, 2020, doi: 10.1007/s10462-018-9648-9.

[3] D. Oladimeji, K. Gupta, N. A. Kose, K. Gundogan, L. Ge, and F. Liang, "Smart Transportation: An Overview of Technologies and Applications," *Sensors*, vol. 23, no. 8, p. 3880, 2023, doi: 10.3390/s23083880.

[4] S. Rizvi, R. Orr, A. Cox, P. Ashokkumar, and M. R. Rizvi, "Identifying the attack surface for IoT network," *Internet of Things*, vol. 9, p. 100162, 2020, doi: 10.1016/j.iot.2020. 100162.

[5] J. Arshad, M. A. Azad, R. Amad, K. Salah, M. Alazab, and R. Iqbal, "A Review of Performance, Energy and Privacy of Intrusion Detection Systems for IoT," *Electronics*, vol. 9, no. 4, p. 629, 2020, doi: 10.3390/electronics9040629.

[6] A. Khraisat and A. Alazab, "A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges," *Cybersecurity*, vol. 4, no. 18, pp. 1–27, 2021, doi: 10.1186/s42400-021-00077-7.

[7] H. Alfares and O. Banimelhem, "Comparative Analysis of Machine Learning Techniques for Handling Imbalance in IoT-23 Dataset for Intrusion Detection Systems," in *2024 11th International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, 2024, pp. 112–119, doi: 10.1109/IOTSMS62296.2024.10710296.

[8] K.-A. Tait, J. S. Khan, F. Alqahtani, A. A. Shah, F. A. Khan, and M. U. Rehman, "Intrusion Detection using Machine Learning Techniques: An Experimental Comparison," in *2021 International Congress of Advanced Technology and Engineering (ICOTEN)*, 2021, pp. 1–10, doi: 10.1109/ICOTEN52080.2021.9493543.

[9] A. S. Dina, A. B. Siddique, and D. Manivannan, "Effect of Balancing Data Using Synthetic Data on the Performance of Machine Learning Classifiers for Intrusion Detection in Computer Networks," *IEEE Access*, vol. 10, pp. 96731–96747, 2022, doi: 10.1109/ACCESS.2022.3205337.

[10] A. El Hariri, M. Mouiti, O. Habibi, And M. Lazaar, "Improving Deep Learning Performance Using Sampling Techniques for IoT Imbalanced Data," *Procedia Computer Science*, vol. 224, pp. 180–187, 2023, doi: 10.1016/j.procs.2023.09.026.

[11] J. Jiang *et al.*, "A dynamic ensemble algorithm for anomaly detection in IoT imbalanced data streams," *Computer and Communications*, vol. 194, pp. 250–257, 2022, doi: 10.1016/j.comcom.2022.07.034.

[12] A. H. Butt, Z. Khan, A. Khan, H. Ghazanfar, R. Zgheib, and F. Kamalov, "Performance of Sampling Methods on Imbalanced Data: Comparative Analysis," in *2024 Advances in Science and Engineering Technology International Conferences (ASET)*, 2024, pp. 1–6, doi: 10.1109/ASET60340.2024. 10708760.

[13] T. Riston *et al.*, "Oversampling Methods for Handling Imbalance Data in Binary Classification," *Computational Science and Its Applications – ICCSA 2023 Work. ICCSA 2023*, pp. 3–23, 2023, doi: 10.1007/978-3-031-37108-0_1.

[14] M. Bakro, R. R. Kumar, M. Husain, Z. Ashraf, A. Ali, and S. I. Yaqoob, "Building a Cloud-IDS by Hybrid Bio-Inspired Feature Selection Algorithms Along With Random Forest Model," *IEEE Access*, vol. 12, pp. 8846–8874, 2024, doi: !0.1109/ACCESS. 2024.3353055.

[15] L. Xue and T. Zhu, "Hybrid resampling and weighted majority voting for multi-class anomaly detection on imbalanced malware and network traffic data," *Engineering Applications of Artificial Intelligence*, vol. 128, p. 107568, 2024, doi: 10.1016/j.engappai.2023.107568.

[16] A. Telikani, J. Shen, J. Yang, and P. Wang, "Industrial IoT Intrusion Detection via Evolutionary Cost-Sensitive Learning and Fog Computing," *IEEE Internet Things Journal*, vol. 9, no. 22, pp. 23260–23271, 2022, doi: 10.1109/JIOT.2022.3188224.

[17] B. S. Sharmila and R. Nagapadma, "RT-IoT2022 [Dataset]." *UCI Machine Learning Repository*, 2023, doi: 10.24432/C5P338.

[18] F. Alghayadh and D. Debnath, "A Hybrid Intrusion Detection System for Smart Home Security," in *2020 IEEE International Conference on Electro Information Technology (EIT)*, 2020, pp. 319–323, doi: 10.1109/EIT48999.2020.9208296.

[19] M. J. Gatea and S. M. Hameed, "An Internet of Things Botnet Detection Model Using Regression Analysis and Linear Discrimination Analysis," *Iraqi Journal of Science (IJS)*, vol. 63, no. 10, pp. 4534–4546, 2022, doi: 10.24996/ijs.2022.63.10.36.

[20] H. M. Rai and K. Chatterjee, "Hybrid CNN-LSTM deep learning model and ensemble technique for automatic detection of myocardial infarction using big ECG data," *Apple Intelligence*, vol. 52, pp. 5366–5384, 2021, doi: 10.1007/s10489-021-02696-6.

[21] D. Rani, N. S. Gill, P. Gulia, F. Arena, and G. Pau, "Design of an Intrusion Detection Model for IoT-Enabled Smart Home," *IEEE Access*, vol. 11, pp. 2509–52526, 2023, doi: 10.1109/ACCESS.2023.3276863.

[22] S. Abdelhamid, I. Hegazy, M. Aref, and M. Roushdy, "Studying The Impact Of Dataset Balancing On Machine Learning-Based Intrusion Detection Systems For Iot," *International Journal of Computational Intelligence*, vol. 24, no. 3, pp. 41–57, 2024, doi: 10.21608/ijicis.2024.317982.1352.

[23] G. Karatas, O. Demir, 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.

[24] R. R.-F. Vaishnavi Shanmugam and E. Hallaji, "Addressing Class Imbalance in Intrusion Detection: A Comprehensive Evaluation of Machine Learning Approaches," *Electronics*, vol. 14, no. 1, p. 69, 2025, doi: 10.3390/electronics14010069.

[25] A. A. Wardana, G. Kołaczek, A. Warzynski, and P. Sukarno, "Collaborative intrusion detection using weighted ensemble averaging deep neural network for coordinated attack detection in heterogeneous network," *International Journal of Information Security*, vol. 23, pp. 3329–3349, 2024, doi: 10.1007/s10207-024-00891-3.

[26] S. Bagui and K. L, "Resampling imbalanced data for network intrusion detection datasets," *Journal of Big Data*, vol. 8, no. 6, 2021, doi: 10.1186/s40537-020-00390-x.

[27] A. Vaishnavi, B. R. Ganesh, A. D. Reddy, and K. L. Kumar, "Ensemble-Learning-Based Deep Neural Network Attack Classification of Imbalanced IoT Intrusion Data," *International Journal of Information Technology and Computer Engineering*, vol. 12, no. 3, pp. 634–646, 2024.

[28] J. Song, X. Wang, M. He, and L. Jin, "CSK-CNN: Network Intrusion Detection Model Based on Two-Layer CNN for Handling Imbalanced Dataset," *Information*, vol. 14, no. 2, p. 130, 2023, doi: https://doi.org/10.3390/info14020130.

[29] A. G. Ayad, N. A. Sakr, and N. A. Hikal, "A hybrid approach for efcient feature selection in anomaly intrusion detection for IoT networks," *Journal of Supercomputer*, vol. 80, pp. 26942–26984, 2002, doi: 10.1007/s11227-024-06409-x.

[30] H. Zhang, L. Huang, C. Q. Wu, and Z. Li, "An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset," *Computer networks*, vol. 177, p. 107315, 2020, doi: 10.1016/j.comnet.2020.107315.

[31] M. A. Talukder *et al.*, "Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction," *Journal of Big Data*, vol. 11, no. 33, 2024, doi: 10.1186/s40537-024-00886-w.

[32] Z. Fan, S. Sohail, F. Sabrina, and X. Gu, "Sampling-Based Machine Learning Models for Intrusion Detection in Imbalanced Dataset," *Electronics*, vol. 13, no. 10, p. 1878, 2024, doi: 10.3390/electronics13101878.

[33] B. Weinger, J. Kim, A. Sim, M. Nakashima, N. Moustafa, and K. J. Wu, "Enhancing IoT anomaly detection performance for federated learning," *Digital Communication Networks*, vol. 8, no. 3, pp. 314–323, 2022, doi: 10.1016/j.dcan.2022.02.007.

[34] I. Sulistiyowati, A. R. Sugiarto, and J. Jamaaluddin, "Smart Laboratory Based On Internet Of Things In The Faculty Of Electrical Engineering, University Of Muhammadiyah Sidoarjo," in *IOP Conference Series: Materials Science and Engineering*, 2020, vol. 874, no. 1, p. 12007.

[35] O. D. Okey *et al.*, "BoostedEnML: Efficient Technique for Detecting Cyberattacks in IoT Systems Using Boosted Ensemble Machine Learning," *Sensors*, vol. 22, no. 19, p. 7409, 2022, doi: 10.3390/s22197409.
[36] B. R. Kikissagbe and M. Adda, "Machine Learning-Based Intrusion Detection Methods in IoT Systems: A Comprehensive Review," *Electronics*, vol. 13, no. 18, p. 3601, 2024, doi: 10.3390/electronics13183601.
[37] F. Thabtah, S. Hammoud, F. Kamalov, and A. Gonsalves, "Data imbalance in classification: Experimental evaluation," *Information Science*, vol. 513, pp. 429–441, 2020, doi: 10.1016/j.ins.2019.11.004.
[38] M. R. Rezaei-Dastjerdehei, A. Mijani, and E. Fatemizadeh, "Addressing Imbalance in Multi-Label Classification Using Weighted Cross Entropy Loss Function," in *2020 27th National and 5th International Iranian Conference on Biomedical Engineering (ICBME)*, 2020, pp. 333–338, doi: 10.1109/ICBME51989.2020.9319440.
[39] P. Thölke *et al.*, "Class imbalance should not throw you off balance: Choosing the right classifiers and performance metrics for brain decoding with imbalanced data," *Neuroimage*, vol. 277, p. 120253, 2023, doi: 10.1016/j.neuroimage.2023.120253.
[40] L. Wang, M. Han, X. Li, N. Zhang, and H. Cheng, "Review of Classification Methods on Unbalanced Data Sets," *IEEE Access*, vol. 9, pp. 64606–64628, 2021, doi: 10.1109/ACCESS.2021.3074243.
[41] N. V. Chereddy and B. K. Bolla, "Evaluating the Utility of GAN Generated Synthetic Tabular Data for Class Balancing and Low Resource Settings," *Multi-disciplinary Trends in Artificial Intelligence*, vol. 14078, 2023, doi: 10.1007/978-3-031-36402-0_4.
[42] K. Pasupa, S. Vatathanavaro, and S. Tungjitnob, "Convolutional neural networks based focal loss for class imbalance problem: a case study of canine red blood cells morphology classification," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, pp. 15259–15275, 2020, doi: 10.1007/s12652-020-01773-x.
[43] H. Patel, D. S. Rajput, G. T. Reddy, C. Iwendi, A. K. Bashir, and O. Jo, "A review on classification of imbalanced data for wireless sensor networks," *International Journal of Distributed Sensor Networks (IJDSN)*, vol. 16, no. 4, pp. 1–15, 2020, doi: 10.1177/1550147720916404.
[44] G. Eom and H. Byeon, "Searching for Optimal Oversampling to Process Imbalanced Data: Generative Adversarial Networks and Synthetic Minority Over-Sampling Technique," *Mathematics*, vol. 11, no. 16, p. 3605, 2023, doi: 10.3390/math11163605.
[45] D. Dablain, B. Krawczyk, and N. V. Chawla, "DeepSMOTE: Fusing Deep Learning and SMOTE for Imbalanced Data," *IEEE Trans. Neural Networks Learning System*, vol. 34, no. 9, pp. 6390–6404, 2023, doi: 10.1109/TNNLS.2021.3136503.
[46] A. Demirpolat, A. K. Sarica, and P. Angin, "ProtÉdge: A few-shot ensemble learning approach to software-defined networking-assisted edge security," *Transactions on Emerging Telecommunications Technologie*, vol. 32, no. 6, p. e4138, 2021, doi: 10.1002/ett.4138.
[47] A. G. Famera, R. M. Shukla, and S. Bhunia, "Cross Device Federated Intrusion Detector for Early Stage Botnet Propagation in IoT," in *2024 IEEE International Systems Conference (SysCon)*, 2024, pp. 1–8, doi: 10.1109/SysCon61195.2024.10553450.
[48] Q. A. Al-Haija and M. Al-Dala'ien, "ELBA-IoT: An Ensemble Learning Model for Botnet Attack Detection in IoT Networks," *Journal of Sensor and Actuator Networks*, vol. 11, no. 1, p. 18, 2022, doi: 10.3390/jsan11010018.
[49] F. Rashid, S. M. A. Gargaare, A. H. Aden, and A. Abdi, "Machine Learning Algorithms for Document Classification: Comparative Analysis," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 4, pp. 260–265, 2022, doi: 10.14569/IJACSA.2022.0130430 PDF.
[50] N. Aslam *et al.*, "Interpretable Machine Learning Models for Malicious Domains Detection Using Explainable Artificial Intelligence (XAI)," *Sustainability*, vol. 14, no. 12, p. 7375, 2022, doi: 10.3390/su14127375.
[51] N. Mylonas, I. Mollas, N. Bassiliades, and G. Tsoumakas, "Exploring local interpretability in dimensionality reduction: Analysis and use cases," *Expert Systems With Applications*, vol. 252, no. Part A, p. 124074, 2024, doi: 10.1016/j.eswa.2024.124074.
[52] J. S. Kushwah, A. Kumar, S. Patel, R. Soni, A. Gawande, and S. Gupta, "Comparative study of regressor and classifier with decision tree using modern tools," *Materials Today: Proceedings*, vol. 56, no. 6, pp. 3571–3576, 2022, doi: 10.1016/j.matpr.2021.11.635.
[53] M. Maddeh, S. Ayouni, S. Alyahya, and F. Hajjej, "Decision tree-based Design Defects Detection," *IEEE Access*, vol. 9, pp. 71606–71614, 2021, doi: 10.1109/ACCESS.

2021.3078724.
[54] C. Yan *et al.*, “Self-weighted Robust LDA for Multiclass Classification with Edge Classes,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 12, no. 1, pp. 1–19, 2020, doi: 10.1145/3418284.
[55] D. V. Premalatha and S. Ramanujam, “Securing the internet of things frontier: a deep learning ensemble for cyber-attack detection in smart environments,” *International Journal of Artificial Intelligence Research (IJAIR)*, vol. 13, no. 4, p. 4736~4746, 2024, doi: 10.11591/ijai.v13.i4.pp4736-4746.
[56] V. Dutta, M. I Chora´s, M. Pawlicki, and R. I Kozik, “Detection of Cyberattacks Traces in IoT Data,” *Journal of Universal Computer Science*, vol. 26, no. 11, pp. 1422–1434, 2020, doi: 10.3897/jucs.2020.075.
[57] R. Qaddoura, A. M. Al-Zoubi, I. Almomani, and H. Faris, “A Multi-Stage Classification Approach for IoT Intrusion Detection Based on Clustering with Oversampling,” *Applied Sciences*, vol. 11, no. 3022, 2021, doi: 10.3390/app11073022.
[58] S. Bibi, N. Zulkifli, G. A. Safdar, and S. Iqbal, "Support Vector Machine (SVM) based Detection for Volumetric Bandwidth Distributed Denial of Service (DVB-DDOS) Attack within Gigabit Passive Optical Network," *Sinergi (Indonesia)*, vol. 29, no. 1, pp. 185-196, 2025, doi: 10.22441/sinergi.2025.1.017