International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 17 (2025) pp. 277–293

Article

Hybrid Sampling Approach to Enhance Intrusion Detection System in IoT Networks

Ralte Lalduhsaka [1,*], Ajoy Kumar Khan [1] and Rothangliana Chawngsangpuii [2]

[1] Department of Computer Engineering, School of Engineering and Technology, Mizoram University, Aizawl 796004, Mizoram, India; ajoyiitg@gmail.com
[2] Department of Information Technology, School of Engineering and Technology, Mizoram University, Aizawl 796004, Mizoram, India; mzut126@mzu.edu.in
* Correspondence author: duskeralte777@gmail.com

Received date: 17 October 2024; Accepted date: 27 February 2025; Published online: 27 March 2025

Abstract: The growing adoption of Internet of Things (IoT) networks has introduced new challenges in cybersecurity and intensified the need for robust security mechanisms, particularly for detecting and mitigating network intrusions. Given the massive amount of data generated by IoT devices and the highly imbalanced nature of intrusion datasets, traditional detection methods often struggle with accuracy, especially in identifying minority class attacks. This paper presents a hybrid sampling approach designed to enhance Intrusion Detection Systems (IDS) for IoT environments by addressing data imbalance. The proposed method combines the Near Miss-1 undersampling technique with Synthetic Minority Over-sampling Technique (SMOTE) to create a more balanced dataset without significant information loss. Using the Bot-IoT dataset, we evaluate the proposed approach across various machine learning algorithms, assessing their performance using key metrics including accuracy, False Positive Rate (FPR), and False Negative Rate (FNR). The imbalance Bot-IoT dataset, which was originally 99.99:0.01 ratio has been resampled to 80:20 ratio. The experimental results demonstrate that Random Forest (RF) and K-Nearest Neighbor (KNN) excel across all metrics achieving 99.98% 99.96% accuracy respectively and FPR of 0.02% and 0.06% respectively. The performance of the proposed approach is compared with other state-of-the-art methods, focusing on binary classification tasks and the ability to detect attacks while minimizing FPR. This study provides valuable insights into the application of hybrid sampling techniques for improving the detection capabilities of IDS in IoT networks.

Keywords: intrusion detection; Bot-IoT; hybrid-sampling; near miss; SMOTE; machine learning

1. Introduction

The Internet of Things (IoT) has emerged as a transformative paradigm in the digital landscape, comprising a network of interconnected physical objects or things. These entities, equipped with sensors, software, and other technological components, facilitate the collection, exchange, and utilization of data. The IoT ecosystem encompasses a diverse array of devices, including household appliances, vehicles, industrial equipment, wearable gadgets, and numerous others [1]. The defining characteristic of IoT is the autonomous communication and interaction between these objects and centralized systems via the internet, often without direct human intervention. The significance of IoT lies in its capacity to interconnect myriad everyday objects, devices, and systems through the internet, enabling data-driven insights, automation, and remote control. This technological revolution enhances efficiency, user experiences, and decision-making across various sectors, profoundly impacting manufacturing, healthcare, agriculture, and urban planning. Moreover, IoT contributes significantly to global sustainability, safety, and innovation efforts.

However, the proliferation of IoT devices and networks has introduced new security challenges, necessitating robust protective measures. In this context, Intrusion Detection Systems (IDS) have become

a critical component of cybersecurity frameworks. An IDS is a sophisticated security mechanism designed to monitor and analyze network communications, system operations, and user interactions. Its primary objective is to identify and address instances of unauthorized or suspicious activities that may indicate a security breach or cyberattack [2]. IDS is highly effective in detecting and preventing network-based attacks like viruses, worms, and data theft. IDS operate by comparing observed patterns against known attack signatures, abnormal behavior, or predefined rules to identify potential threats and vulnerabilities within a computer network. The importance of IDS in cybersecurity cannot be overstated. It plays a proactive role by continuously monitoring network traffic and system activities, enabling swift identification and response to unauthorized or suspicious behavior. IDS offer several key benefits, including: Early threat detection, Reduced system downtime, Insider threat mitigation, Compliance support and Customizable defense mechanisms. These features collectively fortify cybersecurity measures, maintain business continuity, and safeguard sensitive data against a broad spectrum of cyber threats.

Machine learning (ML) has significantly enhanced the capabilities of the IoT and IDS, particularly in addressing the massive volume of data generated within IoT environments and the increasing complexity of cyberattacks. In IoT, ML enables real-time data processing, anomaly detection, and predictive analytics, empowering devices to make autonomous decisions. For instance, ML can analyze patterns in IoT sensor data to detect system faults, predict maintenance needs, or optimize resource usage, which is crucial in sectors like smart healthcare, energy, and transportation [3]. Furthermore, ML allows IoT systems to adapt dynamically, learning from new data streams to improve the precision and efficiency of operations without the need for constant human intervention. ML has revolutionized threat detection by enabling systems to learn from historical data and detect both known and unknown attacks. Traditional rule-based IDS struggle to keep pace with the rapidly evolving nature of cyber threats. However, ML techniques, such as supervised learning, unsupervised learning, and deep learning, allow IDS to detect subtle deviations in network traffic or system behavior, flagging potential security breaches that may bypass conventional detection methods [4]. The scalability of ML solutions makes them ideal for deployment in large-scale IoT networks, where the volume and diversity of data are too overwhelming for traditional IDS to handle effectively. By continuously adapting to emerging threats, ML-powered IDS ensure a higher level of security, safeguarding IoT systems against an ever-changing threat landscape. In this study, we utilize the Bot-IoT dataset [5], specifically created for intrusion detection in IoT environments, to evaluate the effectiveness of IDS. As the most prominent publicly available IoT dataset for intrusion detection, Bot-IoT is expected to yield reliable experimental results. We employ ML techniques to enhance detection results and conduct a detailed analysis of the dataset. Our research makes several significant contributions:

- We perform a comprehensive analysis of the Bot-IoT dataset, uncovering its characteristics and underlying patterns.
- We propose a novel Hybrid Sampling method that combines Near Miss [6] and Synthetic Minority Over-sampling Technique (SMOTE) [7] algorithms to address severe class imbalance, effectively enhancing the representation of minority class instances.
- We rigorously evaluate our methodology on the Bot-IoT dataset, demonstrating superior performance compared to existing IDS approaches.

This study addresses a critical gap in the existing literature by proposing a hybrid sampling method that effectively balances the extreme class imbalance in IoT intrusion detection dataset without compromising data quality. Our approach uniquely combines undersampling of the majority class with oversampling of the minority class, leading to improved performance across multiple ML algorithms.

The remainder of this paper is structured as follows: Section 2 discusses prior studies about IDS in the context of IoT. Section 3 provides a detailed description and analysis of the Bot-IoT dataset and outlines the proposed methodology, including the methods and strategies adopted in this study. Section 4 presents the experimental results and analysis. Section 5 discusses the relevance of this work and demonstrates how it differs from various other state-of-the-art methods. Finally, Section 6 concludes the study and suggests directions for future research in this field.

2. Related Works

The application of ML in the context of IDS within the realm of the IoT has been ongoing for several decades, with a plethora of research endeavors dedicated to this domain. An IDS is a software or hardware component. It monitors the traffic of the network and if it finds a suspicious packet, it alerts the user or system with an alarm. There are two main divisions of IDS: Host-based IDS which needs to be installed on every single computer system (host) and Network-based IDS which is installed for a whole network [8]. ML plays a pivotal role in IDS by enabling the systems to autonomously learn from and adapt to

network data, enhancing their ability to detect and respond to a broad spectrum of cyber threats. ML empowers IDS to discern patterns of normal and abnormal behaviors, making them effective at detecting zero-day attacks [9]. This technology facilitates real-time analysis, automated response, and efficient prioritization of alerts, thereby augmenting the overall efficacy of IDS in safeguarding network integrity and mitigating potential security breaches.

Baich, Marwa, et al. [10] present a state-of-the-art study on IoT-NIDS utilizing ML techniques with the objective of identifying the most effective and commonly used ML approaches. Experiments were conducted to compare ML techniques using the NSL-KDD dataset. Following preprocessing, different feature selection techniques were applied. The models employed in this study include "Decision Tree (DT), Random Forest (RF), Naive Bayes (NB), and Support Vector Machine (SVM)". RF algorithm demonstrated the highest efficiency, with 99.13% accuracy. Mahmoud, Mohamed et al. [11] introduce a novel ML approach named "AE-LSTM" for intrusion detection, employing a 6-layer Autoencoder (AE) with Long Short-Term Memory (LSTM), proving to be highly effective in identifying anomalies. AE-LSTM optimally utilizes the finest reconstruction function, one of the main areas where the differentiation between normal network traffic and other abnormal activity is essential is based on the use of the "NSL-KDD test dataset" for evaluation, in which the performance of the proposed AE-LSTM is significantly higher compared to the use of the traditional methods and achieving the micro and weighted F1-score of 98.69% and 98.70% respectively. Liu and Du [12] discuss feature selection using the Genetic Algorithm applied to detect botnet using the Bot-IoT dataset. This approach is able to achieve the intended aim of reducing the number of features from 40 to 6 in order to achieve a detection accuracy of 99.98% as well as an F1-score of 99.63%, the method outperforms alternatives in training time and accuracy. However, limitations include potential accuracy variations due to diverse factors and the impact of different feature selection processes and classification models on final performance. Even though feature selection reduces dataset complexity and improves the performance of the ML model, the risk of overfitting remains high due to extreme class imbalance. Bakaa and Musawi [13] introduce a novel IDS leveraging RQA which is a non-linear statistical technique. They use the UNSW-NB15 dataset to evaluate their proposed method, identifying the most important features, thus reducing computational cost and also enhancing the anomaly detection. The proposed method achieves high accuracy and F-score using only a single feature, outperforming many existing IDS. Hnamte et al. [14] proposed a DNN based method for detection and mitigation of DDoS using the InSDN dataset and CICIDS2018 dataset achieving 99.98% and 100% accuracy with low loss rate, their results highlight the potential of DNN particularly for the detection of DDoS attack.

Fernando et al. [15] evaluates the performance of unsupervised ML algorithms for unbalanced data using the BoT-IoT dataset. The authors analyze K-means++, DBSCAN, Local Outlier Factor (LOF), and Isolation Forest (I-forest) based on metrics like purity, homogeneity, and adjusted mutual information. K-means++ achieves 95% purity, while I-forest demonstrates best efficiency, using only 10% of CPU resources compared to 16% for other algorithms. The study highlights the potential of unsupervised learning for intrusion detection in resource-constrained environments, advocating for future research in hybrid approaches and real-world validation. Tsogbaatar et al. [16] have presented DeL-IoT, a reliable deep ensemble learning model implemented for IoT networks based on SDN for the detection of anomalous behaviors and the prediction of occurrences. The DeL-IoT framework is composed of three main components: it includes anomalies detection, an intelligent and efficient traffic flow management, and identifying the future condition of devices or Systems. The experiments on testbed and benchmark dataset manifest that this approach gets around 3% more accuracy as compared to isolated models, even within a 1% imbalanced dataset. Notably, their model exhibits superior performance compared to relevant competing models showcased in the existing literature.

Roopak et al. [17] have proposed IDS which integrates deep learning and multi-objective optimization to detect Distributed Denial of Service (DDoS) attacks in IOT network. Their method comprises of a Jumping Gene-based NSGA-II optimization approach to dimensionality reduction of data with CNNs and LSTM networks for classifying the attacks. When experiments where performed on the CICIDS2017 dataset especially in identifying DDoS attacks, it has attained high accuracy of 99.03% and training time which was also cut down 5-folds. This approach proved to be more effective than other existing approaches that were practiced in the field. Further, an ensemble model employing Quadratic Discriminant Analysis (QDA), Naive Bayes & ID3 algorithms was proposed for anomaly detection on the CICIDS2017 and CICIDS2018 datasets where higher accuracy was marked along with lesser false alarms, when compared to the individual algorithms [18]. Other datasets that are commonly used for IDS solution include NSL-KDD, UNSW-NB15, CICIDS2017 [19]. Though these datasets do not represent a pure IoT environment, they share similarities in data types and patterns with datasets specifically designed for IDS in IoT networks.

Alosaimi and Saad [20] proposes a new approach involving deep learning with three levels has been

developed to quickly identify attacks on IoT networks. It also shows detections performance improvement by using the proposed method on the Bot-IoT dataset in comparison to prior approaches, with potential extensions to enhance security across various IoT applications. The research focuses on discovering IoT network attacks using ML techniques, leveraging the Bot-IoT dataset for its attack diversity and network protocols. Two main approaches are employed: data size reduction and addressing data imbalance using SMOTE. 100% accuracy is achieved using the Ensemble bag algorithm with the DT algorithm closely behind at 99.9%.

Numerous researchers have employed various Intrusion Detection Datasets to evaluate IDS in IoT networks. While these datasets encompass a wide array of attacks, including those pertinent to IoT networks, utilizing datasets specifically tailored to IoT environments would enhance the reliability of evaluations. Leevy et al. [21] applied elementary ML techniques to the Bot-IoT dataset, while also Ibrahimi and Hafsa [22] employed conventional ML techniques on the same dataset, employing both binary and multi-class classifications. Both studies assert enhancements in detection accuracy and other performance metrics as a result of their methodologies. Aruna and Karthik [23] introduced a hybrid sampling and anomaly detection method for predicting diabetes. The combination of SMOTE oversampling and ENN undersampling has proven effective in enhancing prediction accuracy. Table 1 gives the summary of the related works.

Table 1. Summary of Related Works.

Reference	**Dataset**	**Methodology**	**Findings**	**Performance**
Mahmoud et al. [11]	NSL-KDD	AE-LSTM: 6-layer Autoencoder with LSTM.	Effective anomaly detection with improved reconstruction functions.	F1-Score (micro): 98.69%, Weighted: 98.70%.
Liu & Du [12]	Bot-IoT	Genetic Algorithm for feature selection.	Reduced features from 40 to 6, improving detection accuracy and efficiency.	Accuracy: 99.98%, F1-Score: 99.63%.
Tsogbaatar et al. [16]	IoT SDN testbed, benchmark	DeL-IoT: Deep ensemble learning model.	Achieved ~3% higher accuracy than isolated models even on imbalanced datasets.	Superior to isolated models.
Roopak et al. [17]	CICIDS2017, CICIDS2018	CNN and LSTM with Jumping Gene-based NSGA-II optimization. Ensemble QDA, NB, ID3.	High accuracy in DDoS detection; reduced training time by 5-fold.	Accuracy: 99.03%.
Alosaimi & Saad [20]	Bot-IoT	Deep learning with Ensemble bagging and SMOTE for imbalance handling.	Achieved 100% accuracy; improved IoT attack detection.	Accuracy: 100%.
Leevy et al. [21]	Bot-IoT	Elementary ML techniques.	Enhanced detection accuracy and performance metrics for IoT attacks.	Improvement noted, but no specific values.
Ibrahimi & Hafsa [22]	Bot-IoT	Binary and multi-class ML techniques.	Improved accuracy and performance on binary/multi-class IoT classifications.	Specific metrics not detailed.

Aruna & Karthik [23]	Custom dataset	Hybrid sampling: SMOTE and ENN undersampling.	Enhanced prediction accuracy in anomaly detection (non-IoT use case: diabetes).	Effective prediction improvement.

3. Proposed Methodology

We have introduced an IDS model utilizing ML specifically designed for IoT environments. This model incorporates a hybrid sampling method and binary classification techniques. In the basic architecture of IDS in a network, he IDS monitors all incoming and outgoing packets and sends an alarm to the systems within the organization if there is any indication of a potential attack in the incoming packet [2].

The IDS monitors all the incoming packets and decides whether the packets are malicious or not as shown in Figure 1. The IDS on detecting malicious nodes among the traffic, sends alarm to each of the system in the network. The proposed method is shown in Figure 2.

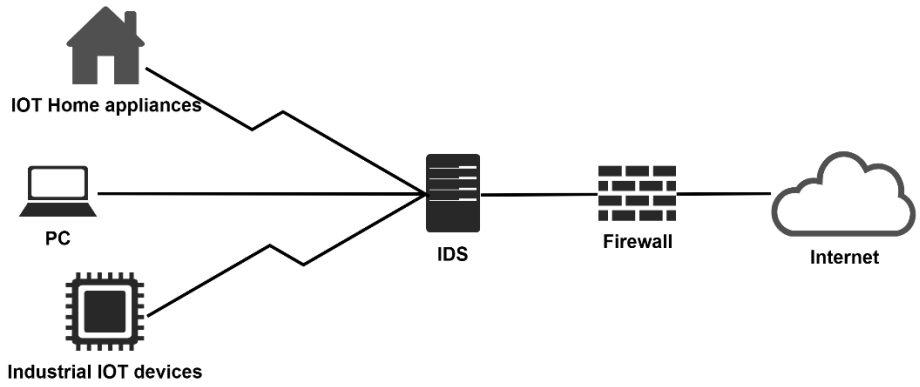

Figure 1. Basic Architecture of IDS in an IOT network.

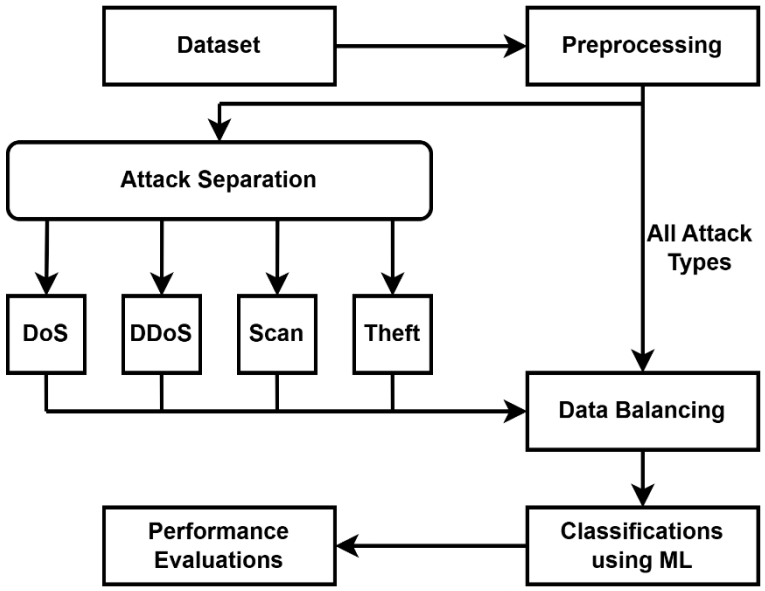

Figure 2. Proposed methodologies.

In the proposed method, we utilize the Bot-IoT dataset for evaluation due to its representation nature of IoT environments. However, this dataset presents a significant challenge: an extreme class imbalance, where majority of the class vastly outnumbers the minority class which is likely to lead the decisions made by the classification algorithm unreliable. To address this, we implement a hybrid sampling method using Near Miss-1 for undersampling the majority class and SMOTE for oversampling the minority class,

thereby balancing the dataset to approximately an 80:20 ratio, where 80% represents attacks and 20% represents benign instances. The Bot-IoT dataset contains four different types of attacks, for each of which a subset of the dataset is created as shown in Figure 2. These subsets of dataset also undergo the data balancing process with the same ratio. Binary classification is then performed on each subset of data as well as the full dataset using selected ML algorithms. The 80:20 ratio was chosen for preserving the majority of attack instances while significantly increasing the representation of normal instances. This ratio is sufficient for effective learning of both classes without overfitting to the minority class or losing important information from the majority class. Alternative ratios were explored, but 80:20 provided the best trade-off between class balance and maintaining dataset integrity.

3.1. Dataset Description and Analysis

The Bot-IoT dataset was established in 2018, it was created by simulating a practical network environment within UNSW Canberra's Cyber Range Lab, dedicated to Intrusion Detection research. It is specifically designed to simulate attacks on IoT devices. This dataset is relevant and offers realistic attack scenarios for IoT environment. It includes wide range of attacks and more realistic anomalies that are specific to IoT. The original PCAP file is around 70GB in size.

In the Bot-IoT dataset repository, there are 74 CSV files containing all the records and attributes, totaling approximately 17GB in size. There are more than 73 million records with 35 attributes in the CSV files. Additionally, there is a 5% dataset comprising 5% of each class label. This 5% dataset is mostly used by researchers due to its smaller size. Table 2 shows the numerical detail of the Bot-IoT dataset.

Table 2. Numerical Analysis of Bot-IoT dataset.

Category	**Subcategory**	**No. of records**
Benign	-	9543
Information Gathering (Scan)	Service Scanning OS Fingerprint	1463364 358275
DDoS	DDoS TCP DDoS UDP DDoS HTTP	19547603 8965106 19771
DoS	DoS TCP DoS UDP DoS HTTP	12315997 20659491 29706
Information Theft	Key logging Data Theft	1469 118

Within this dataset, a spectrum of attacks is encompassed, which can be divided into four main categories;

- Information Gathering (Scan): Information gathering, often referred to as scanning, is a type of cyber-attack where an attacker attempts to gather information about a target system, network, or organization. The goal of this attack is to collect data that can be used for malicious purposes, such as planning a more advanced attack or exploiting vulnerabilities [24].
- DoS: A Denial of Service attack is an attempt to make systems in the network to be unavailable by relaying high traffic or by sending packets that can cause the system to crash [25].
- DDoS: It is a Distributed DoS in which attack is launch from multiple system which results in larger flood with higher capability of breaching the security [26].
- Information Theft: It focuses on stealing sensitive or confidential information from an organization, individual, or system. The attackers aim to gain unauthorized access to data and then exfiltrate or steal it for various purposes, which may include financial gain, espionage, or identity theft [27].

As shown in Table 1, the number of "Benign" instances is extremely small compared to the number of "Attack" instances. Benign instances comprise only around 0.013% of the entire dataset, indicating a significant imbalance. Table 3 contain the description of 35 features of the Bot-IoT dataset.

Table 3. Feature descriptions of Bot-IoT dataset.

Features	**Description**	**Features**	**Description**
pkSeqID	Packet sequence Identity	stime	Record state time
flgs	Flow state flag in transaction	proto	Transaction protocol in network flow

saddr	Source address	sport	Source port number
daddr	Destination address	dport	Destination port number
pkts	Total count of packets	bytes	Total number of bytes
state	Transaction state	ltime	Record last time
seq	Argus sequence number	dur	Record Total duration
mean	Average duration of records	stddev	Standard deviation of duration of records
smac	Source MAC address	dmac	Destination MAC address
sum	Total duration of records	min	Minimum duration of records
max	Maximum duration of records	soui	Source organizational unique identity
doui	Destination organizational unique identity	sco	Source IP country code
dco	Destination IP Country code	spkts	Source-destination packet count
dpkts	Destination-source packet count	sbytes	Source-destination transaction bytes
dbytes	Destination-source transaction bytes	rate	Total packet per second
srate	Source-destination packet per second	drate	Destination-source packet per second
attack	Class label	category	Traffic category
subcategory	Traffic subcategory		

The 35 features of the Bot-IoT dataset contributes uniquely to the identification of specific attack types. For instance, features such as “pkts” and “bytes” are particularly useful for detecting volumetric attacks like DDoS and DoS, as they reflect the volume of traffic. Similarly, the “proto” feature, which indicates the protocol type (e.g., TCP, UDP), helps distinguish between different attack vectors. Features like “state” and “dur” provide insights into the connection state and duration, which are critical for identifying anomalies in network behavior. By leveraging these features, the proposed IDS can effectively differentiate between normal traffic and various attack types. Either “attack”, “category” or “subcategory” can be used as the class label, and in this study, we are using “attack” as the class label as it contains the information of the records being an “attack” or “benign”. After studying the features and their corresponding descriptions, it is evident that some features are derived from others. Additionally, certain features consist of sequential numbers and addresses, which are unlikely to contribute to the classification results.

3.2. Data Pre-processing

The 74 CSV files of Bot-IoT dataset is combined into a single CSV file to make it more suitable for further processing. In this section, we are highlighting the pre-processing steps performed in this study.

- Handling Missing Values: The features “sport” and “dport” where feature “proto” (Transaction protocol in network flow) is “arp” are all empty. These empty values are filled with ’0’. There are also a few missing values that are removed from the dataset.
- Data Cleaning: Some instances exhibit peculiar characteristics compared to others, such as the presence of the feature “proto” being “ipv6-icmp”; or the feature state being “CLO”, “RSP”, “PAR” and so forth. These instances have low occurrences but have the potential to negatively impact the classification results of ML algorithms and are consequently removed from the dataset. In addition, three instances with characteristic data type are identified within the numerical type feature “dport”. These instances are also removed from the dataset.
- Encoding and Normalization: Features like “proto”, “state”, and “flgs” are in nominal categorical format. Onehot encoding [28] is applied to the instances of these features to ensure suitability for ML classifiers. The entire dataset is then normalized [29] to a common scale so that the ML model can evaluate the data effectively, without being biased by differences in feature scales.
- Data separation: The dataset comprises 4 types of attacks which can further be categorized into 10 different types of sub-attacks. A subset of the dataset is created for each type of attack along with the normal (benign) instances in the ratio 80:20. To achieve this ratio, the proposed hybrid sampling

method is used. In addition to these subsets, the main (full) dataset is also resampled in the ratio 80:20.

- Feature cleaning: The features “smac”, “dmac”, “soui”, “doui”, “sco”, and “dco” are all empty features and thus are removed. The features “pkSeqID” and “seq” represent the sequence numbers of the rows and flows, the features “saddr” and “daddr” represent the address of the machines, and lastly the features “category” and “subcategory” represent the attack types. It is imperative not to include these features in the model as they may inadvertently reveal class label information due to the sequential arrangement of classes in the dataset.

3.3. Hybrid Sampling Method for Data Balancing

The Bot-IoT dataset initially contains 35 attributes and 73,370,443 records, with 9,543 records labeled as normal and the remainder as attacks. Following pre-processing, the dataset comprises 37 attributes and 73,359,273 records, with 9,385 normal records. This step ensures that redundant and irrelevant features are removed while preserving relevant information. Denote the initial dataset as $D = \{(x_i, y_i)\}_{i=1}^{n}$, where x_i represents the feature vector, and $y_i \in \{0,1\}$ represents the class label (0 for normal, 1 for attack).

While numerous studies focus on botnet detection models, only a few integrate feature engineering to mitigate issues like duplication and multicollinearity in large datasets. Multicollinearity can be described as a condition where one feature is highly linearly dependent on another. Mathematically, for two features x_j and x_k, multicollinearity exists if:

$$\text{corr}\left(x_j, x_k\right) \approx 1 \text{ or } \beta_j x_j + \beta_k x_k + \epsilon = 0 \tag{1}$$

where $\text{corr}\left(x_j, x_k\right)$ is the Pearson correlation coefficient between feature j and k, β_j and β_k are coefficients, and ϵ is an error term.

Failure to handle such correlations will lead to ‘overfitting' [30], where the model performs correct on the “training data” but not on “unseen data”. Overfitting is mathematically represented by minimizing the empirical risk R_{emp} on the training data but leading to “poor generalization error R_{gen}":

$$R_{emp}(f) = \frac{1}{n}\sum_{i=1}^{n} L(f(x_i), y_i) \text{ but } R_{gen}(f) > R_{emp}(f) \tag{2}$$

where $L(f(x_i), y_i)$ is the “loss function” used to measure the error between the predicted value $f(x_i)$ and the true value y_i.

Class Imbalance in the Bot-IoT dataset is another significant challenge, with a skewed distribution between normal and attack records. Let the number of total records in the dataset be n, where n_0 and n_1 represent the number of normal and attack instances, respectively:

$$n_0 = 9{,}385,\ n_1 = 73{,}359{,}273 - 9{,}385 = 73{,}349{,}888 \tag{3}$$

This results in a highly imbalanced dataset, where:

$$\frac{n_0}{n_1} \approx \frac{9{,}385}{73{,}349{,}888} \approx 0.00013 \tag{4}$$

Researchers have employed various methods to tackle the class imbalance inherent in the Bot-IoT dataset. In this study, we propose a hybrid sampling method that sequentially utilizes Near Miss-1 and SMOTE technique on the dataset, as shown in Figure 3.

In this hybrid sampling method, Near Miss-1 and SMOTE are performed sequentially to achieve a more balanced dataset while also preserving the quality of the data. The Near Miss-1 algorithm basically finds samples belonging to the majority class most similar to the ones in the minority class. The selection is purely based on the extending of majority class examples which are nearest to the minority class in order to retain the informational nature of data. This approach ensures that important patterns and characteristics present in the majority class are not lost during the undersampling process.

The algorithm for Near Miss-1 is depicted in Algorithm 1. This undersampling step does not alter the number of instances in the minority class but decreases the number of instances in the majority class to a predetermined threshold. For each minority class instance $x_0 \in X_0$, the k-nearest neighbors from the majority class are selected based on the Euclidean distance:

$$d(x_0, x_1) = \sqrt{\sum_{i=1}^{d} (x_{0i} - x_{1i})^2} \quad (5)$$

where x_0 is a minority class instance, x_1 is a majority class instance, and d is the dimensionality of the feature space. The algorithm selects the k-nearest neighbors $N_k(x_0) \subseteq X_1$ for each $x_0 \in X_0$, preserving only those majority instances that are closest to the minority class.

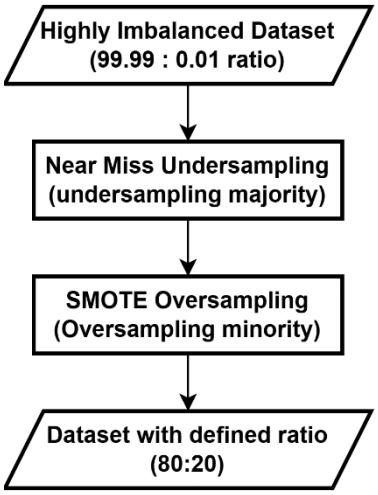

Figure 3. Flow of proposed Hybrid Sampling.

This undersampling process does not alter the number of instances in the minority class, while the instances in the majority class are reduced to a specified threshold. The resulting dataset can be represented as:

$$D_{\text{balanced}} = \{(x_i, y_i) | x_i \in X_0 \cup N_k(x_0), y_i \in \{0,1\}, |N_k(x_0)| \leq t\} \quad (6)$$

After applying Near Miss-1 technique, the dataset become more balanced than before, but is still highly imbalanced for ML techniques. To further balance the dataset, SMOTE is applied to the minority class. For each minority class instance $x_0 \in X_0$, synthetic samples are generated by interpolating between x_0 and one of its k-nearest neighbors x_0^k. The synthetic sample $x_{\text{synthetic}}$ is computed as:

$$x_{\text{synthetic}} = x_0 + \lambda(x_0^k - x_0),\ \lambda \in [0,1] \quad (7)$$

where λ is a random value between 0 and 1, and x_0^k is a randomly selected k-nearest neighbor from the minority class. This process generates synthetic samples, increasing the number of minority instances and balancing the dataset.

The hybrid approach of applying Near Miss-1 and SMOTE sequentially ensures that the dataset is now more balanced and also that the important patterns in the majority class are not lost during undersampling, while SMOTE enriches the minority class with synthetic samples. The resulting resampled dataset is then used for training ML classifiers, ensuring more accurate and reliable classification results.

Algorithm 1. Near Miss-1 Undersampling

Input: Imbalanced dataset (X,y)

Output: Resampled dataset ($X_{\text{resampled}}$, $y_{\text{resampled}}$)

1: Identify Minority and Majority Class instances

- Extract Minority class instances X_{minority} and majority class instances X_{majority}
- Extract corresponding labels y_{minority} and y_{majority}

2: Compute Nearest Neighbors for each minority class instances $x_m \in X_{\text{minority}}$

- For each x_m, calculate distance $D(x_m, x_{Mj}) = ||x_m - x_{Mj}||$ for every $x_{Mj} \in X_{\text{majority}}$

3: Identify k-Nearest Neighbor

- For each minority class instance x_m, identify the set of k-Nearest Neighbor $N_k(x_m)$ from majority class based on computed distances
- $N_k(x_m)=\{x_{Mj} \in X_{majority} \mid \text{rank}(D(x_m,x_{Mj})) \leq k$

4: Select and Combine

- Initialize $X_{resampled}=X_{minority}$ and $y_{resampled}=y_{minority}$
- For each $x_m \in x_{minority}$, add x_m and its k-nearest neighbor $N_k(x_m)$ to $X_{resampled}$
- And the corresponding labes to $y_{resampled}$

5: Combine all selected instances and their corresponding labels to form the final resampled dataset ($X_{resampled}$,$y_{resampled}$)

For every record in the minority class, SMOTE then find 'k' nearest neighbors in the feature space. It then picks one of these neighbors at random and then works out the difference in the features of the original instance and the chosen neighbor. New synthetic samples are then created along the line joining the original instance to the selected neighbor. These synthetic samples are incorporated into the dataset thus creating an increased representation of the minority class. The algorithm for SMOTE is depicted in Algorithm 2.

Algorithm 2. SMOTE Oversampling

Input: Resampled dataset ($X_{resampled}$, $Y_{resampled}$)

Output: Augmented dataset ($X_{augmented}$,$Y_{augmented}$)

1: Initialize empty arrays ($X_{augmented}$,$Y_{augmented}$)

2: Identify minority class instances $X_{minority}$ and their corresponding label $Y_{minority}$

3: For each minority instances (X_i, Y_i) $\in$ ($X_{minority}$,$Y_{minority}$)

Find the "k nearest neighbors" of X_i in the minority class

4: Generate synthetic samples:

For j=1 to N

- Randomly selects one of the neighbors of X_i
- Generate a synthetic instances $X_{synthetic}$ by interpolating between X_i and its selected neighbor
- Append $X_{synthetic}$ to $X_{augmented}$
- Append Y_i to $Y_{augmented}$

5: Concatenate the input dataset ($X_{resampled}$, $Y_{resampled}$) with the generated dataset($X_{augmented}$,$Y_{augmented}$) and store in ($X_{augmented}$,$Y_{augmented}$)

The hybrid sampling method is applied to the full dataset and its subsets. The resulting numbers are presented in Table 4. With this hybrid sampling method, the ratio of the dataset, which originally is worse than 99:1 becomes approximately 80:20, and this ratio is applied to all the subsets. Prior to sampling, each subset contains 9385 instances labeled as 'Normal' (denoted as '0'), as the occurrence of normal behavior is relatively low across all subsets compared to attack types, except for the 'Theft' attack.

The 80:20 ratio as a figure may still suggest that one class is significantly more prevalent than the other, which in turn leads to a situation whereby the majority class dominates the minority class. Nevertheless, the ratio of 80:20 is used here that sedulously ensures the quality of the majority class as well as the minority class. The overuse of resampling techniques on the data comes with the risk for the quality of data. Table 4 illustrate the figure of the number of records for the dataset and the subsets before and after applying the proposed hybrid sampling method.

Table 4. Results of Hybrid Sampling.

Dataset and subsets	Before Resampling	After Resampling
Full Dataset	1 = 7,33,59,273 0 = 9,385	1 = 7,33,592 0 = 1,83,398
DDoS subset	1 = 3,85,32,478 0 = 9,385	1 = 3,85,324 0 = 93,641
DoS subset	1 = 3,30,05,192 0 = 9,385	1 = 3,30,051 0 = 93,641
Scan subset	1 = 18,20,018 0 = 9,385	1 = 1,82,001 0 = 45,500
Theft subset	1 = 1,585 0 = 9,385	1 = 2,000 0 = 8,000

3.4. Machine Learning Techniques and Performance Metrics

In this study, we select four ML algorithms to conduct binary classification on our dataset and its subsets.

- Random Forest (RF): It create multiple decision trees and combines their predictions through voting or averaging. It's robust to imbalanced datasets is its advantage over other algorithms. It inherently handles class imbalance by giving more weight to minority class samples during training.
- Naive Bayes (NB): It is a classification algorithm developed using Bayes' theorem and with homogeneity assumption which assumes nothing more than independence of features. It is computationally efficient but may not be able to model non-symmetrical distributions in the same way.
- Logistic Regression (LR): It is a linear model used for estimating the likelihood of an event occurring and this involves use of one or more predictor variables. Out of all the modeling algorithms, it is particularly efficient for working with datasets wherein the samples are disbursed unevenly; However, it only works best when the samples of the minority class are easily linearly separable from the plethora of samples of the majority class.
- K-Nearest Neighbor (KNN): It is a non-parametric method that classifies new data points based on the majority class among their k nearest neighbors in feature space. In imbalanced datasets, the majority class instances may dominate the neighborhood of test instances, leading to misclassification of minority class instances.

The selection of these four ML algorithms is based on their diverse classification approaches, aiming to evaluate their performance on the resampled dataset. The dataset is divided into two sets for training and testing where training set comprises 80% and test set comprises 20% of the dataset. The training set is applied on the model to make it learn what it has to do while the testing set is used to check the model's performance. The performance metrics used for model evaluation are:

- Accuracy: Proportion of correctly classified instances relative to the number of instances.
- Precision: Proportion of correctly classified positive instances relative to the number of instances predicted as positive.
- Recall: Proportion of correctly classified positive instances relative to the number of actual positive instances.
- False Positive Rate (FPR): Proportion of incorrectly classified negative instances relative to the number of actual negative instances.
- False Negative Rate (FNR): Proportion of incorrectly classified positive instances relative to the number of actual positive instances.

4. Experimental Results and Analysis

Experiments were conducted on a system with Intel i7 12th gen processor, 32GB of RAM, and NVIDIA GeForce RTX 3060 Ti GPU. The computational tasks were executed on this high-performance hardware configuration to ensure efficient processing and reliable results. The experiments were implemented and executed within the Python Jupyter Notebook environment, utilizing Python libraries such as scikit-learn [31], imbalanced-learn [32], and seaborn (sns) [33].

Hyperparameter tuning was conducted to optimize the performance of the ML algorithms. For RF, the number of trees (n_estimators) was tuned within the range of 50 to 400, and the maximum depth of the trees (max_depth) and minimum split (min_samples_split) are put to default to give maximum flexibility which are 0 and 2 respectively. The optimal configuration was found to be n_estimators = 100, which also provided the best balance between accuracy and computational efficiency. There is little to

no change with this tuning of parameter. For KNN, the value of k (number of neighbors) was tested in the range of 3 to 13, with k = 3 yielding the highest accuracy. Logistic Regression was tuned using L2 regularization, with the regularization strength (C) tested in the range of 0.01 to 10, and the optimal value was found to be C = 10. And lastly for NB, parameter tuning was done on var_smoothing parameter which control the amount of variance added to each feature to prevent numerical instabilities. Var_smoothing was explored over log range from 10^{-10} to 10^{0}. The best var_smoothing value is 10^{-7}. All these hyperparameters were selected using a grid search approach with 5-fold cross-validation to ensure robust performance.

The ML classifiers are evaluated using specific performance metrics on the Full Dataset (Table 5) followed by the DoS Subset, DDoS Subset, Scan Subset, and the Theft Subset in coming Tables. These datasets undergo preprocessing (Section 3.2) and are balanced using the hybrid sampling method (Sec Section 3.3) to achieve an approximate ratio of 80:20 between the classes. A visual representation comparing the of performance metrics of different ML algorithms on the Full dataset is shown in Figure 4.

Table 5. Evaluation results on Full dataset

Classifiers	Acc.	FPR	FNR	Precision	Recall
LR	99.48	1.05	0.38	99.73	99.61
NB	91.80	40.04	0.25	90.89	99.74
KNN	99.96	0.06	0.02	99.97	99.96
RF	99.98	0.02	0.01	99.99	99.98

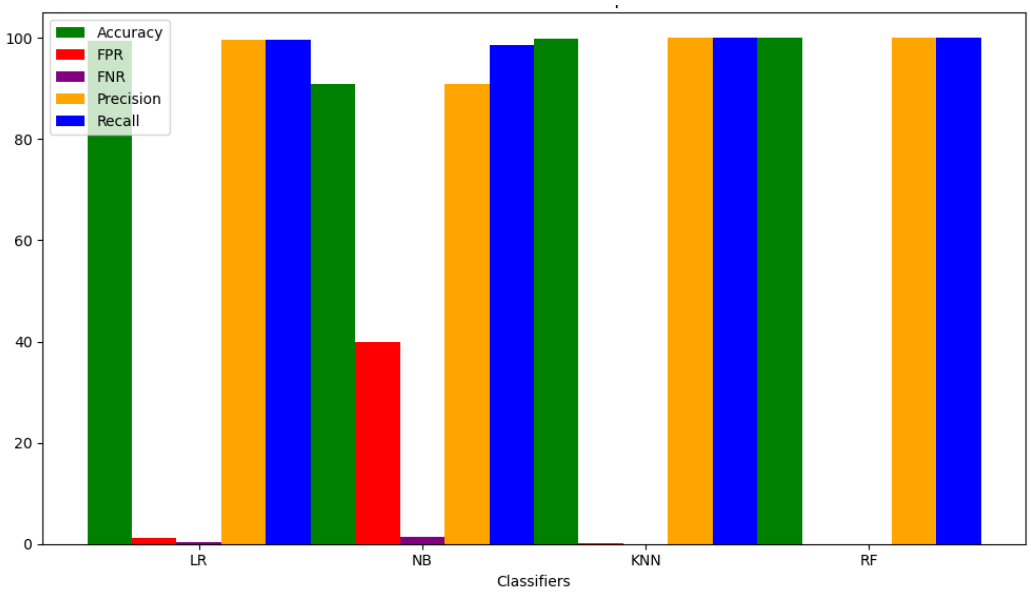

Figure 4. Performance Metrics Comparison.

The data distribution, showcased in Table 4, highlights the distribution before and after sampling. All the ML classifiers were evaluated on the dataset after sampling. In the evaluation of Full dataset after resampling (Table 5), RF outperforms LR, NB and KNN in all the defined performance metrics. Though FPR is mainly focused by researchers, FNR is also equally important as False Negative might lead to the trespassing of networking attacks into the system without being noticed by the IDS or the system.

Table 6 also demonstrates the superiority of RF over other selected classifiers in terms of FPR and FNR. While KNN provides strong competition to RF in the DoS subset, LR and NB do not perform well in comparison to KNN and RF.

Table 6. Evaluation Results on DoS subset.

Classifiers	Acc.	FPR	FNR	Precision	Recall
LR	99.91	0.26	0.03	99.92	99.96
NB	99.89	0.09	1.07	99.97	99.89
KNN	99.98	0.05	0.009	99.98	99.99
RF	99.98	0.03	0.006	99.99	99.99

The evaluation of DDoS subset and Scan subset is shown in Table 7 and Table 8. RF gives superior performance in all metrics.

Table 7. Evaluation Results on DDoS subset.

Classifiers	Acc.	FPR	FNR	Precision	Recall
LR	99.87	0.54	0.01	99.86	99.98
NB	99.61	0.17	0.44	99.95	99.55
KNN	99.97	0.07	0.006	99.98	99.99
RF	99.98	0.04	0.002	99.98	99.99

Table 8. Evaluation results on Scan subset.

Classifiers	Acc.	FPR	FNR	Precision	Recall
LR	98.32	1.30	1.77	99.67	98.22
NB	90.67	42.41	1.11	90.37	98.88
KNN	99.83	0.42	0.09	99.89	99.90
RF	99.98	0.04	0.005	99.98	99.99

In Table 9, on the Theft subset, which is notably smaller than the other subsets and the Full dataset, LR and NB demonstrate significantly higher results compared to RF and KNN, which shows that LR and NB performs well on small dataset.

Table 9. Evaluation Results on Theft subset.

Classifiers	Acc.	FPR	FNR	Precision	Recall
LR	99.90	0.06	0.25	99.74	99.74
NB	99.90	0.06	0.25	99.74	99.74
KNN	99.75	0.06	1.03	99.73	98.96
RF	99.85	0.06	0.51	99.74	99.48

The superior performance of RF across most subsets can be attributed to its ensemble nature, which allows it to capture complex patterns in the data while being robust to noise. However, the strong performance of LR and NB on the Theft subset suggests that simpler models may be more effective for smaller, more balanced datasets where overfitting is a greater risk.

The FPR and FNR chart for the Full dataset and subsets using RF is shown in Figure 5. It is evident from Figure 5 that the Theft subset has high FNR while values of all other FPR and FNR are significantly low.

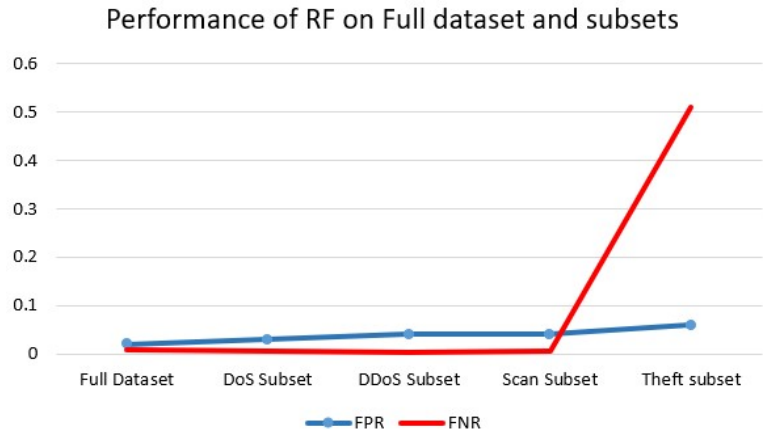

Figure 5. RF Performance on all datasets (FPR and FNR).

In our evaluation, RF outperforms LR, NB, and KNN in terms of FPR, FNR, Precision, and Recall. This may be attributed to RF's robustness to imbalanced datasets. While KNN also yields decent results across the entire datasets, LR and NB do not perform as well as RF and KNN, except for the Theft subset. This discrepancy may be due to their lesser effectiveness with large datasets. Figure 6 and Figure 7 shows the performances of RF on all the datasets using "Accuracy" and "Precision and Recall" respectively.

To provide a clearer visualization of how the model classifies the data in the test set, a confusion

matrix is generated for RF classifier on the Full dataset. The confusion matrix displays the performance of the model by providing the counts of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions.

The confusion matrix (Figure 8) shows the actual number of instances of TP, TN, FP and FN along with their percentage. There are 9 instances of FP and 17 instances of FN which is very low compared to the number of total instances on the test set.

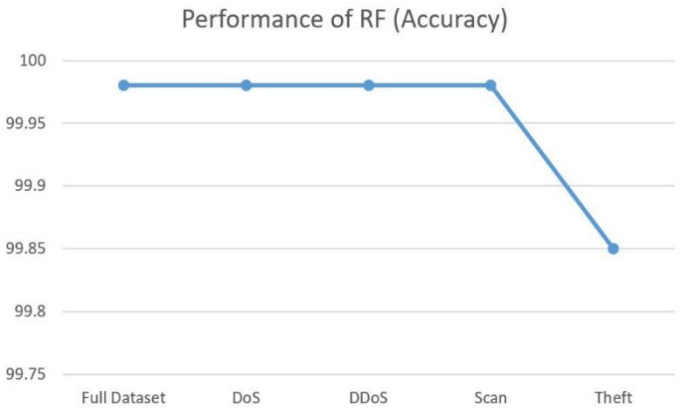

Figure 6. RF performance on all datasets (Accuracy).

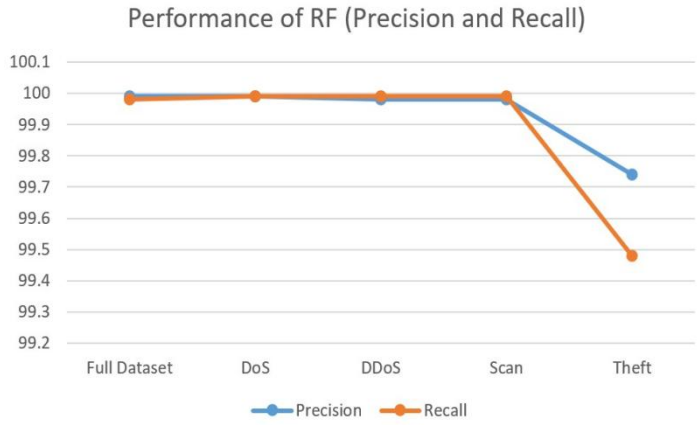

Figure 7. RF performance on all datasets (Precision and Recall).

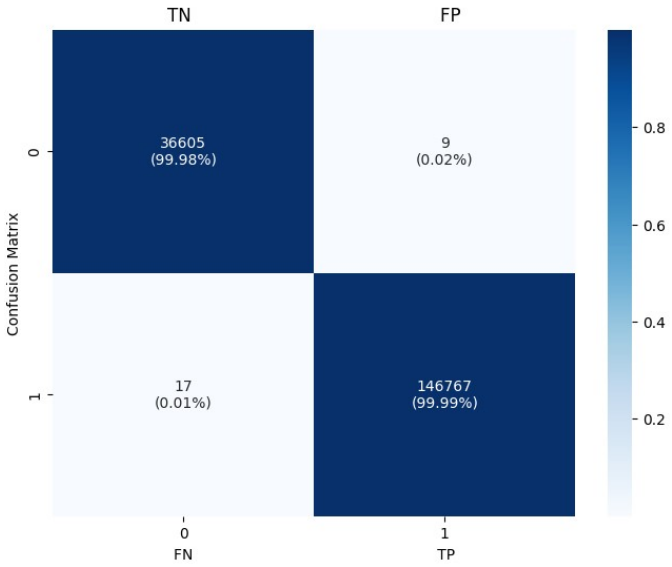

Figure 8. Confusion Matrix of Full Dataset using RF.

5. Discussion

Various researchers have utilized resampling methods to address the extreme imbalance in the Bot-IoT dataset between classes. And most of the researchers used the 5% Bot-IoT dataset which contain 5% of each class labels. Table 10 includes different metrics from recent research studies with the Bot-IoT dataset that involve resampling. Our method is also included for comparison with existing methods.

The proposed hybrid sampling methodology demonstrated remarkable efficacy in addressing the severe class imbalance inherent in the Bot-IoT dataset, enhancing the overall performance of machine learning algorithms used for intrusion detection in IoT networks. By leveraging Near Miss-1 for undersampling and SMOTE for oversampling, the dataset achieved an approximate 80:20 ratio, striking a balance between maintaining data quality and representation of both majority and minority classes. The results underscore the robustness of RF, which consistently outperformed other algorithms across all subsets and the full dataset in different performance metrics. This performance highlights the benefits of ensemble-based methods in capturing intricate patterns within complex datasets while maintaining generalizability.

It is observed from the experimental results that there were variations in performance across the different attack subsets. While RF excelled in most scenarios, simpler models like LR and NB showed superior performance on the Theft subset, which was relatively smaller in size and less complex. This finding suggests that dataset characteristics, including size and balance, significantly influence the effectiveness of classification algorithms. Additionally, while KNN demonstrated competitive performance, particularly on subsets like DoS and DDoS, its reliance on local data density made it more susceptible to misclassification in highly imbalanced scenarios. These insights emphasize the importance of tailoring algorithm selection to the specific characteristics of the dataset and attack type to maximize detection efficiency.

When performing resampling techniques to the dataset, most researchers tend to oversample the minority class. The extreme imbalance in the Bot-IoT dataset can lead to adverse outcomes if the minority class is excessively oversampled to balance the majority class. To mitigate these risks, the hybrid sampling is proposed aiming to prevent overfitting while preserving the quality of the data. Table 10 highlights that our proposed method performs favorably compared to state-of-the-art methods, particularly in terms of FPR.

Table 10. Comparative Analysis with state-of-art work.

Ref.	Sampling Method	Class Ratio	Model	Acc. (%)	FPR (%)
[30]	SMOTE	50:50	KNN	92.1	NA
[34]	Random Undersampling & SMOTE	94:6	ANN	NA	NA
[35]	SMOTE	50:50	DNN	99.80	NA
[36]	SMOTE & Random Undersampling	92:8	ANN	99.98	23.36
[37]	SMOTE	50:50	SVM	99.99	1.3
Our paper	Near Miss & SMOTE	80:20	RF	99.98	0.02

The proposed hybrid sampling method has important implications for real-world IoT security systems. The approach effectively reduces the likelihood of false positives and negatives, which are critical metrics in intrusion detection, to minimize service disruptions and prevent undetected attacks. Furthermore, the study's findings advocate for the adoption of a multi-model framework in IDS deployments, where different algorithms are optimized for specific attack types. This approach could enhance overall system robustness and adaptability, addressing the diverse and evolving threat landscape of IoT environments. Moreover, the varying performance of algorithms across different attack types suggests that a multi-model approach could be more effective in practice. This approach would involve deploying different algorithms optimized for specific types of attacks, creating a more robust and adaptable defense system. By leveraging the strengths of various classifiers for different attack scenarios, such a system could provide more comprehensive protection against the diverse and evolving threat landscape in IoT networks.

6. Conclusion and Future Scope

In conclusion, this study demonstrates the efficacy of a novel hybrid sampling approach in addressing the severe class imbalance inherent in the Bot-IoT dataset, a critical challenge in evaluating IDS for IoT environments. By combining Near Miss-1 for undersampling and SMOTE for oversampling, we

achieved a balanced dataset without compromising data quality, enabling more reliable evaluation of machine learning algorithms for intrusion detection. Our findings reveal that among traditional ML models, RF exhibits superior performance across key metrics including accuracy, FPR, FNR, precision, and recall. KNN shows comparable efficacy, while NB and LR demonstrate particular strength in analyzing the Theft subset, a notably smaller dataset. These results underscore the importance of algorithm selection tailored to specific dataset characteristics and attack types in IoT security contexts.

The implications of this research extend beyond immediate performance improvements, suggesting the potential for multi-model approaches in comprehensive IoT network protection. As IoT ecosystems continue to expand and evolve, the need for sophisticated, adaptive security measures becomes increasingly critical. This study lays a foundation for developing such systems, offering insights into the nuanced performance of different classification algorithms under various attack conditions and dataset characteristics. Future research in this domain could focus on enhancing the realism of botnet traffic simulations and developing adaptive models to address the dynamic nature of IoT ecosystems. Investigations into robust IDS for zero-day attacks and refined sampling techniques could further improve classification performance. Validating the proposed approach across diverse IoT datasets and in real-world deployments would provide insights into its generalizability and practical efficacy. These directions aim to advance IoT security, addressing the evolving challenges posed by sophisticated cyber threats in increasingly interconnected environments.

Author Contributions
All authors contributed equally, and all authors read and approved the final version of the paper.

Funding
This research received no external funding.

Conflict of Interest Statement
The authors declare no conflicts of interest.

Data Availability Statement
The data that supports the findings of this study is openly available at: https://research.unsw.edu.au/projects/bot-iot-dataset (ref [5])

References

1. P. Gokhale, O. Bhat, S. Bhat. “Introduction to iot”, *International Advanced Research Journal in Science, Engineering and Technology*, 5(1), pp. 41–44, 2018. doi: 10.17148/IARJSET.2018.517
2. A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, “Survey of intrusion detection systems: techniques, datasets and challenges”, *Cybersecurity*, 2(1), pp. 1-22, 2019. doi:1 0.1186/s42400-019-0038-7
3. A. Saeed, M. Paul, M. R. Asif, et al. "IoT-based Machine Learning Model for Real-Time Fault Detection and Predictive Maintenance in Smart Industries", *IEEE Access*, 7, pp. 60734-60744, 2019.
4. M. Ring, S. Wunderlich, D. Scheuring, et al. "A Survey of Network-based Intrusion Detection Data Sets", *Computers & Security*, 86, pp. 147-167, 2019. doi: 10.1016/j.cose.2019.06.005
5. N. Koroniotis, N. Moustafa, E. Sitnikova, B. Turnbull. “Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset”, *Future Generation Computer Systems*, 100, pp. 779–796, 2019. doi: 10.1016/j.future.2019.05.041
6. A. Tanimoto, S. Yamada, T. Takenouchi, M. Sugiyama, H. Kashima. “Improving imbalanced classification using near-miss instances”, *Expert Systems with Applications*, 201, pp. 117130, 2022. doi: 10.1016/j.eswa.2022.117130
7. N.V. Chawla, KW. Bowyer, LO. Hall, WP. Kegelmeyer, “Smote: synthetic minority over-sampling technique”, *Journal of artificial intelligence research*, 16, pp. 321-357, 2002. doi: 10.48550/arXiv.1106.1813
8. I. Martins, JS. Resende, PR. Sousa, S. Silva, L. Antunes, J. Gama. ”Host-based IDS: A review and open issues of an anomaly detection system in IoT”, *Future Generation Computer Systems*, 133, pp. 95-113, 2022. doi: 10.1016/j.future.2022.03.001
9. M. Sarhan, S. Layeghy, M. Gallagher, M. Portmann. “From zero-shot machine learning to zero-day attack detection”, *International Journal of Information Security*, 22(4), pp. 947–959, 2023. doi: doi.org/10.1007/s10207-023-00676-0
10. M. Baich, T. Hamim, N. Sael, Y. Chemlal. “Machine learning for iot based networks intrusion detection: a comparative study”, *Procedia Computer Science*, 215, pp. 742–751, 2022. doi: 10.1016/j.procs.2022.12.076
11. M. Mahmoud, M. Kasem, A. Abdallah, HS. Kang. “Ae-lstm: Autoencoder with lstm-based intrusion detection in iot”, In *International Telecommunications Conference (ITC-Egypt)*, pp. 1–6, 2022. doi: 10.1109/ITC-Egypt55520.2022.9855688
12. X. Liu, Y. Du, “Towards effective feature selection for iot botnet attack detection using a genetic algorithm”, *Electronics*, 12(5), pp. 1260, 2023. doi: 10.3390/electronics12051260

13. A. Al-Bakaa, & B. Al-Musawi, “A new intrusion detection system based on using non-linear statistical analysis and features selection techniques. *Computers & Security*, 122, pp. 102906, 2022 doi: 10.1016/j.cose.2022.102906
14. V. Hnamte, A.A. Najar, H. Nhung-Nguyen, J. Hussain, & M.N. Sugali. “DDoS attack detection and mitigation using deep neural network in SDN environment”. *Computers & Security*, 138, 2024, 103661.
15. G.P. Fernando, A.M. Florina, & C.B. Liliana. “Evaluation of the performance of unsupervised learning algorithms for intrusion detection in unbalanced data environments” *IEEE Access*, 12, 2024
16. E. Tsogbaatar, MH. Bhuyan, Y. Taenaka, et al. “Del-iot: A deep ensemble learning approach to uncover anomalies in iot”, *Internet of Things*, 14, pp. 100391, 2021. doi: 10.1016/j.iot.2021.100391
17. M. Roopak, GY. Tian, J. Chambers. “An intrusion detection system against ddos attacks in iot networks”, In *10th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 0562–0567, 2020. doi: 10.1109/CCWC47524.2020.9031206
18. R. Lalduhsaka, N. Bora, AK. Khan, “Anomaly-based intrusion detection using machine learning: An ensemble approach”, *International Journal of Information Security and Privacy*, 16(1), pp. 1–15, 2022. doi: 10.4018/IJISP.311466
19. AA. Anitha, L. Arockiam. ”A review on intrusion detection systems to secure IoT networks”, *International Journal of Computer Networks and Applications*, 9(1), pp. 38-50, 2022. doi: 10.22247/ijcna/2022/211599
20. S. Alosaimi, SM. Almutairi. “An intrusion detection system using bot-iot”, *Applied Sciences*, 13(9), pp. 5427, 2023. doi: 10.3390/app13095427
21. JL. Leevy, TM. Khoshgoftaar, J. Hancock. “Iot attack prediction using big botiot data”, *International Journal of Internet of Things and Cyber-Assurance*, 2(1), pp. 45–61, 2022. doi: 10.1504/IJITCA.2022.124373
22. K. Ibrahimi, H. Benaddi. “Improving the ids for bot-iot dataset-based machine learning classifiers”, In *5th International Conference on Advanced Communication Technologies and Networking (CommNet)*, pp. 1–6, 2022. doi: 10.1109/CommNet56067.2022.9993869
23. DB. Aruna, N. Karthik. “The Effect of Anomaly Detection and Data Balancing in Prediction of Diabetes”, *International Journal of Computer Information Systems and Industrial Management Applications*, 16(2), pp. 184-196, 2024. url: https://cspub-ijcisim.org/index.php/ijcisim/article/view/631
24. JL. Leevy, J. Hancock, TM. Khoshgoftaar, J. Peterson. “Detecting information theft attacks in the bot-iot dataset”, In *20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 807–812, 2021. doi: 10.1109/ICMLA52953.2021.00133
25. A. Mihoub, OB. Fredj, O. Cheikhrouhou, A. Derhab, M. Krichen. “Denial of service attack detection and mitigation for internet of things using looking-back-enabled machine learning techniques”, *Computers & Electrical Engineering*, 98, pp. 107716, 2022. doi: 10.1016/j.compeleceng.2022.107716
26. J. Ramprasath, N. Krishnaraj, V. Seethalakshmi. "Mitigation services on SDN for distributed denial of service and denial of service attacks using machine learning techniques", *IETE Journal of Research*, 70(1), pp. 70-81, 2024. doi: 10.1080/03772063.2022.2142163
27. A. Ahad, MS. Khan, P. Saxena. “Analysis of keylogging spyware for information theft”, In *1st International Conference on Intelligent Computing and Research Trends (ICRT)*, pp. 1-4, 2023. doi: 10.1109/ICRT57042.2023.10146672
28. L. Yu, R. Zhou, R. Chen, KK. Lai. “Missing data preprocessing in credit classification: One-hot encoding or imputation?”, *Emerging Markets Finance and Trade*, 58(2), pp. 472–482, 2022. doi: 10.1080/1540496X.2020.1825935
29. P. Ferreira, DC. Le, N. Zincir-Heywood. “Exploring feature normalization and temporal information for machine learning based insider threat detection”, In *15th International Conference on Network and Service Management (CNSM)*, pp. 1-7, 2019. doi: 10.23919/CNSM46954.2019.9012708
30. S. Pokhrel, R. Abbas, B. Aryal, “Iot security: botnet detection in iot using machine learning”, arXiv preprint, 2021. doi: 10.48550/arXiv.2104.02231
31. F. Pedregosa, G. Varoquaux, A. Gramfort, et al. “Scikit-learn: Machine learning in Python”, *Journal of Machine Learning Research*, 12, pp. 2825–2830, 2011.
32. G. Lemaˆıtre, F. Nogueira, CK. Aridas. “Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning”, *Journal of Machine Learning Research*, 18(17), pp. 1-5, 2017. doi: doi.org/10.48550/arXiv.1609.06570
33. ML. Waskom. “seaborn: statistical data visualization”, *Journal of Open Source Software*, 6(60), pp. 3021, 2021. doi: seaborn: statistical data visualization
34. S. Bagui, K. Li. “Resampling imbalanced data for network intrusion detection datasets”, *Journal of Big Data*, 8(1), pp. 6, 2021. doi: 10.1186/s40537-020-00390-x
35. Y. Hou, Y. Fu, J. Guo, J. Xu, R. Liu, X. Xiang. “Hybrid intrusion detection model based on a designed autoencoder”, *Journal of Ambient Intelligence and Humanized Computing*, 14(8), pp. 10799-10809, 2022. doi: 10.1007/s12652-022-04350-6
36. D. Kulkarni, R. Jaiswal. “An intrusion detection system using extended kalman filter and neural networks for iot networks”, *Journal of Network and Systems Management*, 31(3), pp. 56, 2023. doi: 10.1007/s10922-023-09748-x
37. J. Atuhurra, T. Hara, Y. Zhang, M. Sasabe, S. Kasahara, “Dealing with Imbalanced Classes in Bot-IoT Dataset”, arXiv e-prints:2403.18989, 2024. doi: 10.48550/arXiv.2403.18989.